



银河麒麟高级服务器操作系统 V10

开发接口使用指南

麒麟软件有限公司

2024 年 03 月

目录

一，概述	7
二，如何安装使用开发需要的软件包	7
dnf 命令和 rpm 命令	7
麒麟 V10 中 devel 包	9
三，C/C++语言开发接口——内核与 C 库的 man 手册	9
如何打开 man 文档	11
四，python 语言开发接口——help()	13
如何寻找安装需要的软件包	13
如何查看已安装软件的 API 接口	14
五，java 语言——javadoc	15
麒麟 V10 中 java 相关软件包介绍	15
javadoc 使用方法	16
六，内核驱动开发	17
附 录	20
1 应用编程接口函数说明	20
1.1 字符操作接口	20
1.2 字符串操作接口	22
1.3 内存控制接口	24
1.4 内存及字符串操作接口	25
1.5 日期和时间操作接口	30
1.6 数学函数操作接口	33
1.7 用户与组操作接口	38
1.8 文件操作	46

1.9 文件内容操作	51
1.10 文件权限控制接口	58
1.11 进程操作接口	68
1.12 信号处理接口	73
1.13 环境变量操作接口	78
1.14 终端控制操作接口	79
1.15 输入输出接口	79
1.16 图形编程接口	83
1.16.1 GTK 图形编程	83
1.17 JDK 编程接口	86
2 用户操作接口说明	89
2.1 终端命令	89
3 第三方软件约束要求	100
3.1 第三方应用移植约束要求	100
3.1.1 数据结构	100
4 错误码定义	102
5 sp3 2303 和 sp2 的 API 接口差异	105
glibc-devel	105
kernel-devel	106
postgresql-devel	106
rpm-devel	106
ndctl-devel	106
tpm2-tss-devel	107
gdk-pixbuf2-devel	108
libreport-devel	108
evolution-data-server-devel	108
openssl-devel	108
gobject-introspection-devel	109
mariadb-devel	110
librsvg2-devel	110
libldb-devel	110
glib2-devel	110
rdma-core-devel	117
avahi-devel	120
libcap-ng-devel	120
libbpf-devel	120
libtar-devel	125
audit-devel	125
subunit-devel	125
libsodium-devel	125
ibus-devel	126
clibcni-devel	127
lcr-devel	127
daxctl-devel	127

libfastjson-devel	128
e2fsprogs-devel	128
libbox1-devel	128
harfbuzz-devel	129
openccc-devel	129
systemd-devel	130
vala-devel	130
efl-devel	130
dpdk-devel	130
变动包清单	130
6 sp3 2403 和 sp3 2303 的 API 接口差异	131
libssh-devel	132
rpm-devel	132
libkysec-devel	132
nagios-devel	132
poppler-glib-devel	133
postgresql-devel	135
check-devel	136
libabigail-devel	136
poppler-qt5-devel	136
libtommath-devel	142
nss-softokn-devel	147
bind-devel	147
kmmod-devel	148
kysec-common-devel	148
espeak-ng-devel	148
grpc-devel	149
hsdimm-lite-devel	149
evolution-data-server-devel	149
openssl-devel	149
mariadb-devel	150
proftpd-devel	150
openvswitch-devel	154
libwinpr-devel	155
freerdp-devel	159
poppler-cpp-devel	161
jemalloc-devel	162
dovecot-devel	163
gnutls-devel	187
libmtp-devel	187
samba-devel	187
gdm-devel	188
armadillo-devel	188
jsoncpp-devel	188

poppler-devel	188
cryptsetup-devel	188
papi-devel	189
NetworkManager-libnm-devel	190
avahi-devel	190
python3-devel	190
qt5-qtbase-devel	190
libvncserver-devel	192
liborcus-devel	193
libgcrypt-devel	206
sleuthkit-devel	207
hdf5-mpich-devel	207
protobuf-c-devel	256
libkysec-extend-devel	256
ImageMagick-c++-devel	258
libbpf-devel	259
protobuf-devel	275
varnish-devel	275
nettle-devel	302
libtar-devel	303
eom-devel	303
libdwarf-devel	303
nodejs-devel	320
virglrenderer-devel	320
libkysecwhlist-devel	320
php-devel	321
hdf5-devel	349
libnghttp2-devel	349
libwebsockets-devel	349
libreofficekit-devel	362
lcr-devel	364
libdvdnav-devel	365
e2fsprogs-devel	365
wireshark-devel	365
multipath-tools-devel	366
hiredis-devel	366
asio-devel	374
libevhttp-devel	374
hdf5-openmpi-devel	374
libpcap-devel	423
efivar-devel	424
ImageMagick-devel	426
systemd-devel	427
python3-pillow-devel	428

bash-devel	441
httpd-devel	441
bind-export-devel	443
nss-devel	443
pmix-devel	443
fcitx-devel	456
kernel-headers	457
变动包清单	458

一，概述

开发接口（简称 API）是操作系统中的软件模块提供的、供新开发应用程序源码使用的函数、宏定义等。银河麒麟高级服务器操作系统 V10（文中简称麒麟 V10）包含了大量的开发接口，本文将呈现麒麟 V10 中开发接口相关的各方面信息，希望对麒麟 V10 上的开发工作有所帮助。

二，如何安装使用开发需要的软件包

软件开发工作经常需要安装一些软件包。安装以后，需要查看这个包提供了什么可供我们使用的文件。因此，开发者必须掌握软件包管理工具的基本使用方法。

dnf 命令和 rpm 命令

dnf 命令，可以搜索仓库中的软件、可以安装指定的软件到硬盘上。rpm 命令用于管理已经安装好的软件，以及安装查看下载到本地的 rpm 包。

下面是使用举例和说明。

dnf search '*xdg*'——搜索包名和包描述中包含 xdg 的软件。

```
[kylin@localhost ~]$  
[kylin@localhost ~]$ dnf search '*xdg*'
build
上次元数据过期检查：0:00:04 前，执行于 2022年11月09日 星期三 10时23分10秒。
===== Name 和 Summary 匹配: *xdg* =====
xdg-dbus-proxy-help.noarch : Documents for xdg-dbus-proxy
xdg-desktop-portal-devel.x86_64 : Development files for xdg-desktop-portal
xdg-desktop-portal-gtk.x86_64 : A backend implementation for xdg-desktop-portal
xdg-desktop-portal-help.noarch : Documents for xdg-desktop-portal
xdg-user-dirs-help.noarch : help document for the xdg-user-dirs package
xdg-utils-help.noarch : Help document for the xdg-utils package
===== Name 匹配: *xdg* =====
python2-pyxdg.noarch : Python2 library to access freedesktop APIs
python3-pyxdg.noarch : Python3 library to access freedesktop APIs
xdg-dbus-proxy.x86_64 : Filtering proxy for D-Bus connections
xdg-desktop-portal.x86_64 : A portal frontend service for Flatpak and possibly other desktop containment frameworks
xdg-user-dirs.x86_64 : A tool to help manage "well known" user directories
xdg-user-dirs-gtk.x86_64 : Integrated tools for gnome to help manage directories
xdg-utils.noarch : A set of easily desktop environment integration tools
===== Summary 匹配: *xdg* =====
libcanberra.x86_64 : XDG Sound Theme and Name Specifications
[kylin@localhost ~]$
```

dnf repoquery '*kmod*'——搜寻仓库中包名包含 kmod 的包，注意引号和星号。

dnf repoquery '*xorg*devel*'——搜寻仓库中包名包含 xorg 和 devel 的包。

sudo dnf install kmod-help——安装 kmod-help 包。

```
[kylin@localhost ~]$ sudo dnf install kmod-help-0:27-10.se.01.ky10.x86_64
[sudo] kylin 的密码：
上次元数据过期检查：2:23:22 前，执行于 2022年11月09日 星期三 08时04分44秒。
依赖关系解决。
=====
Package                               Architecture           Version
=====
安装：
kmod-help                             x86_64                  27-10.se.01.ky10
=====
事务概要
-----
安装 1 软件包
总下载：37 k
安装大小：45 k
确定吗？ [y/N]： █
```

`rpm -qa` ——列出所有已经安装的软件。

`rpm -qa | grep python3-help` ——检查是否安装了 `python3-help`。

`sudo dnf remove python3-cups`——卸载软件包 `python3-cups`。如果不能确认提示内容，请输入 N，放弃删除操作。

```
[kylin@localhost ~]$ sudo dnf remove python3-cups
依赖关系解决。
=====
Package                               Architecture           Version
=====
移除：
python3-cups                          x86_64                  1.9.72-24.ky10
移除依赖的软件包：
system-config-printer                 x86_64                  1.5.11-16.01.p01.ky10
清除未被使用的依赖关系：
python3-pycurl                        x86_64                  7.43.0.3-1.ky10
system-config-printer-libs            noarch                  1.5.11-16.01.p01.ky10
=====
事务概要
-----
移除 4 软件包
将会释放空间：6.2 M
确定吗？ [y/N]： █
```

`rpm -ql kmod-help`——查询 `glibc` 这个软件包提供了哪些文件

```
[kylin@localhost ~]$ rpm -ql kmod-help
/usr/share/doc/kmod-help
/usr/share/doc/kmod-help/NEWS
/usr/share/doc/kmod-help/README
/usr/share/doc/kmod-help/TODO
/usr/share/man/man5/depmod.d.5.gz
/usr/share/man/man5/modprobe.conf.5.gz
/usr/share/man/man5/modprobe.d.5.gz
/usr/share/man/man5/modules.dep.5.gz
/usr/share/man/man5/modules.dep.bin.5.gz
/usr/share/man/man8/depmod.8.gz
/usr/share/man/man8/insmod.8.gz
/usr/share/man/man8/kmod.8.gz
/usr/share/man/man8/lsmmod.8.gz
/usr/share/man/man8/modinfo.8.gz
/usr/share/man/man8/modprobe.8.gz
/usr/share/man/man8/rmmmod.8.gz
[kylin@localhost ~]$ █
```

`rpm -qf /usr/share/man/man3/fseek.3.gz`——查询文件 `fseek.3.gz` 属于哪个软件包

```
[root@localhost man3]# rpm -qf fseek.3.gz
man-pages-5.02-6.ky10.noarch
[root@localhost man3]# rpm -qf /usr/share/man/man3/fseek.3.gz
man-pages-5.02-6.ky10.noarch
[root@localhost man3]#
```

dnf repoquery --queryformat %{Sourcerpm},{Name} | sort ——查询结果以源码包名称排序，从中可以看到每个源码包提供的二进制包名称。

```
yp-tools-4.2.3-4.ky10.src.rpm,yp-tools
yp-tools-4.2.3-4.ky10.src.rpm,yp-tools-help
yum-langpacks-0.4.5-10.oel.src.rpm,yum-langpacks
yum-langpacks-0.4.5-10.oel.src.rpm,yum-langpacks-help
yum-metadata-parser-1.1.4-24.ky10.src.rpm,yum-metadata-parser
yum-metadata-parser-1.1.4-24.ky10.src.rpm,yum-metadata-parser-help
zaf-0-0.17.20080714svn.ky10.src.rpm,hyphen-af
zaf-0-0.17.20080714svn.ky10.src.rpm,hyphen-zu
zbar-0.20.1-4.oel.src.rpm,zbar
zbar-0.20.1-4.oel.src.rpm,zbar-devel
zbar-0.20.1-4.oel.src.rpm,zbar-help
zd1211-firmware-1.5-5.ky10.src.rpm,zd1211-firmware
zd1211-firmware-1.5-5.ky10.src.rpm,zd1211-firmware-help
zenity-3.32.0-1.ky10.src.rpm,zenity
zenity-3.32.0-1.ky10.src.rpm,zenity-help
zerofree-1.1.1-3.ky10.src.rpm,zerofree
zerofree-1.1.1-3.ky10.src.rpm,zerofree-help
zeromq-4.1.7-2.ky10.src.rpm,cppzmq-devel
zeromq-4.1.7-2.ky10.src.rpm,zeromq
zeromq-4.1.7-2.ky10.src.rpm,zeromq-devel
zeromq-4.1.7-2.ky10.src.rpm,zeromq-help
zinc-0.3.1-7.ky10.src.rpm,zinc
```

麒麟 V10 中 devel 包

麒麟 V10 的仓库中有一千多个 devel 包，比如 glibc-devel、kmod-devel。这些软件是面向开发人员使用的。大部分开发工作需要的 API 就来自这些软件包。一个明显的例外就是 kernel-headers 包，它是内核空间开放给用户空间的接口。

三，C/C++语言开发接口——内核与 C 库的 man 手册

Linux 内核属于宏内核、功能繁多。除了内务型的内核功能，内核必须能支撑用户空间程序的运行。所以内核提供了 kernel-headers 这个包，是内核空间开放给用户空间的接口。不过我们多数情况不会直接调用 kernel-headers 的 API，应该使用 glibc 提供的系统调用。

C 库，即 glibc 软件包，是用户空间的基础。它提供的 API 可以分为：syscall

和 C 库调用。syscall，即系统调用，是部分内核 API 的封装。

鉴于内核和 C 库的 API 非常重要，麒麟 V10 中已经集成了相关的 man 文档。

这些文档被封装到了 man-pages 软件包中。

```
[root@localhost man3]# rpm -qi man-pages
Name       : man-pages
Version    : 5.02
Release    : 6.ky10
Architecture: noarch
Install Date: 2022年09月15日 星期四 11时16分54秒
Group      : Unspecified
Size       : 3146037
License    : GPL+ and GPLv2+ and BSD and MIT and Copyright only and IEEE
Signature  : RSA/SHA1, 2021年03月16日 星期二 16时14分23秒, Key ID 41f8aeb7a486d9f
Source RPM : man-pages-5.02-6.ky10.src.rpm
Build Date : 2021年03月06日 星期六 17时40分35秒
Build Host : localhost.localdomain
Packager   : Kylin Linux
Vendor     : KylinSoft
URL        : http://www.kernel.org/doc/man-pages/
Summary    : Linux kernel and user-space C library interfaces documentation
Description:
The Linux man-pages project documents the Linux kernel and C library interface that
are employed by user-space programs. With respect to the C library, the primary focus is
the GNU C library (glibc), although, where known, documentation of variations on
other C libraries available for Linux is also included.
[root@localhost man3]#
```

可以按照下面的说明查阅：

/usr/share/man/man2/——这里存放的是 C 库提供的内核调用接口。

```
[root@localhost ~]# cd /usr/share/man/man2
[root@localhost man2]# ls
accept.2.gz      fdetach.2.gz      inb_p.2.gz        mlock2.2.gz      prlimit.2.gz
accept4.2.gz     fgetxattr.2.gz    init_module.2.gz  mlock.2.gz       prlimit64.2.gz
access.2.gz      finit_module.2.gz inl.2.gz           mlockall.2.gz    process_vm_readv.2.gz
acct.2.gz        flock.2.gz         inl_p.2.gz         mmap.2.gz         process_vm_writev.2.gz
add_key.2.gz     fork.2.gz          inotify_add_watch.2.gz modify_ldt.2.gz   pselect.2.gz
adjtimex.2.gz    free_hugepages.2.gz inotify_init.2.gz   mount.2.gz        pselect6.2.gz
afs_syscall.2.gz fremovexattr.2.gz inotify_init.2.gz  move_pages.2.gz   ptrace.2.gz
alarm.2.gz       fsetxattr.2.gz    inotify_rm_watch.2.gz mprotect.2.gz     putmsg.2.gz
alloc_hugepages.2.gz fstat.2.gz        insb.2.gz          mpx.2.gz          putpmsg.2.gz
arch_prctl.2.gz  fstat64.2.gz      insl.2.gz          mq_getsetattr.2.gz pwrite.2.gz
arm_fadvise.2.gz fstatat.2.gz       insw.2.gz          mq_notify.2.gz    pwrite64.2.gz
arm_fadvise64.2.gz fstatat64.2.gz    intro.2.gz         mq_open.2.gz      pwritev.2.gz
arm_sync_file_range.2.gz fstatfs.2.gz      inw_p.2.gz         mq_timedreceive.2.gz query_module.2.gz
bdflush.2.gz     fstatfs64.2.gz    io_cancel.2.gz     mq_timedsend.2.gz quotactl.2.gz
bind.2.gz        fstatvfs.2.gz     ioctln_console.2.gz mremap.2.gz       read.2.gz
bpf.2.gz         fsync.2.gz        ioctl.2.gz          msgctl.2.gz       readahead.2.gz
brk.2.gz         ftruncate.2.gz    ioctl_fat.2.gz     msgget.2.gz       readdir.2.gz
cacheflush.2.gz  ftruncate64.2.gz  ioctl_ficlone.2.gz msgop.2.gz         readlink.2.gz
capget.2.gz      futimesat.2.gz    ioctl_fidemap.2.gz msgrcv.2.gz        readlinkat.2.gz
capset.2.gz      getcontext.2.gz   ioctl_getfsmap.2.gz msgsnd.2.gz        readv.2.gz
chdir.2.gz       getcpu.2.gz       ioctl_iflags.2.gz  msync.2.gz        reboot.2.gz
chown.2.gz       getcwd.2.gz       ioctl_list.2.gz    munlock.2.gz       recv.2.gz
chown32.2.gz     getdents.2.gz     ioctl_ns.2.gz      munlockall.2.gz   recvfrom.2.gz
chroot.2.gz      getdents64.2.gz   ioctl_tty.2.gz     munmap.2.gz        recvmmsg.2.gz
clock_getres.2.gz getdomainname.2.gz ioctl_userfaultfd.2.gz name_to_handle_at.2.gz recvmmsg.2.gz
clock_gettime.2.gz getdtablesize.2.gz io_destroy.2.gz    nanoSleep.2.gz    remap_file_pages.2.gz
clock_nanosleep.2.gz getegid.2.gz       io_getevents.2.gz  newfsstat.2.gz    removexattr.2.gz
clock_settime.2.gz getegid32.2.gz     ioperm.2.gz        _newselect.2.gz   rename.2.gz
clone.2.gz        geteuid.2.gz      iopl.2.gz          _nfservctl.2.gz   renameat.2.gz
clone2.2.gz       geteuid32.2.gz    ioprio_get.2.gz    nice.2.gz          renameat.2.gz
clone3.2.gz       getgid.2.gz        io_prio_set.2.gz   oldfstat.2.gz     request_key.2.gz
```

/usr/share/man/man3/——这里存放的是 C 库调用接口，以及 libev、systemd、openssl

等 API。这一部分的文件数量可能很多，与系统安装的包数量有关。除了软件包

man-pages，提供 man3 手册的软件包的名称中一般都包含 ‘-help’。您的电脑

上显示的内容很可能和下面图片所示内容有差异。

```
ftell.3.gz      RSA_meth_get_multi_prime_keygen.3_openssl.gz  xdr_u_char.3.gz
ftello.3.gz     RSA_meth_get_priv_dec.3_openssl.gz             xdr_u_int.3.gz
ftime.3.gz      RSA_meth_get_priv_enc.3_openssl.gz             xdr_u_long.3.gz
ftok.3.gz       RSA_meth_get_pub_dec.3_openssl.gz         xdr_union.3.gz
ftrylockfile.3.gz RSA_meth_get_pub_enc.3_openssl.gz             xdr_u_short.3.gz
fts.3.gz        RSA_meth_get_sign.3_openssl.gz           xdr_vector.3.gz
fts_children.3.gz RSA_meth_get_verify.3_openssl.gz             xdr_void.3.gz
fts_close.3.gz   RSA_meth_new.3_openssl.gz                   xdr_wrapstring.3.gz
fts_open.3.gz    RSA_meth_set0_app_data.3_openssl.gz xencrypt.3.gz
fts_read.3.gz    RSA_meth_set1_name.3_openssl.gz          xprt_register.3.gz
fts_set.3.gz     RSA_meth_set_bn_mod_exp.3_openssl.gz    xprt_unregister.3.gz
ftw.3.gz        RSA_meth_set_finish.3_openssl.gz       y0.3.gz
function::-.3stap.gz RSA_meth_set_flags.3_openssl.gz             y0f.3.gz
function::abort.3stap.gz RSA_meth_set_init.3_openssl.gz             y0l.3.gz
function::addr.3stap.gz RSA_meth_set_keygen.3_openssl.gz            y1.3.gz
function::addr_to_node.3stap.gz RSA_meth_set_mod_exp.3_openssl.gz           y1f.3.gz
function::ansi_clear_screen.3stap.gz RSA_meth_set_multi_prime_keygen.3_openssl.gz y1l.3.gz
function::ansi_cursor_hide.3stap.gz  RSA_meth_set_priv_dec.3_openssl.gz          ynl.3.gz
function::ansi_cursor_move.3stap.gz  RSA_meth_set_priv_enc.3_openssl.gz          ynf.3.gz
function::ansi_cursor_restore.3stap.gz RSA_meth_set_pub_dec.3_openssl.gz           ynl.3.gz
[root@localhost man3]# pwd
/usr/share/man/man3
[root@localhost man3]# ls | wc -l
7182
[root@localhost man3]#
```

```
Other C libraries available for Linux is also included
[root@localhost man3]# rpm -qf ./.* | sort | uniq

dhcp-help-4.4.2-7.ky10.noarch
e2fsprogs-help-1.45.6-8.ky10.noarch
iproute-help-5.5.0-10.p02.ky10.noarch
libev-help-4.33-3.ky10.x86_64
libqb-devel-1.0.3-10.ky10.ky10.x86_64
libseccomp-help-2.5.0-3.ky10.noarch
libssh2-help-1.9.0-6.ky10.noarch
libxml2-help-2.9.10-25.ky10.noarch
man-pages-5.02-6.ky10.noarch
multipath-tools-help-0.8.4-12.ky10.x86_64
net-snmp-help-5.9-4.p01.ky10.noarch
nfs-utils-help-2.5.1-5.ky10.x86_64
nftables-help-0.9.6-4.ky10.noarch
numactl-devel-2.0.13-4.ky10.x86_64
openssl-help-1.1.1f-15.p05.ky10.noarch
perl-PIRPC-0.2020-14.ky10.noarch
perl-Unicode-EastAsianWidth-12.0-1.ky10.noarch
procps-ng-help-3.3.16-17.ky10.noarch
systemd-devel-243-55.p01.se.01.ky10.x86_64
systemtap-help-4.3-2.ky10.x86_64
[root@localhost man3]#
```

如何打开 man 文档

从前面介绍可知，manX 目录中.gz 结尾的文件就是我们需要的 man 类型的文档。每一个 man 文档对应一个或者一组 API 说明。比如 /usr/share/man/man3/string.3.gz，就是我们经常使用的“#include <strings.h>”对应的帮助。执行命令“man string”可以打开帮助内容，下图是开头部分。可以看到 strcmp、strcpy 等函数的说明。

```
STRING(3) Linux Programmer's Manual STRING
NAME
    stpcpy, strcasecmp, strcat, strchr, strcmp, strcoll, strcpy, strcspn, strdup, strfry, strlen, strncat, strncmp, strncpy, strncasecmp, strpbrk, strchr, strsep, strspn, strstr, strtok, strxfrm, index, rindex - string operations
SYNOPSIS
    #include <strings.h>

    int strcasecmp(const char *s1, const char *s2);
        Compare the strings s1 and s2 ignoring case.

    int strncasecmp(const char *s1, const char *s2, size_t n);
        Compare the first n bytes of the strings s1 and s2 ignoring case.

    char *index(const char *s, int c);
        Return a pointer to the first occurrence of the character c in the string s.

    char *rindex(const char *s, int c);
        Return a pointer to the last occurrence of the character c in the string s.

    #include <string.h>

    char *strcpy(char *dest, const char *src);
        Copy a string from src to dest, returning a pointer to the end of the resulting string at dest.

    char *strcat(char *dest, const char *src);
        Append the string src to the string dest, returning a pointer dest.

    char *strchr(const char *s, int c);
        Return a pointer to the first occurrence of the character c in the string s.

    int strcmp(const char *s1, const char *s2);
```

麒麟 V10 支持 man 命令的补全功能，即输入“man ”以后按 TAB 键可以看到如下图所示提问。

```
[root@localhost man3]# man
Display all 11960 possibilities? (y or n)
```

或者输入“man st”只提示 st 开头的帮助。

```
[root@localhost man]# man st
st
standards      stime          strcpy         string          strongswan_pki strongswan_scepclient strtoumax
stat            storeid_file_rewrite strcspn        strlen         strongswan_pki---acert strongswan_sec-updater strtok
stat64         storeutil     strdup         strncasecmp    strongswan_pki---dn strongswan_swancctl strtok_r
statfs         stpcpy        strdupa       strncat        strongswan_pki---gen strongswan_swancctl.conf strtol
statfs64       stpncpy       strerror      strncmp        strongswan_pki---issue strongswan_sw-collector strtold
statvfs        strace        strerror_l    strncpy       strongswan_pki---keyid strpbrk
statx          strace-log-merge strerror_r    strndup       strongswan_pki---pkcs7 strptime
stdarg         strcasecmp    strfmon_l     strndupa      strongswan_pki---print strchr
stderr         strcasestr    strfmon_l     strnlen       strongswan_pki---pub strsep
stdin          strcat        strfmond      strongswan     strongswan_pki---req strsignal
stdio          strchr        strfromf     strongswan_charon-cmd strongswan_pki---self strspn
stdio ext      strchrnul     strfroml     strongswan.conf strongswan_pki---signcrl strstr
stdout         strcmp        strfry       strongswan_ipsec.conf strongswan_pki---verify strtod
strdup         strcoll       strftime     strongswan_ipsec.secrets strongswan_pt-tls-client strtouq
strxfrm        strxfrm       strxfrm       strxfrm       strxfrm       strxfrm
tty            tty            tty            tty            tty            tty
```

下面是常用举例：

man 2 intro——打开 man2 即内核调用部分的总体介绍。

```
INTRO(2) Linux Programmer's Manual INTRO(2)
NAME
    intro - introduction to system calls
DESCRIPTION
    Section 2 of the manual describes the Linux system calls. A system call is an entry point into the Linux kernel. Usually, system calls are not invoked directly; instead, most system calls have corresponding C library wrapper functions which perform the steps required (e.g., trapping to kernel mode) in order to invoke the system call. Thus, making a system call looks the same as invoking a normal library function.

    In many cases, the C library wrapper function does nothing more than:

    * copying arguments and the unique system call number to the registers where the kernel expects them;
    * trapping to kernel mode, at which point the kernel does the real work of the system call;
```

man 3 intro——打开 man3 即来自 C 库等软件包的 API 函数总体介绍

```
INTRO(3) Linux Programmer's Manual INTRO(3)
NAME
    intro - introduction to library functions
DESCRIPTION
    Section 3 of the manual describes all library functions excluding the library functions (system call wrappers) described in Section 2, which implement system calls.

    Many of the functions described in the section are part of the Standard C Library (libc). Some functions are part of other libraries (e.g., the math library, libm, or the real-time library, librt) in which case the manual page will indicate the linker option needed to link against the required library (e.g., -lm and -lrt, respectively, for the aforementioned libraries).

    In some cases, the programmer must define a feature test macro in order to obtain the declaration of a function from the header file specified in the man page SYNOPSIS section. (Where required, these feature test macros must be defined before including any header files.) In such cases, the required macro is described in the man page. For further information on feature test macros, see feature_test_macros(7).

CONFORMING TO
    Certain terms and abbreviations are used to indicate UNIX variants and standards to which calls in this section conform. See standards(7).

NOTES
```

man 3 adjtime——打开 C 库中 adjtime 函数的使用说明。


```

ADJTIME(3) Linux Programmer's
NAME
    adjtime - correct the time to synchronize the system clock
SYNOPSIS
    #include <sys/time.h>

    int adjtime(const struct timeval *delta, struct timeval *olddelta);

    Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

    adjtime():
        Since glibc 2.19:
            DEFAULT_SOURCE
        Glibc 2.19 and earlier:
            _BSD_SOURCE
DESCRIPTION
    The adjtime() function gradually adjusts the system clock (as returned by gettimeofday(2)),
    pointed to by delta. This structure has the following form:

```

另外 man 命令允许参数是文件名，绝对路径或者相对路径都可以。比如，
“man ./man3/adjtime.3.gz”。

四，python 语言开发接口——help()

如何寻找安装需要的软件包

麒麟 V10 中的 python 相关的软件包的名称中都有“python”。麒麟 V10 支持 python2.7 和 python3.7。下图是举例：

```

python2-gobject-3.36.1-2.ky10.x86_64
python2-gobject-base-3.36.1-2.ky10.x86_64
python2-help-2.7.18-1.se.04.ky10.noarch
python2-mysqldb-1.3.12-8.ky10.x86_64
python2-pyxdg-0.26-2.ky10.noarch
python2-setuptools-44.1.1-1.ky10.noarch
python2-six-1.15.0-2.ky10.noarch
python2-tkinter-2.7.18-1.se.04.ky10.x86_64
python2-tools-2.7.18-1.se.04.ky10.x86_64
python3-3.7.9-18.se.03.ky10.x86_64
python3-anytree-2.8.0-1.ky10.noarch
python3-asn1crypto-1.4.0-1.ky10.noarch
python3-audit-3.0.6.se.02.ky10.x86_64
python3-babel-2.8.0-3.ky10.noarch
python3-bind-9.11.21-10.ky10.noarch
python3-blivet-3.2.2-8.p04.ky10.noarch
python3-blockdev-2.24-8.ky10.x86_64
python3-bson-3.9.0-4.ky10.x86_64
python3-bytesize-2.4.1.ky10.x86_64
python3-cairo-1.19.1-1.ky10.x86_64
python3-cffi-1.14.1-1.ky10.x86_64
python3-chardet-3.0.4-8.ky10.noarch
python3-click-7.0-2.ky10.noarch
python3-configobj-5.0.6-17.ky10.noarch
python3-configobj-1.1.1-17-2.ky10.noarch

```

寻找的方法可以用 dnf repoquery 命令。下图是寻找所有 pam 相关的 python 包：

```

[root@localhost ~]# dnf repoquery '*python3*pam*'
上次元数据过期检查：2:03:02 前，执行于 2022年11月10日 星期四 09时54分02秒。
python3-PyPAM-0:0.5.0-2.ky10.x86_64
python3-pam-0:1.8.4-1.1.ky10.noarch
[root@localhost ~]#

```

找到包名以后，通过 `dnf install` 安装即可。

如何查看已安装软件的 API 接口

在命令行执行 `python3` 命令（或者 `python2`），进入 `python3` 命令行环境，运行 `help()`，进入如下画面：

```
[root@localhost ~]# python3
Python 3.7.9 (default, Jun 10 2022, 11:25:35)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics". Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> |
```

运行 `modules` 命令，列出系统中安装的 `python3.7` 开发模块：

```
help> modules

Please wait a moment while I gather a list of all available modules...

IPy                array                hawkey              resource
OpenSSL            asn1crypto           heapq               rlcompleter
  future__         ast                  hmac                rpm
  abc              asynchat             html               rtplib
  ast              asyncio             http               rtplib_fb
  asyncio          asyncore             idlelib            runpy
  audit            atexit              idna               sched
  bisect           atune_collector     imaplib            script
  blake2           audioop             imghdr            secrets
  bootlocale       audit              imp                select
  bz2              auparse             importlib          selectors
  cffi_backend     babel               inspect            selinux
  codecs           base64              io                 semanage
  codecs_cn        bdb                 ipaddress          seobject
  codecs_hk        binascii            itertools          sepolgen
  codecs_iso2022   binhex             itsdangerous       sepolgen
  codecs_jp        bios               jinja2             serial
  codecs_kr        bisect              json               setools
  codecs_tw        bootloader          jsonpatch          setuptools
  collections       builtins            jsonpointer        shelve
  collections_abc  bz2                 jsonschema         shlex
  compat_pickle    cProfile            jwt                shutil
  compression      calendar           kerberos            signal
  contextvars      cffi               kernel_config      site
  crypt            cgi                 keyword            six
  csv              cgib                kmod               smtpd
  ctypes           chardet             koji               smtpplib
  ctypes_test      chunk              koji_cli           sndhdr
  curses           click               lib2to3            socket
  curses_panel     cloudinit           libcomps           socketserver
  datetime         cmath              libdnf             socks
  dbm              cmd                 linecache          sockshandler
  dbus_bindings    cmmv               locale             spwd
  dbus_glib_bindings code              logging            sqlite3
  decimal          codecs             lsm                sre_compile
  dummy_thread     codeop             lyndbusd           sre_constants
```

假如需要查看模块 json 的开发接口，输入 json 回车可以打开帮助内容。翻页键控制上下浏览，q 按键退出。

```
Enter any module name to get more help. Or, type "modules spam" to search
for modules whose name or summary contain the string "spam".

help> json

Help on package json:

NAME
    json

MODULE REFERENCE
    https://docs.python.org/3.7/library/json

    The following documentation is automatically generated from the Python
    source files. It may be incomplete, incorrect or include features that
    are considered implementation detail and may vary between Python
    implementations. When in doubt, consult the module reference at the
    location listed above.

DESCRIPTION
    JSON (JavaScript Object Notation) <http://json.org> is a subset of
    JavaScript syntax (ECMA-262 3rd edition) used as a lightweight data
    interchange format.

    :mod:`json` exposes an API familiar to users of the standard library
    :mod:`marshal` and :mod:`pickle` modules. It is derived from a
    version of the externally maintained simplejson library.

    Encoding basic Python object hierarchies::

        >>> import json
        >>> json.dumps(['foo', {'bar': ('baz', None, 1.0, 2)}])
        '["foo", {"bar": ["baz", null, 1.0, 2]}]'
        >>> print(json.dumps("\foo\bar"))
        "\"foo\\bar\""
        >>> print(json.dumps('\u1234'))
        "\"\\u1234\""
        >>> print(json.dumps('\'\''))
        "\"\\\""
        >>> print(json.dumps({'c': 0, 'b': 0, 'a': 0}, sort_keys=True))
        {"a": 0, "b": 0, "c": 0}
        >>> from io import StringIO
        >>> io = StringIO()
        >>> json.dump(['streaming API'], io)
        >>> io.getvalue()
        '["streaming API"]'

    Compact encoding::

        >>> import json
```

在文档的末尾有 FILE 字段。所列文件是这一篇帮助对应的模块路径。我们可以通过 rpm -qf 查出对应的软件包名称。

```
Bob Ippolito <bob@redhat.com>

FILE
    /usr/lib64/python3.7/json/__init__.py

(END)

[root@localhost ~]# rpm -qf /usr/lib64/python3.7/json/__init__.py
python3-3.7.9-18.el8.x86_64
[root@localhost ~]#
```

五，java 语言——javadoc

麒麟 V10 中 java 相关软件包介绍

仓库中有四百多个 java 方面的软件包，其中的三百多是 javadoc 包。还有少

量的 help 包。由此可以看出麒麟 V10 对 java 开发的文档支持是非常到位的。

```
[root@localhost ~]# dnf repoquery '*javadoc*' | wc -l
上次元数据过期检查: 1:06:20 前, 执行于 2022年11月10日 星期四 15时55分02秒。
347
[root@localhost ~]# dnf repoquery '*java*' | wc -l
上次元数据过期检查: 1:06:24 前, 执行于 2022年11月10日 星期四 15时55分02秒。
468
[root@localhost ~]#
[root@localhost ~]# dnf repoquery '*java*help*' | wc -l
上次元数据过期检查: 1:07:10 前, 执行于 2022年11月10日 星期四 15时55分02秒。
12
[root@localhost ~]# dnf repoquery '*java*help*' | wc -l
```

下图是搜索出来的前几个 javadoc 包。

```
[root@localhost ~]# dnf repoquery '*javadoc*'
上次元数据过期检查: 1:01:57 前, 执行于 2022年11月10日 星期四 15时55分02秒。
HdrHistogram-javadoc-0:2.1.11-1.ky10.noarch
airline-javadoc-0:0.7-1.ky10.noarch
ant-antunit-javadoc-0:1.3-2.ky10.noarch
antlr-maven-plugin-javadoc-0:2.2-1.ky10.noarch
antlr32-javadoc-0:3.2-1.ky10.noarch
aopalliance-javadoc-0:1.0-18.ky10.noarch
apache-commons-beanutils-javadoc-0:1.9.4-1.ky10.noarch
apache-commons-chain-javadoc-0:1.2-1.ky10.noarch
apache-commons-collections4-javadoc-0:4.1-1.ky10.noarch
apache-commons-csv-javadoc-0:1.5-2.ky10.noarch
apache-commons-daemon-javadoc-0:1.0.15-19.p01.ky10.noarch
apache-commons-discovery-javadoc-2:0.5-1.ky10.noarch
apache-commons-el-javadoc-0:1.0-1.ky10.noarch
apache-commons-jci-javadoc-0:1.1-1.ky10.noarch
apache-commons-jexl-javadoc-0:2.1.1-1.ky10.noarch
apache-commons-jxpath-javadoc-0:1.3-1.ky10.noarch
apache-commons-logging-javadoc-0:1.2-17.ky10.noarch
apache-commons-math-javadoc-0:3.4.1-1.ky10.noarch
apache-commons-ognl-javadoc-0:3.0.2-1.ky10.noarch
apache-commons-validator-javadoc-0:1.5.0-2.ky10.noarch
apache-mime4j-javadoc-0:0.8.1-1.ky10.noarch
apache-mina-javadoc-0:2.0.21-2.ky10.noarch
```

javadoc 使用方法

通过 dnf install 安装需要的 javadoc 或者 help 包, 然后用浏览器打开对应 html

格式的 help 文档即可。比如需要 xml 方面的, 操作截图如下:

```
[root@localhost ~]# dnf repoquery '*xml*javadoc*'
上次元数据过期检查: 1:13:17 前, 执行于 2022年11月10日 星期四 15时55分02秒。
cookxml-javadoc-0:3.0.2-1.ky10.noarch
decentxml-javadoc-0:1.4-1.ky10.noarch
jackson-dataformat-xml-javadoc-0:2.9.8-1.ky10.noarch
pentaho-libxml-javadoc-0:1.1.3-18.ky10.noarch
simple-xml-javadoc-0:2.7.1-1.ky10.noarch
xml-maven-plugin-javadoc-0:1.0.2-1.ky10.noarch
xml-security-javadoc-0:2.0.6-1.ky10.noarch
xmlbeans-javadoc-0:2.6.0-2.ky10.noarch
xmlbeans-maven-plugin-javadoc-0:2.3.3-1.ky10.noarch
xmlrpc-javadoc-1:3.1.3-2.ky10.noarch
[root@localhost ~]#
```



```
[root@localhost ~]# dnf install simple-xml-javadoc-0:2.7.1-1.ky10.noarch
上次元数据过期检查: 1:14:31 前, 执行于 2022年11月10日 星期四 15时55分02秒。
依赖关系解决。

=====
Package                                Architecture                          Version
=====
安装:
simple-xml-javadoc                      noarch                                2.7.1-1.ky10

事务概要
-----
安装 1 软件包

总下载: 114 k
安装大小: 1.6 M
确定吗? [y/N]: y
下载软件包:
simple-xml-javadoc-2.7.1-1.ky10.noarch.rpm
-----
总计
运行事务检查
事务检查成功。
运行事务测试
事务测试成功。
运行事务
准备中:
安装      : simple-xml-javadoc-2.7.1-1.ky10.noarch
验证      : simple-xml-javadoc-2.7.1-1.ky10.noarch

已安装:
simple-xml-javadoc-2.7.1-1.ky10.noarch

完毕!
[root@localhost ~]#
```

```
[root@localhost ~]# rpm -ql simple-xml-javadoc-2.7.1-1.ky10.noarch
/usr/share/javadoc/simple-xml
/usr/share/javadoc/simple-xml/allclasses-frame.html
/usr/share/javadoc/simple-xml/allclasses-noframe.html
/usr/share/javadoc/simple-xml/constant-values.html
/usr/share/javadoc/simple-xml/deprecated-list.html
/usr/share/javadoc/simple-xml/help-doc.html
/usr/share/javadoc/simple-xml/index-all.html
/usr/share/javadoc/simple-xml/index.html
/usr/share/javadoc/simple-xml/org
/usr/share/javadoc/simple-xml/org/simpleframework
/usr/share/javadoc/simple-xml/org/simpleframework/xml
/usr/share/javadoc/simple-xml/org/simpleframework/xml/Attribute.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/Default.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/DefaultType.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/Element.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/ElementArray.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/ElementList.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/ElementListUnion.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/ElementMap.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/ElementMapUnion.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/ElementUnion.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/Namespace.html
/usr/share/javadoc/simple-xml/org/simpleframework/xml/NamespaceList.html
```

file:///usr/share/javadoc/simple-xml/index.html

OVERVIEW PACKAGE **CLASS** TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: FIELD | REQUIRED | OPTIONAL DETAIL: FIELD | ELEMENT

org.simpleframework.xml

Annotation Type ElementList

@Retention(value=RUNTIME)
public @interface **ElementList**

The ElementList annotation represents a method or field that is a Collection for storing entries. The collection object deserialized is typically of the same type as the field. However, a class attribute can be used to override the field type, however the type must be assignable.

```
<list class="java.util.ArrayList">
  <entry name="one"/>
  <entry name="two"/>
  <entry name="three"/>
</list>
```

file:///usr/share/javadoc/simple-xml/org/simpleframework/xml/strategy/package-frame.html

六，内核驱动开发

驱动程序 kernel-devel 包是内核源码生成的、专门用于内核模块开发，包括

驱动模块等，kernel-devel 包含了内核相关 API 和 makefile 等，开发内核模块前需要安装这个包。关于 kernel 和相关 API 的介绍，可[参考麒麟 V10 仓库中 kernel-doc 包](#)。

[kernel-doc](#) 按照模块功能组织的，非常贴合实际的应用场景。具体如下表：

文档模块	内容说明
Driver Basics	驱动程序基础
Device drivers infrastructure	设备驱动架构基础构建
Device Power Management	设备电源管理，主要驱动程序的待机休眠操作。
The Common Clk Framework	通用时钟框架
Bus-Independent Device Accesses	非总线方式设备访问，包括内存映射和 port 地址空间。
Device connections	非父子关系的设备互联
Buffer Sharing and Synchronization	dms-buf 子系统提供的缓存共享和同步，以及 fence 机制。
Device links	上下游设备依赖关系管理
Message-based devices	特定协议的设备驱动使用 MPT 消息机制驱动
Sound Devices	声音
Frame Buffer Library	图形 FB 驱动
Voltage and current regulator API	电压电流调节
Industrial I/O	数模转换等工业设备的输入输出
Input Subsystem	输入设备驱动
Linux USB API	USB 设备
PCI Support Library	PCI 总线操作接口
PCI Hotplug Support Library	PCI 热插拔
Serial Peripheral Interface (SPI)	SPI 总线
I2C and SMBus Subsystem	I2C 总线和 SMBus
High Speed Synchronous Serial Interface (HSI)	高速同步串行接口 HSI
Error Detection And Correction (EDAC) Devices	EDAC 子系统，主要是内存控制器的错误探测
SCSI Interfaces Guide	SCSI 存储协议
libATA Developer's Guide	ATA 存储协议
target and iSCSI Interfaces Guide	网络存储的 iSCSI 协议
MTD NAND Driver Programming Interface	MTD 存储驱动
Parallel Port Devices	并口
16x50 UART Driver	8250 等 16x50 异步串口控制器驱动
Pulse-Width Modulation (PWM)	PWM 脉冲信号调制
W1: Dallas' 1-wire bus	W1 单总线控制
RapidIO Subsystem Guide	嵌入式内部互联的 RapidIO 芯片
Writing s390 channel device drivers	IBM S390 架构上的设备驱动
VME Device Drivers	嵌入式 VME 总线设备驱动

Linux 802.11 Driver Developer's Guide	无线 80211 设备驱动
The Userspace I/O HOWTO	工业 IO 板卡的 UIO 驱动
Linux Firmware API	Linux 固件开发接口
PINCTRL (PIN CONTROL) subsystem	芯片管脚驱动
General Purpose Input/Output (GPIO)	GPIO 接口设备驱动
Miscellaneous Devices	其他设备
DMAEngine documentation	DMA 引擎
Linux kernel SLIMbus support	SLIM 总线设备
SoundWire Documentation	SoundWire 声音接口设备
FPGA Subsystem	FPGA 板卡设备

文档模块中一般都含有相关驱动开发的通用性指南。如下图是 ‘driver basics’ 中的一个 API 接口 `wq_has_sleeper`。接口名称用蓝色标识是通用的做法，也就是说文档中 API 接口名称都是用这种颜色标记。

`bool wq_has_sleeper(struct wait_queue_head * wq_head)`

check if there are any waiting processes

Parameters

`struct wait_queue_head * wq_head`
wait queue head

Description

Returns true if wq_head has waiting processes

Please refer to the comment for waitqueue_active.

除了 API 接口名称，一个接口展示段落一般包含函数的参数说明（Parameters）、或者结构体的成员描述。Description 字段是对该 API 接口功能的描述。

附 录

1 应用编程接口函数说明

1.1 字符操作接口

isalnum 函数

表头文件: `#include <ctype.h>`

函数原型: `int isalnum (int c)`

函数说明: 检查参数 `c` 是否为英文字母或阿拉伯数字, 在标准 `c` 中相当于使用 “`isalpha(c)||isdigit(c)`” 做测试。

参数说明: 参数 `c` 可为任意字符。

返回值: 若参数 `c` 为字母或数字, 则返回 `TRUE`, 否则返回 `NULL(0)`。

附加说明: 此为宏定义, 非真正函数。

isalpha 函数

表头文件: `#include <ctype.h>`

函数原型: `int isalpha (int c)`

函数说明: 检查参数 `c` 是否为英文字母, 在标准 `c` 中相当于使用 “`isupper(c) || islower(c)`” 做测试。

参数说明: 参数 `c` 可为任意字符。

返回值: 若参数 `c` 为英文字母, 则返回 `TRUE`, 否则返回 `NULL(0)`。

附加说明: 此为宏定义, 非真正函数。

isascii 函数

表头文件: `#include <ctype.h>`

函数原型: `int isascii(int c)`

函数说明: 检查参数 `c` 是否为 ASCII 码字符, 也就是判断 `c` 的范围是否在 0 到 127 之间。

返回值: 若参数 `c` 为 ASCII 码字符, 则返回 `TRUE`, 否则返回 `NULL(0)`。

附加说明: 此为宏定义, 非真正函数。

isctrl 函数

表头文件: `#include <ctype.h>`

函数原型: `int isctrl(int c)`

函数说明: 检查参数 `c` 是否为 ASCII 控制码, 也就是判断 `c` 的范围是否在 0 到 30 之间。

返回值: 若参数 `c` 为 ASCII 控制码, 则返回 `TRUE`, 否则返回 `NULL(0)`。

附加说明: 此为宏定义, 非真正函数。

isdigit 函数

表头文件: `#include <ctype.h>`

函数原型: `int isdigit(int c)`

函数说明: 检查参数 `c` 是否为阿拉伯数字 0 到 9。

返回值: 若参数 `c` 为阿拉伯数字, 则返回 `TRUE`, 否则返回 `NULL(0)`。

附加说明： 此为宏定义，非真正函数。

isgraph 函数

表头文件： #include <ctype.h>

函数原型： int isgraph(int c)

函数说明： 检查参数 c 是否为可打印字符，若 c 所对映的 ASCII 码可打印，且非空格字符则返回 TRUE。

返回值： 若参数 c 为可打印字符且非空格，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

islower 函数

表头文件： #include <ctype.h>

函数原型： int islower(int c)

函数说明： 检查参数 c 是否为小写英文字母。

返回值： 若参数 c 为小写英文字母，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

isprint 函数

表头文件： #include <ctype.h>

函数原型： int isprint(int c);

函数说明： 检查参数 c 是否为可打印字符，若 c 所对映的 ASCII 码可打印，其中包含空格字符，则返回 TRUE。

返回值： 若参数 c 为可打印字符，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

isspace 函数

表头文件： #include <ctype.h>

函数原型： int isspace(int c)

函数说明： 检查参数 c 是否为空格字符，也就是判断是否为空格(' ')、定位字符('\t')、CR('\r')、换行('\n')、垂直定位字符('\v')或翻页('\f')的情况。

返回值： 若参数 c 为空格字符，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

ispunct 函数

表头文件： #include <ctype.h>

函数原型： int ispunct(int c)

函数说明： 检查参数 c 是否为标点符号或特殊符号。

返回值： 若参数 c 为标点符号或特殊符号，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

isupper 函数

表头文件： #include <ctype.h>

函数原型： int isupper(int c)

函数说明： 检查参数 c 是否为大写英文字母。

返回值： 若参数 c 为大写英文字母，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

isxdigit 函数

表头文件： #include <ctype.h>

函数原型： int isxdigit (int c)

函数说明： 检查参数 c 是否为 16 进制数字。

返回值： 若参数 c 为 16 进制数字,即 0123456789ABCDEF，则返回 TRUE，否则返回 NULL(0)。

附加说明： 此为宏定义，非真正函数。

1.2 字符串操作接口

atof 函数

表头文件： #include <stdlib.h>

函数原型： double atof(const char *nptr)

函数说明： atof()会扫描参数 nptr 字符串，跳过前面的空格字符，直到遇上数字或正负符号才开始做转换，而再遇到非数字或字符串结束时('\0')才结束转换，并将结果返回。参数 nptr 字符串可包含正负号、小数点或 E(e)来表示指数部分，如 123.456 或 123e-2。

返回值： 返回转换后的浮点型数。

附加说明： atof()与使用 strtod(nptr,(char**)NULL)结果相同。

atoi 函数

表头文件： #include <stdlib.h>

函数原型： int atoi(const char *nptr)

函数说明： atoi()会扫描参数 nptr 字符串，跳过前面的空格字符，直到遇上数字或正负符号才开始做转换，而再遇到非数字或字符串结束时('\0')才结束转换，并将结果返回。

返回值： 返回转换后的整型数。

附加说明： atoi()与使用 strtol(nptr, (char**)NULL, 10)；结果相同。

atol 函数

表头文件： #include <stdlib.h>

函数原型： long atol(const char *nptr)

函数说明： atol()会扫描参数 nptr 字符串，跳过前面的空格字符，直到遇上数字或正负符号才开始做转换，而再遇到非数字或字符串结束时('\0')才结束转换，并将结果返回。

返回值： 返回转换后的长整型数。

附加说明： atol()与使用 strtol(nptr,(char**)NULL,10)；结果相同。

gcvt 函数

表头文件： #include <stdlib.h>

函数原型： char *gcvt(double number, size_t ndigits, char *buf)

函数说明： gcvt()用来将参数 number 转换成 ASCII 码字符串,参数 ndigits 表示显示的位数。gcvt()与 ecvt()和 fcvt()不同的地方在于，gcvt()所转换后的字符串包含小数点或正负符号。若转换成功，转换后的字符串会放在参数 buf 指针所指的空间。

返回值： 返回一字符串指针，此地址即为 buf 指针。

附加说明： 无。

strtod 函数

表头文件: `#include<stdlib.h>`

函数原型: `double strtod(const char *nptr,char **endptr)`

函数说明: `strtod()`会扫描参数 `nptr` 字符串,跳过前面的空格字符,直到遇上数字或正负符号才开始做转换,到出现非数字或字符串结束时(`\0`)才结束转换,并将结果返回。若 `endptr` 不为 `NULL`,则会将遇到不合条件而终止的 `nptr` 中的字符指针由 `endptr` 传回。参数 `nptr` 字符串可包含正负号、小数点或 `E(e)`来表示指数部分。如 `123.456` 或 `123e-2`。

返回值: 返回转换后的浮点型数。

附加说明: 参考 `atof()`。

strtol 函数

表头文件: `#include<stdlib.h>`

函数原型: `long int strtol(const char *nptr,char **endptr,int base)`

函数说明: `strtol()`会将参数 `nptr` 字符串根据参数 `base` 来转换成长整型数。参数 `base` 范围从 2 至 36,或 0。参数 `base` 代表采用的进制方式,如 `base` 值为 10 则采用 10 进制,若 `base` 值为 16 则采用 16 进制等。当 `base` 值为 0 时则是采用 10 进制做转换,但遇到如 `'0x'`前置字符则会使用 16 进制做转换。一开始 `strtol()`会扫描参数 `nptr` 字符串,跳过前面的空格字符,直到遇上数字或正负符号才开始做转换,再遇到非数字或字符串结束时(`\0`)结束转换,并将结果返回。若参数 `endptr` 不为 `NULL`,则会将遇到不合条件而终止的 `nptr` 中的字符指针由 `endptr` 返回。

返回值: 返回转换后的长整型数,否则返回 `ERANGE` 并将错误代码存入 `errno` 中。

附加说明: `ERANGE` 指定的转换字符串超出合法范围。

strtoul 函数

表头文件: `#include<stdlib.h>`

函数原型: `unsigned long int strtoul(const char *nptr,char **endptr,int base)`

函数说明: `strtoul()`会将参数 `nptr` 字符串根据参数 `base` 来转换成无符号的长整型数。参数 `base` 范围从 2 至 36,或 0。参数 `base` 代表采用的进制方式,如 `base` 值为 10 则采用 10 进制,若 `base` 值为 16 则采用 16 进制数等。当 `base` 值为 0 时则是采用 10 进制做转换,但遇到如 `'0x'`前置字符则会使用 16 进制做转换。一开始 `strtoul()`会扫描参数 `nptr` 字符串,跳过前面的空格字符串,直到遇上数字或正负符号才开始做转换,再遇到非数字或字符串结束时(`\0`)结束转换,并将结果返回。若参数 `endptr` 不为 `NULL`,则会将遇到不合条件而终止的 `nptr` 中的字符指针由 `endptr` 返回。

返回值: 返回转换后的长整型数,否则返回 `ERANGE` 并将错误代码存入 `errno` 中。

附加说明: `ERANGE` 指定的转换字符串超出合法范围。

toascii 函数

表头文件: `#include<ctype.h>`

函数原型: `int toascii(int c)`

函数说明: `toascii()`会将参数 `c` 转换成 7 位的 `unsigned char` 值,第八位则会被清除,此字符即会被转成 ASCII 码字符。

返回值: 将转换成功的 ASCII 码字符值返回。

tolower 函数

表头文件: `#include<stdlib.h>`

函数原型: `int tolower(int c)`

函数说明: 若参数 `c` 为大写字母则将该对应的小写字母返回。

返回值: 返回转换后的小写字母, 若不须转换则将参数 `c` 值返回。

附加说明: 无。

`toupper` 函数

表头文件: `#include<ctype.h>`

函数原型: `int toupper(int c)`

函数说明: 若参数 `c` 为小写字母则将该对映的大写字母返回。

返回值: 返回转换后的大写字母, 若不须转换则将参数 `c` 返回。

附加说明: 无。

1.3 内存控制接口

`calloc` 函数

头文件: `#include<stdlib.h>`

函数原型: `void *calloc(size_t nmemb, size_t size)`

函数说明: `calloc()`用来配置 `nmemb` 个相邻的内存单位, 每一单位的大小为 `size`, 并返回 指向第一个元素的指针。这和使用下列的方式效果相同:`malloc(nmemb*size)`;不过, 在利用 `calloc()`配置内存时会将内存内容初始化为 0。

返回值: 若配置成功则返回一指针, 失败则返回 `NULL`。

`free` 函数

头文件: `#include<stdlib.h>`

函数原型: `void free(void*ptr)`

函数说明: 参数 `ptr` 为指向先前由 `malloc()`、`calloc()`或 `realloc()`所返回的内存指针。调用 `free()` 后 `ptr` 所指的内存空间便会被收回。假若参数 `ptr` 所指的内存空间已被收回或是未知的内存 地址, 则调用 `free()`可能会有无法预期的情况发生。若参数 `ptr` 为 `NULL`, 则 `free()`不会有任 何作用。

返回值: 无。

`getpagesize` 函数

头文件: `#include<unistd.h>`

函数原型: `size_t getpagesize(void)`

函数说明: 返回一内存页的大小, 单位为字节 (byte)。此为系统的分页大小, 不一定会 和硬件分页大小相同。

返回值: 内存分页大小。

附加说明: 在 Intelx86 上其返回值应为 4096bytes。

`mmap` 函数

头文件: `#include<unistd.h>`

`#include<sys/mman.h>`

函数原型: `void *mmap(void *start,size_t length,int prot,int flags,int fd,off_toffset)`

函数说明: `mmap()`用来将某个文件内容映射到内存中, 对该内存区域的存取即是直接对 该文件内容的读

写。参数 `start` 指向欲对应的内存起始地址，通常设为 `NULL`，代表让系统自动选定地址，对应成功后该地址会返回。

参数 `length` 代表将文件中多大的部分对应到内存。

参数 `prot` 代表映射区域的保护方式有下列组合：

`PROT_EXEC` 映射区域可被执行

`PROT_READ` 映射区域可被读取

`PROT_WRITE` 映射区域可被写入

`PROT_NONE` 映射区域不能存取

参数 `flags` 会影响映射区域的各种特性：

`MAP_FIXED` 如果参数 `start` 所指的地址无法成功建立映射时，则放弃映射，不对地址做修正。通常不鼓励用此旗标。

`MAP_SHARED` 对映射区域的写入数据会复制回文件内，而且允许其他映射该文件的进程共享。

`MAP_PRIVATE` 对映射区域的写入操作会产生一个映射文件的复制，即私人的“写入时复制”（`copy onwrite`）对此区域作的任何修改都不会写回原来的文件内容。`MAP_ANONYMOUS` 建立匿名映射。此时会忽略参数 `fd`，不涉及文件，而且映射区域无法和其他进程共享。

`MAP_DENYWRITE` 只允许对映射区域的写入操作，其他对文件直接写入的操作将会被拒绝。

`MAP_LOCKED` 将映射区域锁定住，这表示该区域不会被置换（`swap`）。在调用 `mmap()`时必须指定 `MAP_SHARED` 或 `MAP_PRIVATE`。

参数 `fd` 为 `open()`返回的文件描述词，代表欲映射到内存的文件。

参数 `offset` 为文件映射的偏移量，通常设置为 0，代表从文件最前方开始对应，`offset` 必须是分页大小的整数倍。

返回值：若映射成功则返回映射区的内存起始地址，否则返回 `MAP_FAILED(-1)`，错误原因存于 `errno` 中。

错误代码：

`EBADF` 参数 `fd` 不是有效的文件描述词

`EACCES` 存取权限有误。如果是 `MAP_PRIVATE` 情况下文件必须可读，使用 `MAP_SHARED` 则要有 `PROT_WRITE` 以及该文件要能写入。

`EINVAL` 参数 `start`、`length` 或 `offset` 有一个不合法。

`EAGAIN` 文件被锁住，或是有太多内存被锁住。

`ENOMEM` 内存不足。

`munmap` 函数

头文件： `#include<unistd.h>`

`#include<sys/mman.h>`

函数原型： `int munmap(void*start,size_t length)`

函数说明： `munmap()`用来取消参数 `start` 所指的映射内存起始地址，参数 `length` 则是欲取消的内存大小。当进程结束或利用 `exec` 相关函数来执行其他程序时，映射内存会自动解除，但关闭对应的文件描述词时不会解除映射。

返回值：如果解除映射成功则返回 0，否则返回 -1，错误原因存于 `errno` 中错误代码 `EINVAL` 参数 `start` 或 `length` 不合法。

1.4 内存及字符串操作接口

`bcmp` 函数

表头文件: #include<string.h>

函数原型: int bcmp (const void *s1,const void *s2,int n)

函数说明: bcmp()用来比较 s1 和 s2 所指的内存区间前 n 个字节。

返回值: 若参数 s1 和 s2 所指的内存内容都完全相同(包括 n 为 0 的情况)则返回 0 值, 否则返回非零值。

附加说明: 建议使用 memcmp()取代。

bcopy 函数

表头文件: #include <string.h>

函数原型: void bcopy (const void *src,void *dest ,int n)

函数说明: bcopy()与 memcpy()一样都是用来拷贝 src 所指的内存内容前 n 个字节到 dest 所指的地址, 不过参数 src 与 dest 在传给函数时是相反的位置。

返回值: 无。

附加说明: 建议使用 memcpy()取代。

bzero 函数

表头文件: #include<string.h>

函数原型: void bzero(void *s,int n)

函数说明: bzero()会将参数 s 所指的内存区域前 n 个字节, 全部设为零值。相当于调用 memset((void*)s,0,size_tn)。

返回值: 无。

附加说明: 建议使用 memset 取代。

index 函数

表头文件: #include<string.h>

函数原型: char * index(const char *s, int c)

函数说明: index()用来找出参数 s 字符串中第一个出现的参数 c 的地址, 然后将该字符出现的地址返回。字符串结束字符(NULL)也视为字符串一部分。

返回值: 如果找到指定的字符则返回该字符所在地址, 否则返回 0。

memccpy 函数

表头文件: #include<string.h>

函数原型: void * memccpy(void *dest, const void * src, int c,size_t n)

函数说明: memccpy()用来拷贝 src 所指的内存内容前 n 个字节到 dest 所指的地址上。与 memcpy()不同的是, memccpy()会在复制时检查参数 c 是否出现, 若是则返回 dest 中值为 c 的下一个字节地址。

返回值: 返回指向 dest 中值为 c 的下一个字节指针。返回值为 0 表示在 src 所指内存前 n 个字节中没有值为 c 的字节。

memchr 函数

表头文件: #include<string.h>

函数原型: void * memchr(const void *s,int c,size_t n)

函数说明: memchr()从头开始搜寻 s 所指的内存内容前 n 个字节, 直到发现第一个值为 c 的字节, 则返回指向该字节的指针。

返回值: 如果找到指定的字节则返回该字节的指针, 否则返回 0。

memcmp 函数

表头文件: #include<string.h>

函数原型: int memcmp (const void *s1,const void *s2,size_t n)

函数说明: memcmp()用来比较 s1 和 s2 所指的内存区间前 n 个字符。字符串大小的比较是以 ASCII 码表上的顺序来决定,次顺序亦为字符的值。

返回值: 若参数 s1 和 s2 所指的内存内容都完全相同则返回 0 值。s1 若大于 s2 则返回大于 0 的值。s1 若小于 s2 则返回小于 0 的值。

memcpy 函数

表头文件: #include<string.h>

函数原型: void * memcpy (void * dest ,const void *src, size_t n)

函数说明: memcpy()用来拷贝 src 所指的内存内容前 n 个字节到 dest 所指的内存地址上。与 strcpy()不同的是,memcpy()会完整的复制 n 个字节,不会因为遇到字符串结束'\0'而结束。

返回值: 返回指向 dest 的指针。

附加说明: 指针 src 和 dest 所指的内存区域不可重叠。

memmove 函数

表头文件: #include<string.h>

函数原型: void * memmove(void *dest,const void *src,size_t n)

函数说明: memmove()与 memcpy()一样都是用来拷贝 src 所指的内存内容前 n 个字节到 dest 所指的地址上。不同的是,当 src 和 dest 所指的内存区域重叠时,memmove()仍然可以正确的处理,不过执行效率上会比使用 memcpy()略慢些。

返回值: 返回指向 dest 的指针。

附加说明: 指针 src 和 dest 所指的内存区域可以重叠。

memset 函数

表头文件: #include<string.h>

函数原型: void * memset (void *s ,int c, size_t n)

函数说明: memset()会将参数 s 所指的内存区域前 n 个字节以参数 c 填入,然后返回指向 s 的指针。在编写程序时,若需要将某一数组作初始化,memset()会相当方便。

返回值: 返回指向 s 的指针。

附加说明: 参数 c 虽声明为 int, 但必须是 unsigned char , 所以范围在 0 到 255 之间。

rindex 函数

表头文件: #include<string.h>

函数原型: char * rindex(const char *s,int c)

函数说明: rindex()用来找出参数 s 字符串中最后一个出现的参数 c 的地址,然后将该字符出现的地址返回。字符串结束字符(NULL)也视为字符串一部分。

返回值: 如果找到指定的字符则返回该字符所在的地址,否则返回 0。

strcasecmp 函数

表头文件: #include<string.h>

函数原型: int strcasecmp (const char *s1, const char *s2)

函数说明: strcasecmp()用来比较参数 s1 和 s2 字符串,比较时会自动忽略大小写的差异。

返回值: 若参数 s1 和 s2 字符串相同则返回 0。s1 长度大于 s2 长度则返回大于 0 的值,s1 长度若小于

s2 长度则返回小于 0 的值。

strcat 函数

表头文件: #include <string.h>

函数原型: char *strcat (char *dest,const char *src)

函数说明: strcat()会将参数 src 字符串拷贝到参数 dest 所指的字符串尾。第一个参数 dest 要有足够的空间来容纳要拷贝的字符串。

返回值: 返回参数 dest 的字符串起始地址。

strchr 函数

表头文件: #include<string.h>

函数原型: char * strchr (const char *s,int c)

函数说明: strchr()用来找出参数 s 字符串中第一个出现的参数 c 的地址,然后将该字符出现的地址返回。

返回值: 如果找到指定的字符则返回该字符所在地址,否则返回 0。

strcmp 函数

表头文件: #include<string.h>

函数原型: int strcmp(const char *s1,const char *s2)

函数说明: strcmp()用来比较参数 s1 和 s2 字符串。字符串大小的比较是以 ASCII 码表上的顺序来决定,此顺序亦为字符的值。

返回值: 若参数 s1 和 s2 字符串相同则返回 0。s1 若大于 s2 则返回大于 0 的值。s1 若小于 s2 则返回小于 0 的值。

strcoll 函数

表头文件: #include<string.h>

函数原型: int strcoll(const char *s1, const char *s2)

函数说明: strcoll()会依环境变量 LC_COLLATE 所指定的文字排列次序来比较 s1 和 s2 字符串。

返回值: 若参数 s1 和 s2 字符串相同则返回 0。s1 若大于 s2 则返回大于 0 的值。s1 若小于 s2 则返回小于 0 的值。

附加说明: 若 LC_COLLATE 为"POSIX"或"C",则 strcoll()与 strcmp()作用完全相同。

strcpy 函数

表头文件: #include<string.h>

函数原型: char *strcpy(char *dest,const char *src)

函数说明: strcpy()会将参数 src 字符串拷贝至参数 dest 所指的地址。

返回值: 返回参数 dest 的字符串起始地址。

附加说明: 如果参数 dest 所指的内存空间不够大,可能会造成缓冲溢出(buffer Overflow)的错误情况,在编写程序时请特别留意,或者用 strncpy()来取代。

strcspn 函数

表头文件: #included<string.h>

函数原型: size_t strcspn (const char *s,const char * reject)

函数说明: strcspn()从参数 s 字符串的开头计算连续的字符,而这些字符都完全不在参数 reject 所指的字符串中。简单地说,若 strcspn()返回的数值为 n,则代表字符串 s 开头连续有 n 个字符都不含字符串 reject 内的字符。

返回值: 返回字符串 s 开头连续不含字符串 reject 内的字符数目。

strdup 函数

表头文件: #include<string.h>

函数原型: char * strdup(const char *s);

函数说明: strdup()会先用 malloc()配置与参数 s 字符串相同的空间大小,然后将参数 s 字符串的内容复制到该内存地址,然后把该地址返回。该地址最后可以利用 free()来释放。

返回值: 返回一字符串指针,该指针指向复制后的新字符串地址。若返回 NULL 表示内存不足。

1.4.20 strlen 函数

表头文件: #include<string.h>

函数原型: size_t strlen (const char *s)

函数说明: strlen()用来计算指定的字符串 s 的长度,不包括结束字符"\0"。

返回值: 返回字符串 s 的字符数。

strncasecmp 函数

表头文件: #include<string.h>

函数原型: int strncasecmp(const char *s1,const char *s2,size_t n)

函数说明: strncasecmp()用来比较参数 s1 和 s2 字符串第一个前 n 个字符,比较时会自动忽略大小写的差异。

返回值: 若参数 s1 和 s2 字符串相同则返回 0。s1 若大于 s2,则返回大于 0 的值,s1 若小于 s2,则返回小于 0 的值。

strncat 函数

表头文件: #include <string.h>

函数原型: char *strncat(char *dest,const char *src,size_t n)

函数说明: strncat()会将参数 src 字符串拷贝 n 个字符到参数 dest 所指的字符串尾。第一个参数 dest 要有足够的空间来容纳要拷贝的字符串。

返回值: 返回参数 dest 的字符串起始地址。

strncpy 函数

表头文件: #include<string.h>

函数原型: char *strncpy(char *dest,const char *src,size_t n)

函数说明: strncpy()会将参数 src 字符串拷贝前 n 个字符至参数 dest 所指的地址。

返回值: 返回参数 dest 的字符串起始地址。

strpbrk 函数

表头文件: #include <string.h>

函数原型: char *strpbrk(const char *s,const char *accept)

函数说明: strpbrk()用来找出参数 s 字符串中最先出现存在参数 accept 字符串中的任意字符。

返回值: 如果找到指定的字符则返回该字符所在地址,否则返回 0。

strrchr 函数

表头文件: #include<string.h>

函数原型: char *strrchr(const char *s, int c)

函数说明: strrchr()用来找出参数 s 字符串中最后一个出现的参数 c 地址,然后将该字符出现的地址返回。

返回值: 如果找到指定的字符则返回该字符所在地址,否则返回 0。

strspn 函数

表头文件: #include<string.h>

函数原型: size_t strspn (const char *s,const char * accept)

函数说明: strspn()从参数 s 字符串的开头计算连续的字符,而这些字符都完全是 accept 所指字符串中的字符。简单的说,若 strspn()返回的数值为 n,则代表字符串 s 开头连续有 n 个字符都是属于字符串 accept 内的字符。

返回值: 返回字符串 s 开头连续包含字符串 accept 内的字符数目。

strstr 函数

表头文件: `#include<string.h>`

函数原型: `char *strstr(const char *haystack,const char *needle)`

函数说明: `strstr()`会从字符串 `haystack` 中搜寻字符串 `needle`, 并将第一次出现的地址返回。

返回值: 返回指定字符串第一次出现的地址, 否则返回 0。

strtok 函数

表头文件: `#include<string.h>`

函数原型: `char *strtok(char *s,const char *delim)`

函数说明: `strtok()`用来将字符串分割成一个个片段。参数 `s` 指向欲分割的字符串, 参数 `delim` 则为分割字符串, 当 `strtok()`在参数 `s` 的字符串中发现到参数 `delim` 的分割字符时则会将该字符改为'\0' 字符。在第一次调用时, `strtok()`必需给予参数 `s` 字符串, 往后的调用则将参数 `s` 设置成 `NULL`。每次调用成功则返回下一个分割后的字符串指针。

返回值: 返回下一个分割后的字符串指针, 如果已无从分割则返回 `NULL`。

1.5 日期和时间操作接口

asctime 函数

头文件:

`#include<time.h>`

函数原型:

`char *asctime(const struct tm *timeptr)`

函数说明:

`asctime()`将分解的时间值 `tm` 转换成与 `ctime()`相同的空终止字符串。返回值指向静态分配的字符串, 该字符串可能会被后续调用的任何日期、时间函数覆盖。`asctime()`函数将字符串存储在用户提供的缓冲区中, 缓冲区至少有 26 字节的空间。

返回值: 返回一字符串表示目前当地的时间日期, 在检测到错误时返回 0。

ctime 函数

头文件:

`#include<time.h>`

函数原型:

`char *ctime(const time_t *timep)`

函数说明:

`ctime()`将参数 `timep` 所指的 `time_t` 结构中的信息转换成真实世界所使用的时间日期表示方法, 然后将结果以字符串形态返回。此函数已经由时区转换成当地时间, 字符串格式为“Wed Jun 30 21 :49 :08 1993\n”。若再调用相关的时间日期函数, 此字符串可能会被破坏。

返回值: 返回一字符串表示目前当地的时间日期。

gettimeofday 函数

头文件:

`#include <sys/time.h>`

函数原型:

`int gettimeofday ( struct timeval * tv , struct timezone * tz )`

函数说明:

gettimeofday()会把目前的时间有 tv 所指的结构返回, 当地时区的信息则放到 tz 所指的结构中。

timeval 结构定义为:

```
struct timeval{
    long tv_sec; /*秒*/
    long tv_usec; /*微秒*/
};
```

timezone 结构定义为:

```
struct timezone{
    int tz_minuteswest; /*和 Greenwich 时间差了多少分钟*/
    int tz_dsttime; /*日光节约时间的状态*/
};
```

上述两个结构都定义在/usr/include/sys/time.h。

tz_dsttime 所代表的状态如下:

```
DST_NONE /*不使用*/
DST_USA /*美国*/
DST_AUST /*澳洲*/
DST_WET /*西欧*/
DST_MET /*中欧*/
DST_EET /*东欧*/
DST_CAN /*加拿大*/
DST_GB /*大不列颠*/
DST_RUM /*罗马尼亚*/
DST_TUR /*土耳其*/
DST_AUSTALT /*澳洲 (1986 年以后)*/
```

返回值: 成功则返回 0, 失败返回 -1, 错误代码存于 errno。附加说明: EFAULT 指针 tv 和 tz 所指的内存空间超出存取权限。

gmtime 函数

头文件:

```
#include<time.h>
```

函数原型:

```
struct tm*gmtime(const time_t*timep)
```

函数说明:

gmtime()将参数 timep 所指的 time_t 结构中的信息转换成真实世界所使用的时间日期表示方法, 然后将结果由结构 tm 返回。

结构 tm 的定义为:

```
struct tm
{
    int tm_sec;
    int tm_min;
    int tm_hour;
    int tm_mday;
```

```

int tm_mon;
int tm_year;
int tm_wday;
int tm_yday;
int tm_isdst;
};
int tm_sec 代表目前秒数，正常范围为 0-59，但允许至 61 秒。
int tm_min 代表目前分数，范围 0-59。
int tm_hour 从午夜算起的时数，范围为 0-23。
int tm_mday 目前月份的日数，范围 01-31。
int tm_mon
代表目前月份，从一月算起，范围从 0-11。
int tm_year
从 1900 年算起至今的年数。
int tm_wday
一星期的日数，从星期一算起，范围为 0-6。
int tm_yday
从今年 1 月 1 日算起至今的天数，范围为 0-365。
int tm_isdst
日光节约时间的旗标。
此函数返回的时间日期未经时区转换，而是 UTC 时间。
返回值： 返回结构 tm 代表目前 UTC 时间。

```

localtime 函数

头文件：

```
#include<time.h>
```

函数原型：

```
struct tm *localtime(const time_t *timep)
```

函数说明：

localtime()将参数 timep 所指的 time_t 结构中的信息转换成真实世界所使用的时间日期表示方法，然后将结果由结构 tm 返回。结构 tm 的定义请参考 gmtime()。此函数返回的时间日期已经转换成当地时区。

返回值： 返回结构 tm 代表目前的当地时间。

mktime 函数

头文件：

```
#include<time.h>
```

函数原型：

```
time_t mktime(struct tm *timeptr)
```

函数说明：

mktime()用来将参数 timeptr 所指的 tm 结构数据转换成从公元 1970 年 1 月 1 日 0 时 0 分 0 秒算起至今的 UTC 时间所经过的秒数。

返回值： 返回经过的秒数。

settimeofday 函数

头文件:

```
#include<sys/time.h>
```

```
#include<unistd.h>
```

函数原型:

```
int settimeofday ( const struct timeval *tv,const struct timezone *tz)
```

函数说明:

settimeofday()会把目前时间设成由 tv 所指的结构信息, 当地时区信息则设成 tz 所指的结构。详细的说明请参考 gettimeofday()。注意, 只有 root 权限才能使用此函数修改时间。

返回值: 成功则返回 0, 失败返回 -1, 错误代码存于 errno。

附加说明: 错误代码

EPERM

并非由 root 权限调用 settimeofday (), 权限不够。

EINVAL

时区或某个数据是不正确的, 无法正确设置时间。

time 函数

头文件:

```
#include<time.h>
```

函数原型:

```
time_t time(time_t *t)
```

函数说明: 此函数会返回从公元 1970 年 1 月 1 日的 UTC 时间从 0 时 0 分 0 秒算起到现在所经过的秒数。如果 t 并非空指针的话, 此函数也会将返回值: 存到 t 指针所指的内存。

返回值: 成功则返回秒数, 失败则返回((time_t)-1)值, 错误原因存于 errno 中。

1.6 数学函数操作接口

abs 函数

头文件:

```
#include<stdlib.h>
```

函数原型:

```
int abs (int j)
```

函数说明:

abs()用来计算参数 j 的绝对值, 然后将结果返回。

返回值: 返回参数 j 的绝对值结果。

acos 函数

头文件:

```
#include <math.h>
```

函数原型:

```
double acos (double x)
```

函数说明:

acos()用来计算参数 x 的反余弦值, 然后将结果返回。参数 x 范围为 -1 至 1 之间, 超过此范围则会失败。

返回值： 返回 0 至 π 之间的计算结果，单位为弧度，在函数库中角度均以弧度来表示。

错误代码： EDOM 参数 x 超出范围。

附加说明： 使用 GCC 编译时请加入 `-lm`。

asin 函数

头文件：

`#include <math.h>`

函数原型：

`double asin (double x)`

函数说明：

`asin()`用来计算参数 x 的反正弦值，然后将结果返回。参数 x 范围为 -1 至 1 之间，超过此范围则会失败。

返回值： 返回 $-\pi/2$ 至 $\pi/2$ 之间的计算结果。

错误代码： EDOM 参数 x 超出范围。

附加说明： 使用 GCC 编译时请加入 `-lm`。

atan 函数

头文件：

`#include<math.h>`

函数原型：

`double atan(double x)`

函数说明：

`atan()`用来计算参数 x 的反正切值，然后将结果返回。

返回值： 返回 $-\pi/2$ 至 $\pi/2$ 之间的计算结果。

附加说明： 使用 GCC 编译时请加入`-lm`。

atan2 函数

头文件：

`#include<math.h>`

函数原型：

`double atan2(double y,double x)`

函数说明：

`atan2()`用来计算参数 y/x 的反正切值，然后将结果返回。

返回值： 返回 $-\pi/2$ 至 $\pi/2$ 之间的计算结果。

附加说明： 使用 GCC 编译时请加入`-lm`。

ceil 函数

头文件：

`#include <math.h>`

函数原型：

`double ceil (double x)`

函数说明： `ceil()`会返回不小于参数 x 的最小整数值，结果以 `double` 形态返回。

返回值： 返回不小于参数 x 的最小整数值。

附加说明： 使用 GCC 编译时请加入`-lm`。

cos 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double cos(double x)
```

函数说明:

cos()用来计算参数 x 的余弦值,然后将结果返回。

返回值: 返回-1 至 1 之间的计算结果。

附加说明: 使用 GCC 编译时请加入-lm。

cosh 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double cosh(double x)
```

函数说明:

cosh()用来计算参数 x 的双曲线余弦值,然后将结果返回。数学定义式为: $(\exp(x)+\exp(-x))/2$ 。

返回值: 返回参数 x 的双曲线余弦值。

附加说明: 使用 GCC 编译时请加入-lm。

exp 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double exp(double x)
```

函数说明:

exp()用来计算以 e 为底的 x 次方值,即 e^x 值,然后将结果返回。

返回值: 返回 e 的 x 次方计算结果。

附加说明: 使用 GCC 编译时请加入-lm。

frexp 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double frexp( double x, int *exp)
```

函数说明:

frexp()用来将参数 x 的浮点型数切割成底数和指数。底数部分直接返回,指数部分则借参数 exp 指针返回,将返回值:乘以 2 的 exp 次方即为 x 的值。

返回值: 返回参数 x 的底数部分,指数部分则存于 exp 指针所指的地址。

附加说明: 使用 GCC 编译时请加入-lm。

ldexp 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double ldexp(double x,int exp)
```

函数说明:

ldexp()用来将参数 x 乘上 2 的 exp 次方值,即 $x*2^{exp}$ 。

返回值: 返回计算结果。

附加说明: 使用 GCC 编译时请加入`-lm`。

log 函数

头文件:

```
#include <math.h>
```

函数原型:

```
double log (double x)
```

函数说明:

log () 用来计算以 e 为底的 x 对数值,然后将结果返回。

返回值: 返回参数 x 的自然对数值。

错误代码 EDOM 参数 x 为负数,ERANGE 参数 x 为零值,零的对数值无定义。

附加说明: 使用 GCC 编译时请加入`-lm`。

log10 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double log10(double x)
```

函数说明:

log10()用来计算以 10 为底的 x 对数值,然后将结果返回。

返回值: 返回参数 x 以 10 为底的对数值。

代码 EDOM 参数 x 为负数。RANGE 参数 x 为零值,零的对数值无定义。

附加说明:

使用 GCC 编译时请加入`-lm`。

pow 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double pow(double x,double y);
```

函数说明:

pow()用来计算以 x 为底的 y 次方值,即 x^y 值,然后将结果返回。

返回值: 返回 x 的 y 次方计算结果。

错误代码 EDOM 参数 x 为负数且参数 y 不是整数。

附加说明: 使用 GCC 编译时请加入`-lm`。

sin 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double sin(double x)
```

函数说明:

sin () 用来计算参数 x 的正弦值, 然后将结果返回。

返回值: 返回-1 至 1 之间的计算结果。

附加说明: 使用 GCC 编译时请加入-lm。

sinh 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double sinh( double x)
```

函数说明:

sinh()用来计算参数 x 的双曲线正弦值, 然后将结果返回。数学定义式为: $(\exp(x)-\exp(-x))/2$ 。

返回值: 返回参数 x 的双曲线正弦值。

附加说明: 使用 GCC 编译时请加入-lm。

sqrt 函数

头文件:

```
#include<math.h>
```

函数原型:

```
double sqrt(double x)
```

函数说明:

sqrt()用来计算参数 x 的平方根, 然后将结果返回。参数 x 必须为正数。

返回值: 返回参数 x 的平方根值。

错误代码

EDOM 参数 x 为负数。

附加说明: 使用 GCC 编译时请加入-lm。

tan 函数

头文件:

```
#include <math.h>
```

函数原型:

```
double tan(double x)
```

函数说明:

tan()用来计算参数 x 的正切值, 然后将结果返回。

返回值: 返回参数 x 的正切值。

附加说明: 使用 GCC 编译时请加入-lm。

tanh 函数

头文件:

#include<math.h>

函数原型:

double tanh(double x)

函数说明:

tanh()用来计算参数 x 的双曲线正切值,然后将结果返回。数学定义式为: $\sinh(x)/\cosh(x)$ 。

返回值: 返回参数 x 的双曲线正切值。

附加说明: 使用 GCC 编译时请加入-lm。

1.7 用户与组操作接口

endgrent 函数

表头文件: #include<grp.h>

#include<sys/types.h>

函数原型: void endgrent(void)

函数说明: endgrent()用来关闭由 getgrent()所打开的密码文件。

返回值:无。

附加说明: 无。

endpwent 函数

表头文件: #include<pwd.h>

#include<sys/types.h>

函数原型: void endpwent(void)

函数说明: endpwent()用来关闭由 getpwent()所打开的密码文件。

返回值:无。

附加说明: 无。

endutent 函数

表头文件: #include<utmp.h>

函数原型: void endutent(void)

函数说明: endutent()用来关闭由 getutent 所打开的 utmp 文件。

返回值:无。

附加说明:无。

fgetgrent 函数

表头文件: #include<grp.h>

#include<stdio.h>

#include<sys/types.h>

函数原型: struct group * getgrent(FILE * stream)

函数说明:fgetgrent()会从参数 stream 指定的文件读取一行数据,然后以 group 结构将该

数据返回。参数 stream 所指定的文件必须和 /etc/group 相同的格式。group 结构定义请参考 getgrent()。

返回值:返回 group 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

fgetpwent 函数

表头文件: #include<pwd.h>

#include<stdio.h>

#include<sys/types.h>

函数原型: struct passwd *fgetpwent(FILE *stream)

函数说明:

fgetpwent()会从参数 stream 指定的文件读取一行数据,然后以 passwd 结构将

该数据返回。参数 stream 所指定的文件必须和/etc/passwd 相同的格式。passwd 结构定义请参考 getpwent()。

返回值:返回 passwd 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

getegid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: gid_t getegid(void)

函数说明: getegid()用来取得执行目前进程有效组织别码。有效的组织别码用来决定进程执行时组的权限。

返回值:返回有效的组织别码。

geteuid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: uid_t geteuid(void)

函数说明: geteuid()用来取得执行目前进程有效的用户识别码。

返回值: 返回有效的用户识别码。

getgid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: gid_t getgid(void)

函数说明: getgid()用来取得执行目前进程的组织别码。

返回值:返回组织别码。

getgrent 函数

表头文件: #include<grp.h>

#include <sys/types.h>

函数原型: struct group *getgrent(void)

函数说明: getgrent()用来从组文件(/etc/group)中读取一项组数据,该数据以 group 结构返

回。第一次调用时会取得第一项组数据,之后每调用一次就会返回下一项数据,直到已无任何数据时返回 NULL。

struct group{

char *gr_name; /*组名称*/

char *gr_passwd; /* 组密码*/

gid_t gr_gid; /*组织别码*/

char **gr_mem; /*组成员账号*/

}

返回值:返回 group 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

getgrgid 函数

表头文件: #include<grp.h>

#include<sys/types.h>

函数原型: struct group * getgrgid(gid_t gid)

函数说明: getgrgid() 用来依参数 gid 指定的组织识别码逐一搜索组文件,找到时便将该组的数据以 group 结构返回。group 结构请参考 getgrent()。

返回值:返回 group 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

getgrnam 函数

表头文件: #include<grp.h>

#include<sys/types.h>

函数原型: struct group * getgrnam(const char * name)

函数说明: getgrnam() 用来逐一搜索参数 name 指定的组名称,找到时便将该组的数据以 group 结构返回。group 结构请参考 getgrent()。

返回值:返回 group 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

getgroups 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: int getgroups(int size, gid_t list[])

函数说明: getgroups() 用来取得目前用户所属的组代码。参数 size 为 list[] 所能容纳的 gid_t 数目。如果参数 size 值为零,此函数仅会返回用户所属的组数。

返回值:返回组织识别码,如有错误则返回-1。

getpw 函数

表头文件: #include<pwd.h>

#include<sys/types.h>

函数原型: int getpw(uid_t uid, char *buf)

函数说明: getpw() 会从/etc/passwd 中查找符合参数 uid 所指定的用户账号数据,找不到相关数据就返回-1,所返回的 buf 字符串格式如下:账号:密码:用户识别码(uid):组织识别码(gid):全名:根目录:shell。

返回值:返回 0 表示成功,有错误发生时返回-1。

附加说明:

a) getpw() 会有潜在的安全性问题,请尽量使用别的函数取代。

b) 使用 shadow 的系统已把用户密码抽出/etc/passwd,因此使用 getpw() 取得的密码将为“x”。

getpwent 函数

表头文件: #include<pwd.h>

#include<sys/types.h>

函数原型: struct passwd * getpwent(void)

函数说明: getpwent() 用来从密码文件(/etc/passwd)中读取一项用户数据,该用户的

数据以 passwd 结构返回。第一次调用时会取得第一位用户数据,之后每调用一次就会返回下一项数据,直到已无任何数据时返回 NULL。

passwd 结构定义如下

```
struct passwd{
char * pw_name; /*用户账号*/
char * pw_passwd; /*用户密码*/
uid_t pw_uid; /*用户识别码*/
gid_t pw_gid; /*组识别码*/
char * pw_gecos; /*用户全名*/
char * pw_dir; /*家目录*/
char * pw_shell; /* 所使用的 shell 路径*/
};
```

返回值:返回 passwd 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

附加说明: getpwent()在第一次调用时会打开密码文件,读取数据完毕后可使用 endpwent()来关闭该密码文件。错误代码 ENOMEM 内存不足,无法配置 passwd 结构。

getpwnam 函数

表头文件: #include<pwd.h>

#include<sys/types.h>

函数原型: struct passwd * getpwnam(const char * name)

函数说明: getpwnam()用来逐一搜索参数 name 指定的账号名称,找到时便将该用户的数据以 passwd 结构返回。passwd 结构请参考 getpwent()。

返回值:返回 passwd 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

getpwuid 函数

表头文件: #include<pwd.h>

#include<sys/types.h>

函数原型: struct passwd * getpwuid(uid_t uid)

函数说明: getpwuid()用来逐一搜索参数 uid 指定的用户识别码,找到时便将该用户的数据以结构返回结构请参考将该用户的数据以 passwd 结构返回。passwd 结构请参考 getpwent()。

返回值:返回 passwd 结构数据,如果返回 NULL 则表示已无数据,或者有错误发生。

getuid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: uid_t getuid(void)

函数说明: getuid()用来取得执行目前进程的用户识别码。返回值: 用户识别码。

getutent 函数

表头文件: #include<utmp.h>

函数原型: struct utmp * getutent(void)

函数说明: getutent()用来从 utmp 文件(/var/run/utmp)中读取一项登录数据,该数据以 utmp

结构返回。第一次调用时会取得第一位用户数据,之后每调用一次就会返回下一项数据,直到已无任何数据时返回 NULL。

utmp 结构定义如下

```
struct utmp
{
    short int ut_type; /*登录类型*/
    pid_t ut_pid; /*login 进程的 pid*/
    char ut_line[UT_LINESIZE]; /*登录装置名,省略了 “/dev/” */
    char ut_id[4]; /* Inittab ID*/
    char ut_user[UT_NAMESIZE]; /*登录账号*/
    char ut_host[UT_HOSTSIZE]; /*登录账号的远程主机名称*/
    struct exit_status ut_exit; /* 当类型为 DEAD_PROCESS 时进程的结束状态*/
    long int ut_session; /*Sessioc ID*/
    struct timeval ut_tv; /*时间记录*/
    int32_t ut_addr_v6[4]; /*远程主机的网络地址*/
    char __unused[20]; /* 保留未使用*/
};
```

ut_type 有以下几种类型:

EMPTY 此为空的记录。

RUN_LVL 记录系统 run-level 的改变

BOOT_TIME 记录系统开机时间

NEW_TIME 记录系统时间改变后的时间

OLD_TIME 记录当改变系统时间时的时间。

INIT_PROCESS 记录一个由 init 衍生出来的进程。

LOGIN_PROCESS 记录 login 进程。

USER_PROCESS 记录一般进程。

DEAD_PROCESS 记录一结束的进程。

ACCOUNTING 目前尚未使用。

exit_status 结构定义:

```
struct exit_status
{
    short int e_termination; /*进程结束状态*/
    short int e_exit; /*进程退出状态*/
};
```

timeval 的结构定义请参考 gettimeofday()。

相关常数定义如下:

UT_LINESIZE 32

UT_NAMESIZE 32

UT_HOSTSIZE 256

返回值:返回 utmp 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

附加说明: getutent()在第一次调用时会打开 utmp 文件,读取数据完毕后可使用 endutent()来关闭该 utmp 文件。

getutid 函数

表头文件: #include<utmp.h>

函数原型: struct utmp *getutid(const struct utmp *ut)

getutid()用来从目前 utmp 文件的读写位置逐一往后搜索参数 ut 指定的记录,

函数说明:如果 ut->ut_type 为 RUN_LVL,BOOT_TIME,NEW_TIME,OLD_TIME 其中之一则查找与 ut->ut_type 相符的记录;若 ut->ut_type 为 INIT_PROCESS,LOGIN_PROCESS,USER_PROCESS 或 DEAD_PROCESS 其中之一,则查找与 ut->ut_id 相符的记录。找到相符的记录便将该数据以 utmp 结构返回。utmp 结构请参考 getutent()。

返回值:返回 utmp 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

getutline 函数

表头文件: #include<utmp.h>

函数原型: struct utmp * getutline (const struct utmp *ut)

函数说明:getutline() 用来从目前 utmp 文件的读写位置逐一往后搜索 ut_type 为 USER_PROCESS 或 LOGIN_PROCESS 的记录,而且 ut_line 和 ut->ut_line 相符。找到相符的记录便将该数据以 utmp 结构返回,utmp 结构请参考 getutent()。

返回值:返回 utmp 结构数据,如果返回 NULL 则表示已无数据,或有错误发生。

initgroups 函数

表头文件: #include<grp.h>

#include<sys/types.h>

函数原型: int initgroups(const char *user,gid_t group)

函数说明:initgroups()用来从组文件(/etc/group)中读取一项组数据,若该组数据的成员中有参数 user 时,便将参数 group 组识别码加入到此数据中,user 不能为 NULL。

返回值:执行成功则返回 0,失败则返回-1,错误码存于 errno。

pututline 函数

表头文件: #include<utmp.h>

函数原型: struct pututline(const struct utmp *ut)

函数说明: pututline()用来将参数 ut 的 utmp 结构记录到 utmp 文件中。此函数会先用 getutid()来取得正确的写入位置,如果没有找到相符的记录则会加入到 utmp 文件尾,utmp 结构请参考 getutent()。

返回值:无。

附加说明: 需要有写入/var/run/utmp 的权限。

seteuid 函数

表头文件: #include <sys/types.h>

#include <unistd.h>

函数原型: int seteuid(uid_t euid)

函数说明: seteuid()用来重新设置执行目前进程的有效用户识别码。

返回值:执行成功则返回 0,失败则返回-1,错误代码存于 errno。

附加说明: 请参考 setuid。

setfsgid 函数

表头文件: #include<sys/fsuid.h>

函数原型: int setfsgid(uid_t fsgid)

函数说明: setfsgid()用来重新设置目前进程的文件系统的组织识别码。一般情况下,文件系统的组织识别码(fsgid)与有效的组织识别码(egid)是相同的。如果是超级用户调用此函数,参数 fsgid 可以为任何值,否则参数 fsgid 必须为 real/effective/saved 的组织识别码之一。

返回值:执行成功则返回 0,失败则返回-1,错误代码存于 errno。

setfsuid 函数

表头文件: #include<sys/fsuid.h>

函数原型: int setfsuid(uid_t fsuid)

函数说明: setfsuid()用来重新设置目前进程的文件系统的用户识别码。一般情况下,文件系统的用户识别码(fsuid)与有效的用户识别码(euid)是相同的。如果是超级用户调用此函数,参数 fsuid 可以为任何值,否则参数 fsuid 必须为 real/effective/saved 的用户识别码之一。

返回值:执行成功则返回 0,失败则返回-1,错误代码存于 errno。

setgid 函数

表头文件: #include<sys/types.h>

#include <unistd.h>

函数原型: int setgid(gid_t gid)

函数说明:setgid()用来将目前进程的真实组织识别码(real gid)设成参数 gid 值。如果是超级用户身份执行此调用,则 real、effective 与 savedgid 都会设成参数 gid。

返回值:设置成功则返回 0,失败则返回-1,错误代码存于 errno 中。

错误代码:EPERM 并非以超级用户身份调用,而且参数 gid 并非进程的 effective gid 或 saved gid 值之一。EINVAL group id 在当前命名空间无效。

setgrent 函数

表头文件: #include<grp.h>

#include<sys/types.h>

函数原型: void setgrent(void)

函数说明: setgrent()用来将 getgrent()的读写地址指回组文件开头,以允许重复扫描。

返回值:无。

附加说明: 请参考 setpwent()。

setgroups 函数

表头文件: #include<grp.h>

函数原型: int setgroups(size_t size,const gid_t * list)

函数说明: setgroups()用来将 list 数组中所标明的组加入到目前进程的组设置中。参数 size 为 list()的 gid_t 数目,最大值为 NGROUP(32)。

返回值:设置成功则返回 0,如有错误则返回-1。

错误代码:

EFAULT 参数 list 数组地址不合法。

EPERM 权限不足,必须是 root 权限。

EINVAL 参数 size 值大于 NGROUP(32)。

setpwent 函数

表头文件: #include<pwd.h>

#include<sys/types.h>

函数原型: void setpwent(void)

函数说明: setpwent()用来将 getpwent()的读写地址指回密码文件开头。

返回值:无。

setregid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: int setregid(gid_t rgid,gid_t egid)

函数说明: setregid()用来将参数 rgid 设为目前进程的真实组识别码,将参数 egid 设置为目前进程的有效组识别码。如果参数 rgid 或 egid 值为-1,则对应的识别码不会改变。

返回值:执行成功则返回 0,失败则返回-1,错误代码存于 errno。

setreuid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: int setreuid(uid_t ruid,uid_t euid)

函数说明: setreuid()用来将参数 ruid 设为目前进程的真实用户识别码,将参数 euid 设置为目前进程的有效用户识别码。如果参数 ruid 或 euid 值为-1,则对应的识别码不会改变。

返回值:执行成功则返回 0,失败则返回-1,错误代码存于 errno。

setuid 函数

表头文件: #include<unistd.h>

#include<sys/types.h>

函数原型: int setuid(uid_t uid)

函数说明: setuid()用来重新设置执行目前进程的用户识别码。不过,要让此函数有作用,其有效的用户识别码必须为 0(root)。在 Linux 下,当 root 使用 setuid()来转换成其他用户识别码时,root 权限会被抛弃,完全转换成该用户身份,也就是说,该进程往后将不再具有可 setuid()的权利,如果只是向暂时抛弃 root 权限,稍后想重新取回权限,则必须使用 seteuid()。

返回值:执行成功则返回 0,失败则返回-1,错误代码存于 errno。

附加说明: 一般在编写具 setuid root 的程序时,为减少此类程序带来的系统安全风险,在使用完 root 权限后建议马上执行 setuid(getuid());来抛弃 root 权限。此外,进程 uid 和 euid 不一致时 Linux 系统将不会产生 core dump。

setutent 函数

表头文件: #include<utmp.h>

函数原型: void setutent(void)

函数说明: setutent()用来将 getutent()的读写地址指回 utmp 文件开头。

utmpname 函数

表头文件: #include<utmp.h>

函数原型: int utmpname(const char * file)

函数说明: utmpname () 设置其他 utmp 函数要访问的 utmp 格式文件的名称。如果在使用其他函数之前不使用 utmpname () 设置文件名, 则它们将采用<paths.h>中定义的_PATH_UTMP。

返回值:无。

1.8 文件操作

open 函数

头文件: #include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

函数原型: int open(const char *pathname, int flags)

int open(const char *pathname, int flags, mode_t mode)

参数说明: pathname 为将要打开/创建的文件名字, 可以使用绝对路径或相对路径, 使用相对路径时要注意当前进程所在的工作目录

flags 为以下选项:

O_RDONLY 以只读方式打开文件

O_WRONLY 以只写方式打开文件

O_RDWR 以可读可写方式打开文件

注: 以上 3 种为互斥选项, 不可同时使用, 但可以与以下选项做运算组合 OR (|) 使用

O_CREAT 如果要打开的文件不存在则创建该文件

O_EXCL 如果已经指定了 O_CREAT 且要创建的文件已存在, 则返回错误, 否则将创建该文件

O_NOCTTY 如果要打开的文件为终端设备, 不会将该设备用作终端设备

O_TRUNC 如果要打开的文件已存在, 则清空文件全部内容, 即长度截断为 0

O_APPEND 以追加方式写入文件, 即每次在文件最末尾写入新的内容

O_NONBLOCK 如果要打开的为 FIFO/块文件/字符文件, 则以非阻塞模式打开文件, 并且后继 I/O 也是非阻塞模式

O_NDELAY 同 O_NONBLOCK

O_SYNC 以同步方式打开文件, 等待物理 I/O 结束后再 write, 包括文件属性更新

O_NOFOLLOW 如果要打开的文件为符号链接, 则返回失败

O_DIRECTORY 如果要打开的文件并非目录, 则返回失败, Linux2.2 以上支持

mode 为文件创建时生效, 真正建立文件时会受到 umask 值影响, 因此该文件权限应为 mode-umask, 支持以下选项:

S_IRWXU 00700 权限, 该文件所有者具有可读、可写、可执行权限

S_IRUSR 或 S_IREAD 00400 权限, 该文件所有者具有可读权限

S_IWUSR 或 S_IWRITE 00200 权限, 改文件所有者具有可写权限

S_IXUSR 或 S_IEXEC 00100 权限, 该文件所有者具有可执行权限

S_IRWXG 00070 权限, 该文件用户组具有可读、可写、可执行权限

S_IRGRP 00040 权限, 该文件用户组具有可读权限

S_IWGRP 00020 权限, 该文件用户组具有可写权限

S_IXGRP 00010 权限, 该文件用户组具有可执行权限

S_IRWXO 00007 权限, 其他用户具有可读、可写、可执行权限

S_IROTH 00004 权限, 其他用户具有可读权限

S_IWOTH 00002 权限, 其他用户具有可写权限

S_IXOTH 00001 权限, 其他用户具有可执行权限

返回值: 成功, 返回文件描述符

失败，返回-1，同时更新 errno 的值，错误代码：

EEXIST (17) pathname 所指的文件已存在，却使用了 O_CREAT 和 O_EXCL

EACCES (13) pathname 所指文件不符合权限要求

EROFS (30) 需要写入权限的文件存在于只读文件系统内

EFAULT (14) pathname 指针超出可存取内存空间或地址错误

EINVAL (22) mode 不正确或无效参数

ENAMETOOLONG (36) pathname 文件名过长

ENOTDIR (20) pathname 不是目录

ENOMEM (12) 内存不足

ELOOP (40) pathname 有过多的符号链接问题

EIO (5) I/O 错误

close 函数

头文件： #include <unistd.h>

函数原型： int close(int fd)

函数说明： 关闭由 open()函数打开的文件，将文件占用的资源释放

参数说明： fd 为文件描述符，即 open()函数或 creat()函数的返回值

返回值： 成功，返回 0

失败，返回-1，同时更新 errno 的值，错误代码：

EBADF (9) 错误的文件描述符

read 函数

头文件： #include <unistd.h>

函数原型： ssize_t read(int fd, void *buf, size_t count)

函数说明： 该函数负责从 fd 指向的文件中，最多读取 count 个字节，写到 buf 指向的缓冲区，成功后返回实际读取的字节数

参数说明： fd 为文件描述符，即 open()函数或 creat()函数的返回值

buf 指向接收缓冲区

count 为读取的最大字节数

返回值： 成功，返回实际读取到的字节数，若返回 0 则表示已到达文件末尾或无可读数据；此外，每次读取会使光标移动到读取位置处

失败，返回-1，同时更新 errno 的值，错误代码：

EINTR (4) 调用被中断

EAGAIN (11) 再次尝试，当使用非阻塞 I/O (O_NONBLOCK) 并且无数据可读取时返回该值

EBADF (9) 错误的文件描述符

write 函数

头文件： #include <unistd.h>

函数原型： ssize_t write(int fd, const void *buf, size_t count)

函数说明： 该函数负责将 buf 指向的缓冲区数据，最多写入 count 个字节数，到 fd 指向的文件，成功后返回实际写入的字节数

参数说明： fd 为文件描述符，即 open()函数或 creat()函数的返回值

buf 指向写入缓冲区

count 为写入的最大字节数

返回值： 成功，返回实际写入的字节数

失败，返回-1，同时更新 errno 的值，错误代码：

EINTR（4）调用被中断

EAGAIN（11）再次尝试，当使用非阻塞 I/O（O_NONBLOCK）并且无数据可读取时返回该值

EBADF（9）错误的文件描述符

creat 函数

头文件： #include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

函数原型： int creat(const char *pathname, mode_t mode)

函数说明： 该函数负责创建名为 pathname 的文件

参数说明： pathname 为将要打开/创建的文件路径字符串

mode 与 open()函数中 mode 参数相同

返回值： 成功，返回文件描述符

失败，返回-1，同时更新 errno 的值，错误代码：

EEXIST（17）pathname 所指的文件已存在，却使用了 O_CREAT 和 O_EXCL

EACCES（13）pathname 所指文件不符合权限要求

EROFS（30）需要写入权限的文件存在于只读文件系统内

EFAULT（14）pathname 指针超出可存取内存空间或地址错误

EINVAL（22）mode 不正确或无效参数

ENAMETOOLONG（36）pathname 文件名过长

ENOTDIR（20）pathname 不是目录

ENOMEM（12）内存不足

ELOOP（40）pathname 有过多的符号链接问题

dup 函数

头文件： #include <unistd.h>

函数原型： int dup(int fd)

函数说明： 该函数负责复制 fd 文件描述符，通常用来做重定向

参数说明： fd 为将被复制的文件描述符，即 open()函数或 creat()函数的返回值

返回值： 成功，返回当前可用文件描述符的最小值

失败，返回-1，同时更新 errno 的值，错误代码：

EBADF（9）错误的文件描述符

dup2 函数

头文件： #include <unistd.h>

函数原型： int dup2(int fd1, int fd2)

函数说明： 同 dup()函数，该函数负责复制 fd1 文件描述符到 fd2

参数说明： fd1 为将被复制的文件描述符，即 open()函数或 creat()函数的返回值

fd2 为复制后的文件描述符名称，若 fd2 是已打开的文件描述符，则 fd2 会先被关闭

返回值： 成功，返回新的描述符，该文件描述符与原文件描述符共享各种文件状态，包括锁定、光标、权限

失败，返回-1，同时更新 errno 的值，错误代码：

EBADF (9) 错误的文件描述符

flock 函数

头文件: `#include <sys/file.h>`

函数原型: `int flock(int fd, int operation)`

函数说明: 该函数负责按照 operation 所指定的方式, 对 fd 指向的文件进行加锁/解锁

参数说明: fd 为将被复制的文件描述符, 即 open()函数或 creat()函数的返回值

operation 选项如下:

LOCK_SH 建立共享锁, 多个进程可同时对同一文件做共享锁

LOCK_EX 建立互斥锁, 同一文件同时只能有一个互斥锁

LOCK_UN 解除文件锁定状态

LOCK_NB 无法建立锁定时, 此操作可不被中断, 立即返回进程, 通常与 LOCK_SH 或 LOCK_EX 做 OR() 组合。

返回值: 成功, 返回 0

失败, 返回-1, 同时更新 errno 的值

fcntl 函数

头文件: `#include <unistd.h>`

`#include <fcntl.h>`

函数原型: `int fcntl(int fd, int cmd, .../*arg*/)`

函数说明: 该函数负责根据 cmd 的指令修改 fd 指向的文件属性, 参数 arg 会根据 cmd 的变化而有不同的定义

参数说明: fd 为文件描述符, 即 open()函数或 creat()函数的返回值

cmd 代表操作指令, 包括以下选项:

F_DUPFD 用来查找大于或等于 arg 参数的最小且未使用的文件描述符, 并且复制 fd 的文件描述符, 执行成功则返回新复制的文件描述符, 请参考 dup2()函数

F_GETFD 获取 close-on-exec 标志符, 若此标志符的 FD_CLOEXEC 位为 0, 代表在调用 exec()相关函数时文件将不会被关闭

F_SETFD 设置 close-on-exec 标志符, 该标志符以参数 arg 的 FD_CLOEXEC 位决定

F_GETFL 获取文件描述符的状态标志符, 即 open()函数的 flags 参数

F_SETFL 设置文件描述符的状态标志符, 参数 arg 为新的标志符, 但只允许 O_APPEND、O_NONBLOCK、O_ASYNC 位改变, 其他标志位不受影响

F_GETLK 获取文件锁定状态

F_SETLK 设置文件锁定状态, 此时 flock 结构体的 l_type 值必须是 F_RDLCK 或 F_WRLCK 或 F_UNLCK, 若果失败则返回-1, 错误码为 EACCES 或 EAGAIN

F_SETLKW 设置文件锁定状态, 同 F_SETLK, 但如果无法设置锁定时, 此调用会一直等待锁定动作成功为止, 若等待过程中被信号中断则立即返回-1, 错误码为 EINTR, 其中 arg 为 struct flock 结构体指针, 定义如下:

```
struct flock {
    short int l_type; /*锁定的状态*/
    short int l_whence; /*决定 l_start*/
    off_t l_start; /*锁定区域的起始位置*/
    off_t l_len; /*锁定区域的长度*/
    off_t l_pid; /*锁定动作的进程号*/
};
```

```
}
```

其中 l_type 有三种状态：

F_RDLCK 建立一个读写锁

F_WRLCK 建立一个写入锁

F_UNLCK 删除锁定

l_whence 有三种状态：

SEEK_SET 以文件开头为锁定起始位置

SEEK_CUR 以当前光标为锁定起始位置

SEEK_END 以文件结尾为锁定起始位置

返回值： 成功，返回 0

失败，返回-1，同时更新 errno 的值

fsync 函数

头文件： #include <unistd.h>

函数原型： int fsync(int fd)

函数说明： 该函数负责将 fd 所指向的文件数据，由系统缓冲区写入磁盘，等待 I/O 动作完成后再返回，确保数据同步

参数说明： fd 为文件描述符，即 open()函数或 creat()函数的返回值

返回值： 成功，返回 0

失败，返回-1，同时更新 errno 的值

sync 函数

头文件： #include <unistd.h>

函数原型： int sync(void)

函数说明： 该函数负责将内核中修改过的缓冲区排入写队列，不等待实际 I/O 操作完成，直接返回，所以该函数返回时并不能保证数据已经完全写入磁盘

参数说明： 无

返回值： 成功，返回 0

lseek 函数

头文件： #include <sys/types.h>

#include <unistd.h>

函数原型： off_t lseek(int fd, off_t offset, int whence)

函数说明： 每个已打开文件都有一个读写位置，即光标，当文件打开时光标默认位置是文件开头，若以附加方式打开（如 O_APPEND），则光标指向文件结尾，使用 read()或 write()函数时，光标位置会随之增加，lseek()函数是用来控制光标位置的

参数说明： fd 为文件描述符，即 open()函数或 creat()函数的返回值

offset 是相对于 whence 的偏移量

whence 有以下选项：

SEEK_SET 指向文件开头（offset 不允许为负值）

SEEK_CUR 指向光标当前位置（offset 可以为负值）

SEEK_END 指向文件结尾（offset 可以为负值）

返回值： 成功，返回光标当前位置

失败，返回-1，同时更新 errno 的值，错误代码：

ESPIPE (29) 无效的偏移

附加说明: Linux 不允许 lseek()函数设置 tty 文件, 此动作会返回 ESPIPE (29)

mkstemp 函数

头文件: #include <stdlib.h>

函数原型: int mkstemp(char *template)

函数说明: 该函数负责建立唯一的临时文件, 该函数将以可读写权限打开文件, 若文件不存在则创建该文件

参数说明: template 为临时文件名称字符串, 该字符串声明时必须以数组形式, 并且最后 6 个字符必须是 XXXXXX

返回值: 成功, 返回 0

失败, 返回-1, 同时更新 errno 的值, 错误代码 EINVAL (22) mode 不正确或无效参数, 优先排查最后 6 个字符是否为 XXXXXX

1.9 文件内容操作

fopen 函数

头文件: #include <stdio.h>

函数原型: FILE *fopen(const char *pathname, const char *mode)

函数说明: 该函数负责以 mode 指定的形式打开 pathname 指向的文件

参数说明: pathname 为将要打开/创建的文件路径字符串, 可以使用绝对路径或相对路径, 使用相对路径时要注意当前进程所在的当前目录位置

mode 有以下选项:

r 打开只读文件, 该文件必须存在

r+ 打开可读写文件, 该文件必须存在

w 打开只写文件, 若文件已存在则内容会被清空, 即长度为 0, 若文件不存在则创建该文件

w+ 打开可读写文件, 若文件已存在则内容会被清空, 即长度为 0, 若文件不存在则创建该文件

a 以追加方式打开只写文件, 若文件已存在则写入内容会被追加到文件末尾, 若文件不存在则创建该文件

a+ 以追加方式打开可读写文件, 若文件已存在则写入内容会被追加到文件末尾, 若文件不存在则创建该文件

b 需要与上述其中一种参数组合使用, 用来告诉函数库打开的文件为二进制文件, 并非文本文件; 但在 POSIX 系统, 以及 Linux 中会忽略该字符

返回值: 成功, 返回文件流指针

失败, 返回 NULL, 同时更新 errno 的值

附加说明: 由 fopen()函数创建的新文件具有

S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IROTH|S_IWOTH 权限 (0666), 此文件也会参考 umask 值

freopen 函数

头文件: #include <stdio.h>

函数原型: FILE *freopen(const char *pathname, const char *mode, FILE *stream)

函数说明: 该函数负责先将 stream 文件流关闭, 然后以 mode 方式重新打开 pathname 所指向的文件

参数说明: pathname 为将要打开/创建的文件路径字符串, 可以使用绝对路径或相对路径, 使用

相对路径时要注意当前进程所在的当前目录位置

mode 同 fopen()函数的 mode 选项

stream 已打开的文件流指针，即 fopen()或 freopen()函数的返回值

返回值： 成功，返回新的文件流指针

失败，返回 NULL，同时更新 errno 的值

fdopen 函数

头文件： #include <stdio.h>

函数原型： FILE *fdopen(int fd, const char *mode)

函数说明： 该函数负责以 mode 形式将 fd 文件描述符转换为文件流指针并返回

参数说明： fd 为文件描述符，即 open()函数或 creat()函数的返回值

mode 同 fopen()函数的 mode 选项，此形式必须和原先文件描述词读写模式相同

返回值： 成功，返回文件流指针

失败，返回 NULL，同时更新 errno 的值

fileno 函数

头文件： #include <stdio.h>

函数原型： int fileno(FILE *stream)

函数说明： 该函数负责根据 stream 指向的文件流获取文件描述符并返回

参数说明： stream 为文件流指针

返回值： 成功，返回文件描述符

失败，返回-1

clearerr 函数

头文件： #include <stdio.h>

函数原型： void clearerr(FILE *stream)

函数说明： 该函数负责清除 stream 指向的文件流中使用的错误标志位

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开值

返回值： 无

fclose 函数

头文件： #include <stdio.h>

函数原型： int fclose(FILE *stream)

函数说明： 该函数负责关闭由 fopen()或 freopen()函数打开的文件流，此动作会让缓冲区内容写入文件，并释放系统所提供的文件资源

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开

返回值： 成功，返回 0

失败，返回 EOF（-1），同时更新 errno 的值，错误代码：

EBADF（9）错误的文件描述符

fread 函数

头文件： #include <stdio.h>

函数原型： size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)

函数说明： 该函数负责从 stream 文件流中读取数据到 ptr 指向的缓冲区，总大小为 size * nmemb 字节

参数说明: ptr 指向用来存放数据的缓冲区
size 每块数据大小 (步长)
nmemb 数据块数 (步数)
stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 返回实际读取的数据块数

附加说明: 当返回值小于 nmemb 时, 可能是读到文件结尾或发生错误, 此时需要使用 feof()或 ferror()来确认具体是哪种情况

fwrite 函数

头文件: #include <stdio.h>

函数原型: size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream)

函数说明: 该函数负责将 ptr 指向的缓冲区数据写入 stream 文件流中, 总大小为 size * nmemb 字节

参数说明: ptr 指向用来存放数据的缓冲区
size 每块数据大小 (步长)
nmemb 数据块数 (步数)
stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 返回实际写入的数据块数

feof 函数

头文件: #include <stdio.h>

函数原型: int feof(FILE *stream)

函数说明: 该函数负责判断是否读取到文件末尾

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 文件末尾, 返回 0
非文件末尾, 返回非 0

1.9.10 fgetc 函数

头文件: #include <stdio.h>

函数原型: int fgetc(FILE *stream)

函数说明: 该函数负责从 stream 指向的文件流读取一个字符并返回

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 成功, 返回读取到的字符
失败, 返回 EOF (-1), 读取到文件末尾

fgets 函数

头文件: #include <stdio.h>

函数原型: char *fgets(char *str, int size, FILE *stream)

函数说明: 该函数负责从 stream 指向的文件流读取一行数据写入 str 所指向的缓冲区, 直到读取到换行符\n 或读到文件结尾或已读取 (size-1) 个字符就会停止, 最后加上 NULL 作为结尾

参数说明: str 指向的缓冲区, 用来存储读取后的字符串
size 为最多读取的字符数 (包括结尾的 NULL), 通常使用 str 数组长度
stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 成功, 返回与 str 相同
失败, 返回 NULL, 读取到文件末尾或没有读到任何数据, 更新 eof; 发生错误, 更新 errno 的值

getc 函数

头文件: #include <stdio.h>

函数原型: int getc(FILE *stream)

函数说明: 该函数负责从 stream 指向的文件流读取一个字符并返回; 该函数与 fgetc()函数作用相同, 但该函数是宏定义, 并非真正的函数, 因此不能将其作为指针传递

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 成功, 返回读取到的字符

失败, 返回 EOF (-1), 读取到文件末尾

ungetc 函数

头文件: #include <stdio.h>

函数原型: int ungetc(int c, FILE *stream)

函数说明: 该函数负责将参数 c 写回到 stream 所指定的文件流中, 这个被写回的字符被下一个读取文件流的函数取得

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

c 代表将被写回文件流的字符

返回值: 成功, 返回被写回的参数 c

失败, 返回 EOF (-1)

getchar 函数

头文件: #include <stdio.h>

函数原型: int getchar(void)

函数说明: 该函数负责从标准输入设备读取一个字符, 然后将该字符从 unsigned char 转换成 int 后返回; 该函数是 getc(stdin)的宏定义, 并非真正的函数, 因此不能将其作为指针传递

参数说明: 无

返回值: 成功, 返回读取到的字符

失败, 返回 EOF (-1)

gets 函数

头文件: #include <stdio.h>

函数原型: char *gets(char *str)

函数说明: 该函数负责从标准输入设备读取字符串并写入 str 指向的缓冲区; 直到读取到换行符或发生 EOF (-1) 时为止, 换行符会被替换为 '\0' 以此结束字符串

参数说明: str 指向存储字符串的缓冲区

返回值: 成功, 返回与 str 相同指针

失败, 返回 NULL, 此时需要使用 feof()或 ferror()来确认具体是哪种情况

fputc 函数

头文件: #include <stdio.h>

函数原型: int fputc(int c, FILE *stream)

函数说明: 该函数负责将字符 c 转换为 unsigned char 后写入 stream 指向的文件流中

参数说明: c 为将要写入的字符

stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值： 成功，返回写入的字符 c

失败，返回 EOF（-1）

fputs 函数

头文件： #include <stdio.h>

函数原型： int fputs(const char *str, FILE *stream)

函数说明： 该函数负责将 str 指向的字符串写入 stream 指向的文件流中

参数说明： str 为将要写入的字符串

stream 为文件流指针，由 fopen()或 freopen()函数打开

返回值： 成功，返回写入的字符个数

失败，返回 EOF（-1）

putc 函数

头文件： #include <stdio.h>

函数原型： int putc(int c, FILE *stream)

函数说明： 该函数负责将参数 c 转换为 unsigned char 后写入到 stream 指向的文件流；该函数与 fputs()

函数作用相同，但该函数是宏定义，并非真正的函数，因此不能将其作为指针传递

参数说明： c 为将要写入的字符

stream 为文件流指针，由 fopen()或 freopen()函数打开

返回值： 成功，返回写入的字符

失败，返回 EOF（-1）

putchar 函数

头文件： #include <stdio.h>

函数原型： int putchar (int c)

函数说明： 该函数负责将参数 c 写入到标准输出设备,该函数与 putc()函数作用相同,但该函数是 putc(c, stdout)的宏定义，并非真正的函数，因此不能将其作为指针传递

参数说明： c 为将要写入的字符

返回值： 成功，返回写入的字符

失败，返回 EOF（-1）

1.9.20 puts 函数

头文件： #include <stdio.h>

函数原型： int puts (const char *str)

函数说明： 该函数负责将 str 指向的字符串输入到标准输出设备，并在结尾自动输出一个换行符

参数说明： str 指向将要输出的字符串

返回值： 成功，返回输出的字符串长度

失败，返回 EOF（-1）

ftell 函数

头文件： #include <stdio.h>

函数原型： long ftell(FILE *stream)

函数说明： 该函数负责获取 stream 指向的文件流当前读写位置，即光标位置

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开

返回值： 成功，返回光标位置

失败，返回-1，同时更新 errno 的值，错误代码：

EBADF (9) 错误的文件描述符

fseek 函数

头文件： #include <stdio.h>

函数原型： int fseek(FILE *stream, long offset, int whence)

函数说明： 每个已打开文件都有一个读写位置，即光标，当文件打开时光标默认位置是文件开头，若以附加方式打开（如 O_APPEND），则光标指向文件结尾，使用 fread()或 fwrite()函数时，光标位置会随之增加，fseek()函数是用来控制光标位置的

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开值

offset 是相对于 whence 的偏移量

whence 有以下选项：

SEEK_SET 指向文件开头（offset 不允许为负值）

SEEK_CUR 指向光标当前位置（offset 可以为负值）

SEEK_END 指向文件结尾（offset 可以为负值）

返回值： 成功，返回 0

失败，返回-1，同时更新 errno 的值，错误代码：

ESPIPE (29) 无效的偏移

附加说明： fseek()不像 lseek()会返回当前位置，因此必须使用 ftell()来获取光标位置

rewind 函数

头文件： #include <stdio.h>

函数原型： void rewind(FILE *stream)

函数说明： 该函数负责将 stream 指向的文件流当前读写位置移动到文件开头，相当于 fseek(stream, 0, SEEK_SET)

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开

返回值： 无

fflush 函数

头文件： #include <stdio.h>

函数原型： int fflush(FILE *stream)

函数说明： 该函数负责强制将缓冲区中的数据写入到 stream 指定的文件流中，如果 stream 为 NULL 则会将所有打开的文件数据更新

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开值

返回值： 成功，返回 0

失败，返回 EOF (-1)，同时更新 errno 的值，错误代码：

EBADF (9) 错误的文件描述符

setbuf 函数

头文件： #include <stdio.h>

函数原型： void setbuf(FILE *stream, char *buf)

函数说明： 在打开文件流后，读取内容之前，调用该函数可以设置文件流的缓冲区

参数说明： stream 为文件流指针，由 fopen()或 freopen()函数打开

buf 指向长度为 BUFSIZ (stdio.h 中定义) 的自定义缓冲区

返回值: 无

补充说明: 通常在设置之后该流就是全缓冲的, 但是如果该流与一个终端设备相关, 那么某些系统也可以将其设置为行缓冲。为了关闭缓冲, 可以将 buf 参数设置为 NULL, 此时相当于调用 setvbuf(stream, buf, buf?_IOFBF:_IONBF, BUFSIZ)

setbuffer 函数

头文件: #include <stdio.h>

函数原型: void setbuffer(FILE *stream, char *buf, size_t size)

函数说明: 在打开文件流后, 读取内容之前, 调用该函数可以设置文件流的缓冲区

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

buf 指向自定义缓冲区

size 为缓冲区大小

返回值: 无

setlinebuf 函数

头文件: #include <stdio.h>

函数原型: void setlinebuf(FILE *stream)

函数说明: 该函数用来设置文件流以换行为依据的缓冲 IO, 即行缓冲, 也叫线性缓冲

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

返回值: 无

补充说明: 相当于调用 setvbuf(stream, (char *)NULL, _IOLBF, 0)

setvbuf 函数

头文件: #include <stdio.h>

函数原型: void setvbuf(FILE *stream, char *buf, int mode, size_t size)

函数说明: 在打开文件流后, 读取内容之前, 调用该函数可以设置文件流的缓冲区

参数说明: stream 为文件流指针, 由 fopen()或 freopen()函数打开

buf 指向自定义缓冲区, 如果为 NULL 则为无缓冲

size 为缓冲区大小

mode 为以下选项:

_IONBF 无缓冲 IO

_IOLBF 以换行为依据的无缓冲 IO

_IOFBF 完全无缓冲 IO

返回值: 成功 0, 失败非零。

mktemp 函数

头文件: #include <stdlib.h>

函数原型: char *mktemp(char *template)

函数说明: 该函数负责建立唯一的临时文件名, 该函数将以可读写权限打开文件, 若文件不存在则创建该文件

参数说明: template 为临时文件名称字符串, 该字符串声明时必须以数组形式, 并且最后 6 个字符必须是 XXXXXX

返回值: 成功, 返回生成的文件名字符串

失败, 返回 NULL, 同时更新 errno 的值

1.10 文件权限控制接口

access 函数

表头文件: #include<unistd.h>

函数原型: int access(const char * pathname,int mode)

函数说明: access()会检查是否可以读/写某一已存在的文件。参数 mode 有几种情况组合, R_OK,W_OK,X_OK 和 F_OK。R_OK,W_OK 与 X_OK 用来检查文件是否具有读取、写入和执行的权限。F_OK 则是用来判断该文件是否存在。由于 access()只作权限的核查,并不理会文件形态或文件内容,因此,如果一目录表示为“可写入”,表示可以在该目录中建立新文件等操作,而非意味此目录可以被当做文件处理。例如,你会发现 DOS 的文件都具有“可执行”权限,但用 execve()执行时则会失败。

返回值: 若所有欲查核的权限都通过了检查则返回 0 值,表示成功,只要有一权限被禁止则返回-1。

错误代码:

EACCESS 参数 pathname 所指定的文件不符合所要求测试的权限。

EROFS 欲测试写入权限的文件存在于只读文件系统内。

EFAULT 参数 pathname 指针超出可存取内存空间。

EINVAL 参数 mode 不正确。

ENAMETOOLONG 参数 pathname 太长。

ENOTDIR 参数 pathname 为一目录。

ENOMEM 核心内存不足。

ELOOP 参数 pathname 有过多符号连接问题。

EIO I/O 存取错误。

附加说明: 使用 access()作用户认证方面的判断要特别小心,例如在 access()后再做 open()的空文件可能会造成系统安全上的问题。

alphasort 函数

表头文件: #include<dirent.h>

函数原型: int alphasort(const struct dirent **a,const struct dirent **b)

函数说明: alphasort() 和 versionsort() 函数可以用作比较函数 compar()。前者使用 strcoll(3)对目录项进行排序,后者使用 strverscmp(3) 对字符串 (*a) ->d\u name 和 (*b) ->d\u name 进行排序。

返回值: 如果第一个参数分别小于、等于或大于第二个参数,则 alphasort() 和 versionsort()函数返回小于、等于或大于零的整数。返回值参考 qsort()。

chdir 函数

表头文件: #include<unistd.h>

函数原型: int chdir(const char * path)

函数说明: chdir()用来将当前的工作目录改变成以参数 path 所指的目录。

返回值: 执行成功则返回 0,失败返回-1,errno 为错误代码。

chmod 函数

表头文件: #include<sys/stat.h>

函数原型: `int chmod(const char * path, mode_t mode)`

函数说明: `chmod()`会依参数 `mode` 权限来更改参数 `path` 指定文件的权限。

参数 `mode` 有下列数种组合:

`S_ISUID 04000` 文件的(set user-id on execution)位

`S_ISGID 02000` 文件的(set group-id on execution)位

`S_ISVTX 01000` 文件的 sticky 位

`S_IRUSR(S_IREAD) 00400` 文件所有者具可读取权限

`S_IWUSR(S_IWRITE) 00200` 文件所有者具可写入权限

`S_IXUSR(S_IEXEC) 00100` 文件所有者具可执行权限

`S_IRGRP 00040` 用户组具可读取权限

`S_IWGRP 00020` 用户组具可写入权限

`S_IXGRP 00010` 用户组具可执行权限

`S_IROTH 00004` 其他用户具可读取权限

`S_IWOTH 00002` 其他用户具可写入权限

`S_IXOTH 00001` 其他用户具可执行权限

只有该文件的所有者或有效用户识别码为 0,才可以修改该文件权限。基于系统安全,如果欲将数据写入一执行文件,而该执行文件具有 `S_ISUID` 或 `S_ISGID` 权限,则这两个位会被清除。如果一目录具有 `S_ISUID` 位权限,表示在此目录下只有该文件的所有者或 `root` 可以删除该文件。

返回值:权限改变成功返回 0,失败返回-1,错误原因存于 `errno`。

错误代码:

`EPERM` 进程的有效用户识别码与欲修改权限的文件拥有者不同,而且也不具 `root` 权限。

`EACCESS` 参数 `path` 所指定的文件无法存取;

`EROFS` 欲写入权限的文件存在于只读文件系统内;

`EFAULT` 参数 `path` 指针超出可存取内存空间;

`EINVAL` 参数 `mode` 不正确;

`ENAMETOOLONG` 参数 `path` 太长;

`ENOENT` 指定的文件不存在;

`ENOTDIR` 参数 `path` 路径并非一目录;

`ENOMEM` 核心内存不足;

`ELOOP` 参数 `path` 有过多符号连接问题;

`EIO I/O` 存取错误。

chown 函数

表头文件: `#include <unistd.h>`

函数原型: `int chown(const char * path, uid_t owner, gid_t group)`

函数说明:`chown()`会将参数 `path` 指定文件的所有者变更为参数 `owner` 代表的用户,而将

该文件的组变更为参数 `group` 组。如果参数 `owner` 或 `group` 为-1,对应的所有者或组不会有所改变。`root` 与文件所有者皆可改变文件组,但所有者必须是参数 `group` 组的成员。当 `root` 用 `chown()`改变文件所有者或组时,该文件若具有 `S_ISUID` 或 `S_ISGID` 权限,则会清除此权限位,此外如果具有 `S_ISGID` 权限但不具 `S_IXGRP` 位,则该文件会被强制锁定,文件模式会保留。

返回值:成功则返回 0,失败返回-1,错误原因存于 `errno`。

chroot 函数

表头文件: #include<unistd.h>

函数原型: int chroot(const char * path)

函数说明: chroot()用来改变根目录为参数 path 所指定的目录。只有超级用户才允许改变根目录,子进程将继承新的根目录。

返回值:调用成功则返回 0,失败则返回-1,错误代码存于 errno。

错误代码:

EPERM 权限不足,无法改变根目录。

EFAULT 参数 path 指针超出可存取内存空间。

ENAMETOOLONG 参数 path 太长。

ENOTDIR 路径中的目录存在但却非真正的目录。

EACCESS 存取目录时被拒绝

ENOMEM 核心内存不足。

ELOOP 参数 path 有过多符号连接问题。

EIO I/O 存取错误。

closedir 函数

表头文件: #include<sys/types.h>

#include<dirent.h>

函数原型: int closedir(DIR *dirp)

函数说明: closedir()关闭参数 dir 所指的目录流。

返回值:关闭成功则返回 0,失败返回-1,错误原因存于 errno 中。

错误代码: EBADF 参数 dir 为无效的目录流。

fchdir 函数

表头文件: #include<unistd.h>

函数原型: int fchdir(int fd)

函数说明: fchdir()用来将当前的工作目录改变成以参数 fd 所指的文件描述词。

返回值:执行成功则返回 0,失败返回-1,errno 为错误代码。

fchmod 函数

表头文件: #include<sys/stat.h>

函数原型: int fchmod(int fd, mode_t mode)

函数说明: fchmod()会依参数 mode 权限来更改参数 fildes 所指文件的权限。参数 fildes 为已打开文件的文件描述词。参数 mode 请参考 chmod()。

返回值:权限改变成功则返回 0,失败返回-1,错误原因存于 errno。

错误原因:

EBADF 参数 fildes 为无效的文件描述词;

EPERM 进程的有效用户识别码与欲修改权限的文件所有者不同,而且也不具 root 权限;

EROFS 欲写入权限的文件存在于只读文件系统内;

EIO I/O 存取错误。

fchown 函数

表头文件: #include<unistd.h>

函数原型: int fchown(int fd, uid_t owner, gid_t group)

函数说明: fchown()会将参数 fd 指定文件的所有者变更为参数 owner 代表的用户,而将该文件的组变更为参数 group 组。如果参数 owner 或 group 为-1,对映的所有者或组有所改变。参数 fd 为已打开的文件描述词。当 root 用 fchown()改变文件所有者或组时,该文件若具有 S_ISUID 或 S_ISGID 权限,则会清除此权限位。

返回值:成功则返回 0,失败则返回-1,错误原因存于 errno。

错误代码:

EBADF 参数 fd 文件描述词为无效的或该文件已关闭;

EPERM 进程的有效用户识别码与欲修改权限的文件所有者不同,而且也不具 root 权限,

或是参数 owner、group 不正确;

EROFS 欲写入的文件存在于只读文件系统内;

ENOENT 指定的文件不存在;

EIO I/O 存取错误。

fstat 函数

表头文件: #include<sys/stat.h>

#include<unistd.h>

#include<sys/types.h>

函数原型: int fstat(int fd, struct stat *statbuf)

函数说明: fstat()用来将参数 fides 所指的文件状态,复制到参数 buf 所指的结构中(struct stat)。Fstat()与 stat()作用完全相同,不同处在于传入的参数为已打开的文件描述词。详细内容请参考 stat()。

返回值:执行成功则返回 0,失败返回-1,错误代码存于 errno。

ftruncate 函数

表头文件: #include<unistd.h>

函数原型: int ftruncate(int fd, off_t length)

函数说明: ftruncate()会将参数 fd 指定的文件大小改为参数 length 指定的大小。参数 fd 为已打开的文件描述词,而且必须是以写入模式打开的文件。如果原来的文件大小比参数 length 大,则超过的部分会被删去。

返回值:执行成功则返回 0,失败返回-1,错误原因存于 errno。

错误代码:

EBADF 参数 fd 文件描述词为无效的或该文件已关闭;

EINVAL 参数 fd 为一 socket 并非文件,或是该文件并非以写入模式打开。

getcwd 函数

表头文件: #include<unistd.h>

函数原型: char * getcwd(char * buf, size_t size)

函数说明: getcwd()会将当前的工作目录绝对路径复制到参数 buf 所指的内存空间,参数 size 为 buf 的空间大小。在调用此函数时,buf 所指的内存空间要足够大,若工作目录绝对路径的字符串长度超过参数 size 大小,则返回值 NULL,errno 的值则为 ERANGE。倘若参数 buf 为 NULL,getcwd()会依参数 size 的大小自动配置内存(使用 malloc()),如果参数 size 也为 0,则 getcwd()会依工作目录绝对路径的字符串程度来决定所配置的内存大小,进程可以在使用完此字符串后利用 free()来释放此空间。

返回值: 执行成功则将结果复制到参数 buf 所指的内存空间,或是返回自动配置的字符串

指针。失败返回 NULL,错误代码存于 errno。

link 函数

表头文件: #include<unistd.h>

函数原型: int link (const char * oldpath,const char * newpath)

函数说明:link()以参数 newpath 指定的名称来建立一个新的连接(硬连接)到参数 oldpath 所指定的已存在文件。如果参数 newpath 指定的名称为一已存在的文件则不会建立连接。

返回值:成功则返回 0,失败返回-1,错误原因存于 errno。

附加说明: link()所建立的硬连接无法跨越不同文件系统,如果需要请改用 symlink()。

错误代码:

EXDEV 参数 oldpath 与 newpath 不是建立在同一文件系统;

EPERM 参数 oldpath 与 newpath 所指的文件系统不支持硬连接;

EROFS 文件存在于只读文件系统中;

EFAULT 参数 oldpath 或 newpath 指针超出可存取内存空间;

ENAMETOOLONG 参数 oldpath 或 newpath 太长;

ENOMEM 核心内存不足;

EEXIST 参数 newpath 所指的文件名已存在;

EMLINK 参数 oldpath 所指的文件已达最大连接数目;

ELOOP 参数 pathname 有过多符号连接问题;

ENOSPC 文件系统的剩余空间不足;

EIO I/O 存取错误。

lstat 函数

表头文件: #include<sys/stat.h>

#include<unistd.h>

#include<sys/types.h>

函数原型: int lstat(const char *pathname, struct stat *statbuf)

函数说明:lstat()与 stat()作用完全相同,都是取得参数 file_name 所指的文件状态,其差别在于,当文件为符号连接时,lstat()会返回该 link 本身的状态。详细内容请参考 stat()。

返回值:执行成功则返回 0,失败返回-1,错误代码存于 errno。

opendir 函数

表头文件: #include<sys/types.h>

#include<dirent.h>

函数原型: DIR * opendir(const char * name)

函数说明: opendir()用来打开参数 name 指定的目录,并返回 DIR*形态的目录流,和 open()类似,接下来对目录的读取和搜索都要使用此返回值。

返回值:成功则返回 DIR* 型态的目录流,打开失败则返回 NULL。

错误代码:

EACCESS 权限不足;

EMFILE 已达到进程可同时打开的文件数上限;

ENFILE 已达到系统可同时打开的文件数上限;

ENOTDIR 参数 name 非真正的目录;

ENOENT 参数 name 指定的目录不存在,或是参数 name 为一空字符串;

ENOMEM 核心内存不足。

readdir 函数

表头文件: #include<dirent.h>

函数原型: struct dirent * readdir(DIR * dir)

函数说明: readdir()返回参数 dir 目录流的下个目录进入点。

结构 dirent 定义如下:

struct dirent *readdir(DIR *dirp);

```
struct dirent {
    ino_t      d_ino;
    off_t      d_off;
    unsigned short d_reclen;
    unsigned char d_type;
    char        d_name[256];
};d_ino 此目录进入点的 inode
```

d_off 目录文件开头至此目录进入点的位移

d_reclen d_name 的长度,不包含 NULL 字符

d_type d_name 所指的文件类型

d_name 文件名

返回值:成功则返回下个目录进入点,有错误发生或读取到目录文件尾则返回 NULL。

附加说明:EBADF 参数 dir 为无效的目录流。

readlink 函数

表头文件: #include<unistd.h>

函数原型: ssize_t readlink(const char *pathname, char *buf, size_t bufsiz)

函数说明: readlink()会将参数 path 的符号连接内容存到参数 buf 所指的内存空间,返回的内容不是以 NULL 作字符串结尾,但会将字符串的字符数返回。若参数 bufsiz 小于符号连接的内容长度,过长的内容会被截断。

返回值:执行成功则传符号连接所指的文件路径字符串,失败则返回-1,错误代码存于 errno。

错误代码:

EACCESS 取文件时被拒绝,权限不够;

EINVAL 参数 bufsiz 为负数;

EIO I/O 存取错误;

ELOOP 欲打开的文件有过多符号连接问题;

ENAMETOOLONG 参数 path 的路径名称太长;

ENOENT 参数 path 所指定的文件不存在;

ENOMEM 核心内存不足;

ENOTDIR 参数 path 路径中的目录存在但却非真正的目录。

remove 函数

表头文件: #include<stdio.h>

函数原型: int remove(const char * pathname)

函数说明:remove()会删除参数 pathname 指定的文件。如果参数 pathname 为一文件,则

调用 unlink()处理,若参数 pathname 为一目录,则调用 rmdir()来处理。请参考 unlink()与 rmdir()。

返回值:成功则返回 0,失败则返回-1,错误原因存于 errno。

错误代码:

EROFS 欲写入的文件存在于只读文件系统内;

EFAULT 参数 pathname 指针超出可存取内存空间;

ENAMETOOLONG 参数 pathname 太长;

ENOMEM 核心内存不足;

ELOOP 参数 pathname 有过多符号连接问题;

EIO I/O 存取错误。

rename 函数

表头文件: #include<stdio.h>

函数原型: int rename(const char * oldpath,const char * newpath)

函数说明:rename()会将参数 oldpath 所指定的文件名称改为参数 newpath 所指的文件名称。若 newpath 所指定的文件已存在,则会被删除。

返回值:执行成功则返回 0,失败返回-1,错误原因存于 errno。

rewinddir 函数

表头文件: #include<sys/types.h>

#include<dirent.h>

函数原型: void rewinddir(DIR *dirp)

函数说明: rewinddir()用来设置参数 dir 目录流目前的读取位置为原来开头的读取位置。

返回值:无。

错误代码: EBADF dir 为无效的目录流。

seekdir 函数

表头文件: #include<dirent.h>

函数原型: void seekdir(DIR *dirp, long loc)

函数说明: seekdir()用来设置参数 dir 目录流目前的读取位置,在调用 readdir()时便从此新位置开始读取,参数 loc 是先前调用的 telldir 的返回值。

返回值:无。

错误代码: EBADF 参数 dir 为无效的目录流。

stat 函数

表头文件: #include<sys/stat.h>

#include <sys/stat.h>

#include<unistd.h>

函数原型: int stat(const char *pathname, struct stat *statbuf)

函数说明: stat()用来将参数 file_name 所指的文件状态,复制到参数 buf 所指的结构中。

下面是 struct stat 内各参数的说明:

```
struct stat {  
    dev_t      st_dev;          /* ID of device containing file */  
    ino_t      st_ino;          /* Inode number */  
    mode_t     st_mode;         /* File type and mode */  
    nlink_t    st_nlink;        /* Number of hard links */
```

```

uid_t      st_uid;          /* User ID of owner */
gid_t      st_gid;          /* Group ID of owner */
dev_t      st_rdev;         /* Device ID (if special file) */
off_t      st_size;         /* Total size, in bytes */
blksize_t  st_blksize;      /* Block size for filesystem I/O */
blkcnt_t   st_blocks;       /* Number of 512B blocks allocated */
/* Since Linux 2.6, the kernel supports nanosecond
   precision for the following timestamp fields.
   For the details before Linux 2.6, see NOTES. */
struct timespec st_atim; /* Time of last access */
struct timespec st_mtim; /* Time of last modification */
struct timespec st_ctim; /* Time of last status change */
#define st_atime st_atim.tv_sec      /* Backward compatibility */
#define st_mtime st_mtim.tv_sec
#define st_ctime st_ctim.tv_sec
};

```

st_dev 文件的设备编号;

st_ino 文件的 i-node;

st_mode 文件的类型和存取的权限;

st_nlink 连到该文件的硬连接数目,刚建立的文件值为 1;

st_uid 文件所有者的用户识别码;

st_gid 文件所有者的组织识别码;

st_rdev 若此文件为装置设备文件,则为其设备编号;

st_size 文件大小,以字节计算;

st_blksize 文件系统的 I/O 缓冲区大小;

st_blocks 占用文件区块的个数,每一区块大小为 512 个字节;

st_atime 文件最近一次被存取或被执行的时间,一般只有在用 mknod、utime、read、write 与 truncate 时改变;

st_mtime 文件最后一次被修改的时间,一般只有在用 mknod、utime 和 write 时才会改变;

st_ctime i-node 最近一次被更改的时间,此参数会在文件所有者、组、权限被更改时更新先返回值:执行成功则返回 0,失败返回-1,错误代码存于 errno。

错误代码:

ENOENT 参数 file_name 指定的文件不存在;

ENOTDIR 路径中的目录存在但却非真正的目录;

ELOOP 欲打开的文件有过多符号连接问题,上限为 16 符号连接;

EFAULT 参数 buf 为无效指针,指向无法存在的内存空间;

EACCESS 存取文件时被拒绝;

ENOMEM 核心内存不足;

ENAMETOOLONG 参数 file_name 的路径名称太长。

symlink 函数

表头文件: #include<unistd.h>

函数原型: int symlink(const char *target, const char *linkpath) 函数说明:

symlink()以参数 newpath 指定的名称来建立一个新的连接(符号连接)到参数

oldpath 所指定的已存在文件。参数 oldpath 指定的文件不一定要存在,如果参数 newpath 指定的名称为一已存在的文件则不会建立连接。

返回值:成功则返回 0,失败返回-1,错误原因存于 errno。

错误代码:

EPERM 参数 oldpath 与 newpath 所指的文件系统不支持符号连接;

EROFS 欲测试写入权限的文件存在于只读文件系统内;

EFAULT 参数 oldpath 或 newpath 指针超出可存取内存空间;

ENAMETOOLONG 参数 oldpath 或 newpath 太长;

ENOMEM 核心内存不足;

EEXIST 参数 newpath 所指的文件名已存在;

EMLINK 参数 oldpath 所指的文件已达到最大连接数目;

ELOOP 参数 pathname 有过多符号连接问题;

ENOSPC 文件系统的剩余空间不足;

EIO I/O 存取错误。

telldir 函数

表头文件: #include<dirent.h>

函数原型: long telldir(DIR *dirp)

函数说明: telldir()返回参数 dir 目录流目前的读取位置。此返回值:代表距离目录文件开头的偏移量返回值:返回下个读取位置,有错误发生时返回-1。

错误代码:EBADF 参数 dir 为无效的目录流。

truncate 函数

表头文件: #include<unistd.h>

#include <sys/types.h>

函数原型: int truncate(const char * path,off_t length)

函数说明: truncate()会将参数 path 指定的文件大小改为参数 length 指定的大小。如果原来的文件大小比参数 length 大,则超过的部分会被删去。

返回值:执行成功则返回 0,失败返回-1,错误原因存于 errno。

错误代码:

EACCESS 参数 path 所指定的文件无法存取;

EROFS 欲写入的文件存在于只读文件系统内;

EFAULT 参数 path 指针超出可存取内存空间;

EINVAL 参数 path 包含不合法字符;

ENAMETOOLONG 参数 path 太长;

ENOTDIR 参数 path 路径并非一目录;

EISDIR 参数 path 指向一目录;

ETXTBUSY 参数 path 所指的文件为共享程序,而且正被执行中;

ELOOP 参数 path' 有过多符号连接问题;

EIO I/O 存取错误。

umask 函数

表头文件: #include<sys/types.h>

#include<sys/stat.h>

函数原型: mode_t umask(mode_t mask)

函数说明: umask()会将系统 umask 值设成参数 mask&0777 后的值,然后将先前的 umask 值返回。在使用 open()建立新文件时,该参数 mode 并非真正建立文件的权限,而是 (mode&~umask)的权限值。例如,在建立文件时指定文件权限为 0666,通常 umask 值默认为 022,则该文件的真正权限则为 0666&~022=0644,也就是 rw-r--r--返回值,此调用不会有错误值返回。

返回值:为原先系统的 umask 值。

unlink 函数

表头文件: #include<unistd.h>

函数原型: int unlink(const char * pathname);

函数说明: unlink()会删除参数 pathname 指定的文件。如果该文件名为最后连接点,但还有其他进程打开了此文件,则在所有关于此文件的文件描述词皆关闭后才会删除。如果参数 pathname 为一符号连接,则此连接会被删除。

返回值:成功则返回 0,失败返回-1,错误原因存于 errno。

错误代码:

EROFS 文件存在于只读文件系统内;

EFAULT 参数 pathname 指针超出可存取内存空间;

ENAMETOOLONG 参数 pathname 太长;

ENOMEM 核心内存不足;

ELOOP 参数 pathname 有过多符号连接问题;

EIO I/O 存取错误。

utime 函数

表头文件: #include<sys/types.h>

#include<utime.h>

函数原型: int utime(const char *filename, const struct utimbuf *times)

函数说明: utime()用来修改参数 filename 文件所属的 inode 存取时间。

结构 utimbuf 定义如下

```
struct utimbuf{
```

```
time_t actime;
```

```
time_t modtime;
```

```
};
```

返回值: 如果参数 times 为空指针(NULL),则该文件的存取时间和更改时间全部会设为目前时间。执行成功则返回 0,失败返回-1,错误代码存于 errno。

错误代码:

EACCESS 存取文件时被拒绝,权限不足;

ENOENT 指定的文件不存在。

utimes 函数

表头文件: #include<sys/time.h>

函数原型: int utimes(const char *filename, const struct timeval times[2])

函数说明: utimes()用来修改参数 filename 文件所属的 inode 存取时间和修改时间。

结构 timeval 定义如下

```

struct timeval {
long tv_sec;
long tv_usec; /* 微秒*/
};

```

返回值: `utimes()`系统调用类似,但是 `times` 参数指的是数组而不是结构。这个数组的元素是 `timeval` 结构,允许精度为 1 微秒的 specifying 时间戳。执行成功则返回 0。失败返回-1,错误代码存于 `errno`。

错误代码:

`EACCESS` 存取文件时被拒绝,权限不足;

`ENOENT` 指定的文件不存在。

1.11 进程操作接口

`atexit` 函数

表头文件: `#include<stdlib.h>`

函数原型: `int atexit (void (*function)(void));`

函数说明: `atexit()`用来设置一个程序正常结束前调用的函数。当程序通过调用 `exit()`或从 `main` 中返回时,参数 `function` 所指定的函数会先被调用,然后才真正由 `exit()`结束程序。

返回值: 如果执行成功则返回 0, 否则返回一个非 0 值。

`execl` 函数

表头文件: `#include<unistd.h>`

函数原型: `int execl(const char * path,const char * arg,...)`

函数说明: `execl()`用来执行参数 `path` 字符串所代表的文件路径, 接下来的参数代表执行该文件时传递过去的 `argv(0)`、`argv[1]`……, 最后一个参数必须用空指针(NULL)作结束。

返回值: 如果执行成功则函数不会返回, 执行失败则直接返回-1, 失败原因存于 `errno` 中。

`execlp` 函数

表头文件: `#include<unistd.h>`

函数原型: `int execlp(const char * file,const char * arg,……)`

函数说明: `execlp()`会从 `PATH` 环境变量所指的目录中查找符合参数 `file` 的文件名, 找到后便执行该文件, 然后将第二个以后的参数当做该文件的 `argv[0]`、`argv[1]`……, 最后一个参数必须用空指针(NULL)作结束。

返回值: 如果执行成功则函数不会返回, 执行失败则直接返回-1, 失败原因存于 `errno` 中。

错误代码: 参考 `execve()`。

`execv` 函数

表头文件: `#include<unistd.h>`

函数原型: `int execv (const char * path, char * const argv[ ])`

函数说明: `execv()`用来执行参数 `path` 字符串所代表的文件路径, 与 `execl()`不同的地方在于 `execve()`只需两个参数, 第二个参数利用数组指针来传递给执行文件。

返回值: 如果执行成功则函数不会返回, 执行失败则直接返回-1, 失败原因存于 `errno` 中。

错误代码: 请参考 `execve ( )`。

`execve` 函数

表头文件: `#include<unistd.h>`

函数原型: `int execve(const char * filename, char * const argv[ ], char * const envp[ ])`

函数说明: `execve()`用来执行参数 `filename` 字符串所代表的文件路径, 第二个参数是利用数组指针来传递给执行文件, 最后一个参数则为传递给执行文件的新环境变量数组。

返回值: 如果执行成功则函数不会返回, 执行失败则直接返回-1, 失败原因存于 `errno` 中。

错误代码:

EACCES

a) 欲执行的文件不具有用户可执行的权限。

b) 欲执行的文件所属的文件系统是以 `noexec` 方式挂上。

c) 欲执行的文件或 `script` 翻译器非一般文件。

EPERM

a) 进程处于被追踪模式, 执行者并不具有 `root` 权限, 欲执行的文件具有 `SUID` 或 `SGID` 位。

b) 欲执行的文件所属的文件系统是以 `nosuid` 方式挂上, 欲执行的文件具有 `SUID` 或 `SGID` 位元, 但执行者并不具有 `root` 权限。

E2BIG 参数数组过大

ENOEXEC 无法判断欲执行文件的执行文件格式, 有可能是格式错误或无法在此平台执行。

EFAULT 参数 `filename` 所指的字符串地址超出可存取空间范围。

ENAMETOOLONG 参数 `filename` 所指的字符串太长。

ENOENT 参数 `filename` 字符串所指定的文件不存在。

ENOMEM 核心内存不足。

ENOTDIR 参数 `filename` 字符串所包含的目录路径并非有效目录。

EACCES 参数 `filename` 字符串所包含的目录路径无法存取, 权限不足。

ELOOP 过多的符号连接。

ETXTBUSY 欲执行的文件已被其他进程打开而且正把数据写入该文件中。

EIO I/O 存取错误。

ENFILE 已达到系统所允许的打开文件总数。

EMFILE 已达到系统所允许单一进程所能打开的文件总数。

EINVAL 欲执行文件的 `ELF` 执行格式不只一个 `PT_INTERP` 节区。

EISDIR `ELF` 翻译器为一目录。

ELIBBAD `ELF` 翻译器有问题。

execvp 函数

表头文件: `#include <unistd.h>`

函数原型: `int execvp(const char *file, char * const argv [])`

函数说明: `execvp()`会从 `PATH` 环境变量所指的目录中查找符合参数 `file` 的文件名, 找到后便执行该文件, 然后将第二个参数 `argv` 传给该欲执行的文件。

返回值: 如果执行成功则函数不会返回, 执行失败则直接返回-1, 失败原因存于 `errno` 中。

错误代码: 请参考 `execve()`。

exit 函数

表头文件: `#include <stdlib.h>`

函数原型: `void exit(int status)`

函数说明: `exit()`用来正常终结目前进程的执行, 并把参数 `status` 返回给父进程, 而进程所有的缓冲区数据会自动写回并关闭未关闭的文件。

返回值: 无。

`_exit` 函数

表头文件: `#include<unistd.h>`

函数原型: `void _exit(int status)`

函数说明: `_exit()`用来立刻结束目前进程的执行,并把参数 `status` 返回给父进程,并关闭未关闭的文件。此函数调用后不会返回,并且会传递 `SIGCHLD` 信号给父进程,父进程可以由 `wait` 函数取得子进程结束状态。

返回值: 无。

附加说明: `_exit()` 不会处理标准 I/O 缓冲区,如要更新缓冲区请使用 `exit()`。

`vfork` 函数

表头文件: `#include<unistd.h>`

函数原型: `pid_t vfork(void)`

函数说明: `vfork()`会产生一个新的子进程,其子进程会复制父进程的数据与堆栈空间,并继承父进程的用户代码,组代码,环境变量、已打开的文件代码、工作目录和资源限制等。

Linux 使用 `copy-on-write(COW)`技术,只有当其中一进程试图修改欲复制的空间时才会做真正的复制动作,由于这些继承的信息是复制而来,并非指相同的内存空间,因此子进程对这些变量的修改和父进程并不会同步。此外,子进程不会继承父进程的文件锁定和未处理的信号。注意,Linux 不保证子进程会比父进程先执行或晚执行,因此编写程序时要注意死锁或竞争条件的发生。

返回值: 如果 `vfork()`成功则在父进程会返回新建的子进程代码(PID),而在新建的子进程中则返回 0。

如果 `vfork` 失败则直接返回-1,失败原因存于 `errno` 中。

错误代码:

EAGAIN 这将超过系统范围内的进程总数限制,或者超过单个用户执行的进程总数限制。

ENOMEM 内存不足,无法配置核心所需的数据结构空间。

`getpgid` 函数

表头文件: `#include<unistd.h>`

函数原型: `pid_t getpgid(pid_t pid)`

函数说明: `getpgid()`用来取得参数 `pid` 指定进程所属的组织识别码。如果参数 `pid` 为 0,则会取得目前进程的组织识别码。

返回值: 执行成功则返回组织识别码,如果有错误则返回-1,错误原因存于 `errno` 中。

错误代码: ESRCH 找不到符合参数 `pid` 指定的进程。

`getpgrp` 函数

表头文件: `#include<unistd.h>`

函数原型: `pid_t getpgrp(void)`

函数说明: `getpgrp()`用来取得目前进程所属的组织识别码。此函数相当于调用 `getpgid(0)`;

返回值: 返回目前进程所属的组织识别码。

`getpid` 函数

表头文件: `#include<unistd.h>`

函数原型: `pid_t getpid(void)`

函数说明: `getpid()`用来取得目前进程的进程识别码,许多程序利用取到的此值来建立临时文件,以避免临时文件相同带来的问题。

返回值： 目前进程的进程识别码。

getppid 函数

表头文件： #include<unistd.h>

函数原型： pid_t getppid(void)

函数说明： getppid()用来取得目前进程的父进程识别码。

返回值： 目前进程的父进程识别码。

getpriority 函数

表头文件： #include<sys/time.h>

#include<sys/resource.h>

函数原型： int getpriority(int which,int who)

函数说明： getpriority()可用来取得进程、进程组和用户的进程执行优先权。

参数 which 有三种数值，参数 who 则依 which 值有不同定义

which who 代表的意义

PRIO_PROCESS who 为进程识别码

PRIO_PGRP who 为进程的组织识别码

PRIO_USER who 为用户识别码

此函数返回的数值介于-20 至 20 之间，代表进程执行优先权，数值越低代表有较高的优先次序，执行会较频繁。

返回值： 返回进程执行优先权，如有错误发生返回值：则为-1 且错误原因存于 errno。

附加说明： 由于返回值：有可能是-1，因此要同时检查 errno 是否存有错误原因。最好在调用次函数前先清除 errno 变量。

错误代码： ESRCH 参数 which 或 who 可能有错，而找不到符合的进程。EINVAL 参数 which 值错误。

nice 函数

表头文件： #include<unistd.h>

函数原型： int nice(int inc)

函数说明： nice()用来改变进程的进程执行优先顺序。参数 inc 数值越大则优先顺序排在越后面，即表示进程执行会越慢。只有超级用户才能使用负的 inc 值，代表优先顺序排在前面，进程执行会较快。

返回值： 如果执行成功则返回 0，否则返回-1，失败原因存于 errno 中。

错误代码： EPERM 一般用户企图转用负的参数 inc 值改变进程优先顺序。

on_exit 函数

表头文件： #include<stdlib.h>

函数原型： int on_exit(void (*function)(int, void*),void *arg)

函数说明： on_exit()用来设置一个程序正常结束前调用的函数。当程序通过调用 exit()或从 main 中返回时，参数 function 所指定的函数会先被调用，然后才真正由 exit()结束程序。

参数 arg 指针会传给参数 function 函数。

返回值： 如果执行成功则返回 0，否则返回-1，失败原因存于 errno 中。

附加说明： 无。

setpgid 函数

表头文件： #include<unistd.h>

函数原型： `int setpgid(pid_t pid, pid_t pgid)`

函数说明： `setpgid()`将参数 `pid` 指定进程所属的组织识别码设为参数 `pgid` 指定的组织识别码。如果参数 `pid` 为 0，则会用来设置目前进程的组织识别码，如果参数 `pgid` 为 0，则会以目前进程的进程识别码来取代。

返回值： 执行成功则返回组织识别码，如果有错误则返回-1，错误原因存于 `errno` 中。

错误代码：

EINVAL 参数 `pgid` 小于 0；

EPERM 进程权限不足，无法完成调用；

ESRCH 找不到符合参数 `pid` 指定的进程。

`setpgrp` 函数

表头文件： `#include <unistd.h>`

函数原型： `int setpgrp(void)`

函数说明： `setpgrp()`将目前进程所属的组织识别码设为目前进程的进程识别码。此函数相当于调用 `setpgid(0,0)`。

返回值： 执行成功则返回组织识别码，如果有错误则返回-1，错误原因存于 `errno` 中。

`setpriority` 函数

表头文件： `#include <sys/time.h>`

`#include <sys/resource.h>`

函数原型： `int setpriority(int which, int who, int prio)`

函数说明： `setpriority()`可用来设置进程、进程组和用户的进程执行优先权。参数 `which` 有三种数值，参数 `who` 则依 `which` 值有不同定义。

`which who` 代表的意义

PRIO_PROCESS `who` 为进程识别码

PRIO_PGRP `who` 为进程的组识别码

PRIO_USER `who` 为用户识别码

参数 `prio` 介于-20 至 20 之间。代表进程执行优先权，数值越低代表有较高的优先次序，执行会较频繁。

此优先权默认是 0，而只有超级用户（`root`）允许降低此值。

返回值： 执行成功则返回 0，如果有错误发生返回值：则为-1，错误原因存于 `errno`。

ESRCH 参数 `which` 或 `who` 可能有错，而找不到符合的进程。

EINVAL 参数 `which` 值错误。

EPERM 权限不够，无法完成设置。

EACCES 一般用户无法降低优先权。

`system` 函数

表头文件： `#include <stdlib.h>`

函数原型： `int system(const char * string)`

函数说明： `system()`会调用 `fork()`产生子进程，由子进程来调用 `/bin/sh -c string` 来执行参数 `string` 字符串所代表的命令，此命令执行完后随即返回原调用的进程。在调用 `system()`期间 `SIGCHLD` 信号会被暂时搁置，`SIGINT` 和 `SIGQUIT` 信号则会被忽略。

返回值： 如果 `system()`在调用 `/bin/sh` 时失败则返回 127，其他失败原因返回-1。若参数 `string` 为空指针（`NULL`），则返回非零值。如果 `system()`调用成功则最后会返回执行 `shell` 命令后的返回值,但是此返回值也有可能为 `system()`调用 `/bin/sh` 失败所返回的 127，因此最好能再检查 `errno` 来确认执行成功。

附加说明： 在编写具有 `SUID/SGID` 权限的程序时请勿使用 `system()`，`system()`会继承环境变量，通过环境

变量可能会造成系统安全的问题。

wait 函数

表头文件: `#include<sys/types.h>`

`#include<sys/wait.h>`

函数原型: `pid_t wait (int * status)`

函数说明: `wait()`会暂时停止目前进程的执行,直到有信号来到或子进程结束。如果在调用 `wait()`时子进程已经结束,则 `wait()`会立即返回子进程结束状态值。子进程的结束状态值会由参数 `status` 返回,而子进程的进程识别码也会一块返回。如果不在意结束状态值,则参数 `status` 可以设成 `NULL`。子进程的结束状态值请参考 `waitpid()`。

返回值: 如果执行成功则返回子进程识别码(PID),如果有错误发生则返回-1。失败原因存于 `errno` 中。

附加说明: 无。

waitpid 函数

表头文件: `#include<sys/types.h>`

`#include<sys/wait.h>`

函数原型: `pid_t waitpid(pid_t pid,int * status,int options)`

函数说明: `waitpid()`会暂时停止目前进程的执行,直到有信号来到或子进程结束。如果在调用 `waitpid()`时子进程已经结束,则 `waitpid()`会立即返回子进程结束状态值。子进程的结束状态值会由参数 `status` 返回,而子进程的进程识别码也会一块返回。如果不在意结束状态值,则参数 `status` 可以设成 `NULL`。参数 `pid` 为欲等待的子进程识别码,其他数值意义如下:

`pid<-1` 等待进程组识别码为 `pid` 绝对值的任何子进程。

`pid=-1` 等待任何子进程,相当于 `wait()`。

`pid=0` 等待进程组识别码与目前进程相同的任何子进程。

`pid>0` 等待任何子进程识别码为 `pid` 的子进程。

参数 `option` 可以为 0 或下面的 OR 组合

`WNOHANG` 如果没有任何已经结束的子进程则马上返回,不予以等待。

`WUNTRACED` 如果子进程进入暂停执行情况则马上返回,但结束状态不予以理会。子进程的结束状态返回后存于 `status`,以下有几个宏可判别结束情况。

`WIFEXITED(status)`如果子进程正常结束则为非 0 值。

`WEXITSTATUS(status)`取得子进程 `exit()`返回的结束代码,一般会先用 `WIFEXITED` 来判断是否正常结束才能使用此宏。

`WIFSIGNALED(status)`如果子进程是因为信号而结束则此宏值为真。

`WTERMSIG(status)`取得子进程因信号而中止的信号代码,一般会先用 `WIFSIGNALED` 来判断后才使用此宏。

`WIFSTOPPED(status)`如果子进程处于暂停执行情况则此宏值为真。一般只有使用

`WUNTRACED` 时才会有此情况。

`WSTOPSIG(status)`取得引发子进程暂停的信号代码,一般会先用 `WIFSTOPPED` 来判断后才使用此宏。

返回值: 如果执行成功则返回子进程识别码(PID),如果有错误发生则返回-1。失败原因存于 `errno` 中。

1.12 信号处理接口

alarm 函数

表头文件: `#include<unistd.h>`

函数原型: unsigned int alarm(unsigned int seconds)

函数说明: alarm()用来设置信号 SIGALRM 在经过参数 seconds 指定的秒数后传送给目前的进程。如果参数 seconds 为 0, 则之前设置的闹钟会被取消, 并将剩下的时间返回。

返回值: 返回之前闹钟的剩余秒数, 如果之前未设闹钟则返回 0。

kill 函数

表头文件: #include<sys/types.h>

#include<signal.h>

函数原型: int kill(pid_t pid,int sig)

函数说明: kill()可以用来送参数 sig 指定的信号给参数 pid 指定的进程。参数 pid 有几种情况:

pid>0 将信号传给进程识别码为 pid 的进程;

pid=0 将信号传给和目前进程相同进程组的所有进程;

pid=-1 将信号广播传送给系统内所有的进程;

pid<0 将信号传给进程组识别码为 pid 绝对值的所有进程;

返回值: 执行成功则返回 0, 如果有错误则返回-1。

pause 函数

表头文件: #include<unistd.h>

函数原型: int pause(void)

函数说明: pause()会令目前的进程暂停(进入睡眠状态), 直到被信号(signal)所中断。

返回值: 只返回-1。

错误代码: EINTR 有信号到达中断了此函数。

sigaction 函数

表头文件: #include<signal.h>

函数原型: int sigaction(int signum,const struct sigaction *act ,struct sigaction *oldact)

函数说明: sigaction()会依参数 signum 指定的信号编号来设置该信号的处理函数。参数 signum 可以指定 SIGKILL 和 SIGSTOP 以外的所有信号。

如参数结构 sigaction 定义如下

struct sigaction

```
{
void (*sa_handler) (int);
sigset_t sa_mask;
int sa_flags;
void (*sa_restorer) (void);
}
```

sa_handler 此参数和 signal()的参数 handler 相同, 代表新的信号处理函数, 其他意义请参考 signal()。

sa_mask 用来设置在处理该信号时暂时将 sa_mask 指定的信号搁置。

sa_restorer 此参数没有使用。

sa_flags 用来设置信号处理的其他相关操作, 下列的数值可用 OR 运算(|)组合。

SA_NOCLDSTOP: 如果参数 signum 为 SIGCHLD, 则当子进程暂停时并不会通知父进程。

SA_ONESHOT/SA_RESETHAND:当调用新的信号处理函数前, 将此信号处理方式改为系统预设的方式。

SA_RESTART:被信号中断的系统调用会自行重启。

SA_NOMASK/SA_NODEFER:在处理此信号未结束前不理睬此信号的再次到来。

如果参数 `oldact` 不是 `NULL` 指针，则原来的信号处理方式会由此结构 `sigaction` 返回。

返回值： 执行成功则返回 0，如果有错误则返回-1。

错误代码：

`EINVAL` 参数 `sigum` 不合法， 或是企图拦截 `SIGKILL/SIGSTOPSIGKILL` 信号；

`EFAULT` 参数 `act`，`oldact` 指针地址无法存取；

`sigaddset` 函数

表头文件： `#include<signal.h>`

函数原型： `int sigaddset(sigset_t *set,int sigum)`

函数说明： `sigaddset()`用来将参数 `sigum` 代表的信号加入至参数 `set` 信号集里。

返回值： 执行成功则返回 0，如果有错误则返回-1。

错误代码： `EINVAL` 参数 `sigum` 非合法的信号编号。

`sigdelset` 函数

表头文件： `#include<signal.h>`

函数原型： `int sigdelset(sigset_t *set,int sigum)`

函数说明： `sigdelset()`用来将参数 `sigum` 代表的信号从参数 `set` 信号集里删除。

返回值： 执行成功则返回 0，如果有错误则返回-1。

错误代码： `EINVAL` 参数 `sigum` 非合法的信号编号。

`sigemptyset` 函数

表头文件： `#include<signal.h>`

函数原型： `int sigemptyset(sigset_t *set)`

函数说明： `sigemptyset()`用来将参数 `set` 信号集初始化并清空。

返回值： 执行成功则返回 0，如果有错误则返回-1。

错误代码： `EINVAL` 参数 `sigum` 非合法的信号编号。

`sigfillset` 函数

表头文件： `#include<signal.h>`

函数原型： `int sigfillset(sigset_t *set)`

函数说明： `sigfillset()`用来将参数 `set` 信号集初始化，然后把所有的信号加入到此信号集里。

返回值： 执行成功则返回 0，如果有错误则返回-1。

错误代码： `EINVAL` 参数 `sigum` 非合法的信号编号。

`sigismember` 函数

表头文件： `#include<signal.h>`

函数原型： `int sigismember(const sigset_t *set,int sigum)`

函数说明： `sigismember()`用来测试参数 `sigum` 代表的信号是否已加入至参数 `set` 信号集里。如果信号集里已有该信号则返回 1，否则返回 0。

返回值： 信号集已有该信号则返回 1，没有则返回 0。如果有错误则返回-1。

错误代码： `EINVAL` 参数 `sigum` 非合法的信号编号。

`signal` 函数

表头文件： `#include<signal.h>`

函数原型: void (*signal(int signum,void(* handler)(int)))(int)

函数说明: signal()会依参数 signum 指定的信号编号来设置该信号的处理函数。当指定的信号到达时就会跳转到参数 handler 指定的函数执行。如果参数 handler 不是函数指针,则必须是下列两个常数之一:

SIG_IGN 忽略参数 signum 指定的信号。

SIG_DFL 将参数 signum 指定的信号重设为核心预设的信号处理方式。

返回值: 返回先前的信号处理函数指针,如果有错误则返回 SIG_ERR(-1)。

附加说明: 在信号发生跳转到自定的 handler 处理函数执行后,系统会自动将此处理函数换回原来系统预设的处理方式,如果要改变此操作请改用 sigaction()。

错误代码: EINVAL 参数 signum 非合法的编号。

sigpending 函数

表头文件: #include<signal.h>

函数原型: int sigpending(sigset_t *set)

函数说明: sigpending()会将被搁置的信号集合由参数 set 指针返回。

返回值: 执行成功则返回 0,如果有错误则返回-1。

错误代码: EFAULT 参数 set 指针地址无法存取。

sigprocmask 函数

表头文件: #include<signal.h>

函数原型: int sigprocmask(int how,const sigset_t *set,sigset_t * oldset)

函数说明: sigprocmask()可以用来改变目前的信号遮罩,其操作依参数 how 来决定。

SIG_BLOCK 新的信号遮罩由目前的信号遮罩和参数 set 指定的信号遮罩作联集。

SIG_UNBLOCK 将目前的信号遮罩删除掉参数 set 指定的信号遮罩。

SIG_SETMASK 将目前的信号遮罩设成参数 set 指定的信号遮罩。

如果参数 oldset 不是 NULL 指针,那么目前的信号遮罩会由此指针返回。

返回值: 执行成功则返回 0,如果有错误则返回-1。

错误代码:

EFAULT 参数 set, oldset 指针地址无法存取;

EINVAL 在 how 中指定的值是无效的,或者内核不支持在 sigsetsize 中传递的大小

sleep 函数

表头文件: #include<unistd.h>

函数原型: unsigned int sleep(unsigned int seconds)

函数说明: sleep()会令目前的进程暂停,直到达到参数 seconds 所指定的时间,或是被信号所中断。

返回值: 若进程暂停到参数 seconds 所指定的时间则返回 0,若有信号中断则返回剩余秒数。

ferror 函数

表头文件: #include<stdio.h>

函数原型: int ferror(FILE *stream)

函数说明: ferror()用来检查参数 stream 所指定的文件流是否发生了错误情况,如有错误发生则返回非 0 值。

返回值: 如果文件流有错误发生则返回非 0 值。

perror 函数

表头文件: `#include<stdio.h>`

函数原型: `void perror(const char *s)`

函数说明: `perror()`用来将上一个函数发生错误的原因输出到标准错误(stderr)。参数 `s` 所指的字符串会先打印出,后面再加上错误原因字符串。此错误原因依照全局变量 `errno` 的值来决定要输出的字符串。

返回值: 无。

`strerror` 函数

表头文件: `#include<string.h>`

函数原型: `char *strerror(int errnum)`

函数说明: `strerror()`用来依参数 `errnum` 的错误代码来查询其错误原因的描述字符串,然后将该字符串指针返回。

返回值: 返回描述错误原因的字符串指针。

`mkfifo` 函数

表头文件: `#include<sys/types.h>`

`#include<sys/stat.h>`

函数原型: `int mkfifo(const char *pathname,mode_t mode)`

函数说明: `mkfifo()`会依参数 `pathname` 建立特殊的 FIFO 文件,该文件必须不存在,而参数 `mode` 为该文件的权限 (`mode%~umask`),因此 `umask` 值也会影响到 FIFO 文件的权限。

`Mkfifo()`建立的 FIFO 文件其他进程都可以用读写一般文件的方式存取。当使用 `open()`来打开 FIFO 文件时,`O_NONBLOCK` 旗标会有影响:

a) 当使用 `O_NONBLOCK` 旗标时,打开 FIFO 文件来读取的操作会立刻返回,但是若还没有其他进程打开 FIFO 文件来读取,则写入的操作会返回 `ENXIO` 错误代码。

b) 没有使用 `O_NONBLOCK` 旗标时,打开 FIFO 来读取的操作会等到其他进程打开 FIFO 文件来写人才正常返回。同样地,打开 FIFO 文件来写入的操作会等到其他进程打开 FIFO 文件来读取后才正常返回。

返回值: 若成功则返回 0,否则返回-1,错误原因存于 `errno` 中。

错误代码:

`EACCESS` 参数 `pathname` 所指定的目录路径无可执行的权限;

`EDQUOT` 用户在文件系统上的磁盘块或索引节点的配额已经用完

`EEXIST` 参数 `pathname` 所指定的文件已存在;

`ENAMETOOLONG` 参数 `pathname` 的路径名称太长;

`ENOENT` 参数 `pathname` 包含的目录不存在;

`ENOSPC` 文件系统的剩余空间不足;

`ENOTDIR` 参数 `pathname` 路径中的目录存在但却非真正的目录;

`EROFS` 参数 `pathname` 指定的文件存在于只读文件系统内;

`EBADF` 参数 `dirfd` 不是一个有效的文件描述符;

`ENOTDIR` `pathname` 是一个相对路径, `dirfd` 是一个引用文件而不是目录的文件描述符。

`pclose` 函数

表头文件: `#include<stdio.h>`

函数原型: `int pclose(FILE *stream)`

函数说明: `pclose()`用来关闭由 `popen` 所建立的管道及文件指针。参数 `stream` 为先前由 `popen()`所返回的文件指针。

返回值: 返回子进程的结束状态。如果有错误则返回-1,错误原因存于 `errno` 中。

错误代码：ECHILD pclose()无法取得子进程的结束状态;

EINVAL 类型参数无效

pipe 函数

表头文件：#include<unistd.h>

函数原型：int pipe(int filedes[2])

函数说明：pipe()会建立管道，并将文件描述词由参数 filedes 数组返回。filedes[0]为管道里的读取端，filedes[1]则为管道的写入端。

返回值：若成功则返回零，否则返回-1，错误原因存于 errno 中。

错误代码：

EMFILE 进程已用完文件描述词最大量；

ENFILE 系统已无文件描述词可用；

ENFILE 用户可以分配给管道的内存已经达到了硬限制，调用者没有特权

EFAULT 参数 filedes 数组地址不合法。

EINVAL 无效的参数标志

popen 函数

表头文件：#include<stdio.h>

函数原型：FILE *popen(const char *command,const char *type)

函数说明：popen()会调用 fork()产生子进程，然后从子进程中调用/bin/sh -c 来执行参数 command 的指令。参数 type 可使用“r”代表读取，“w”代表写入。依照此 type 值，popen()会建立管道连到子进程的标准输出设备或标准输入设备，然后返回一个文件指针。随后进程便可利用此文件指针来读取子进程的输出设备或是写入到子进程的标准输入设备中。此外，所有使用文件指针(FILE*)操作的函数也都可以使用，除了 fclose()以外。

返回值：若成功则返回文件指针，否则返回 NULL，错误原因存于 errno 中。

错误代码：EINVAL 参数 type 不合法。

ECHILD pclose()无法取得子进程的结束状态

注意事项：在编写具 SUID/SGID 权限的程序时请尽量避免使用 popen()，popen()会继承环境变量，通过环境变量可能会造成系统安全的问题。

1.13 环境变量操作接口

getenv 函数

表头文件：#include<stdlib.h>

函数原型：char *getenv(const char *name)

函数说明：getenv()用来取得参数 name 环境变量的内容。参数 name 为环境变量的名称，如果该变量存在则会返回指向该内容的指针。环境变量的格式为 name = value。

返回值：执行成功则返回指向该内容的指针，找不到符合的环境变量名称则返回 NULL。

putenv 函数

表头文件：#include<stdlib.h>

函数原型：int putenv(const char *string)

函数说明：putenv()用来改变或增加环境变量的内容。参数 string 的格式为 name = value，

如果该环境变量原先存在，则变量内容会依参数 string 改变，否则此参数内容会成为新的环境变量。

返回值：执行成功则返回 0，有错误发生则返回-1。

错误代码: ENOMEM 内存不足,无法配置新的环境变量空间。

setenv 函数

表头文件: #include<stdlib.h>

函数原型: int setenv(const char *name,const char * value,int overwrite)

函数说明: setenv()用来改变或增加环境变量的内容。参数 name 为环境变量名称字符串。参数 value 则为变量内容,参数 overwrite 用来决定是否要改变已存在的环境变量。如果 overwrite 不为 0,而该环境变量原已有内容,则原内容会被改为参数 value 所指的变量内容。如果 overwrite 为 0,且该环境变量已有内容,则参数 value 会被忽略。

返回值: 执行成功则返回 0,有错误发生时返回-1。

错误代码: ENOMEM 内存不足,无法配置新的环境变量空间。

1.14 终端控制操作接口

getopt 函数

表头文件: #include<unistd.h>

函数原型: int getopt(int argc,char * const argv[],const char * optstring)

函数说明: getopt()用来分析命令行参数。参数 argc 和 argv 是由 main()传递的参数个数和内容。参数 optstring 则代表欲处理的选项字符串。此函数会返回在 argv 中下一个的选项字母,此字母会对应参数 optstring 中的字母。如果选项字符串里的字母后接着冒号 “:”,则表示还有相关的参数,全局变量 optarg 即会指向此额外参数。如果 getopt()找不到符合的参数则会印出错误信息。如果不希望 getopt()印出错误信息,则只要将全局变量 opterr 设为 0 即可。

返回值: 如果找到符合的参数则返回此参数字母,如果参数不包含在参数 optstring 的选项字母则返回 “?” 字符,分析结束则返回-1。

isatty 函数

表头文件: #include<unistd.h>

函数原型: int isatty(int desc)

函数说明: 如果参数 desc 所代表的文件描述词为一终端机则返回 1,否则返回 0。

返回值: 如果文件为终端机则返回 1,否则返回 0。

ttyname 函数

表头文件: #include<unistd.h>

函数原型: char * ttyname(int desc)

函数说明: 如果参数 desc 所代表的文件描述词为一终端机,则会将此终端机名称由一字符串指针返回,否则返回 NULL。

返回值: 如果成功则返回指向终端机名称的字符串指针,有错误情况发生时则返回 NULL。

1.15 输入输出接口

fprintf 函数

表头文件: #include<stdio.h>

函数原型: `int fprintf(FILE * stream, const char * format,.....)`

函数说明: `fprintf()`会根据参数 `format` 字符串来转换并格式化数据,然后将结果输出到参数 `stream` 指定的文件中,直到出现字符串结束('\0')为止。

返回值: 关于参数 `format` 字符串的格式请参考 `printf()`。成功则返回实际输出的字符数,失败则返回-1,错误原因存于 `errno` 中。

`fscanf` 函数

表头文件: `#include<stdio.h>`

函数原型: `int fscanf(FILE * stream ,const char *format,...)`

函数说明: `fscanf()`会从参数 `stream` 的文件流中读取字符串,再根据参数 `format` 字符串来转换并格式化数据。格式转换形式请参考 `scanf()`。转换后的结构存于对应的参数内。

返回值: 成功则返回参数数目,失败则返回-1,错误原因存于 `errno` 中。

附加说明: 无。

`printf` 函数

表头文件: `#include<stdio.h>`

函数原型: `int printf(const char * format,.....)`

函数说明: `printf()`会根据参数 `format` 字符串来转换并格式化数据,然后将结果写出到标准输出设备,直到出现字符串结束('\0')为止。参数 `format` 字符串可包含下列三种字符类型:

- a) 一般文本,伴随直接输出;
- b) ASCII 控制字符,如 `\t`、`\n` 等;
- c) 格式转换字符。

格式转换为一个百分比符号(%)及其后的格式字符所组成。一般而言,每个%符号在其后都必需有一 `printf()` 的参数与之相呼应(只有当%%转换字符出现时会直接输出%字符),而欲输出的数据类型必须与其相对应的转换字符类型相同。

`Printf()`格式转换的一般形式如下

`%(flags)(width)(.prec)type`

以中括号括起来的参数为选择性参数,而%与 `type` 则是必要的。底下先介绍 `type` 的几种形式整数

`%d` 整数的参数会被转成一有符号的十进制数字;

`%u` 整数的参数会被转成一无符号的十进制数字;

`%o` 整数的参数会被转成一无符号的八进制数字;

`%x` 整数的参数会被转成一无符号的十六进制数字,并以小写 `abcdef` 表示;

`%X` 整数的参数会被转成一无符号的十六进制数字,并以大写 `ABCDEF` 表示浮点型数;

`%f` `double` 型的参数会被转成十进制数字,并取到小数点以下六位,四舍五入;

`%e` `double` 型的参数以指数形式打印,有一个数字会在小数点前,六位数字在小数点后,而在指数部分会以小写的 `e` 来表示;

`%E` 与 `%e` 作用相同,唯一区别是指数部分将以大写的 `E` 来表示;

`%g` `double` 型的参数会自动选择以 `%f` 或 `%e` 的格式来打印,其标准是根据欲打印的数值及所设置的有效位数来决定;

`%G` 与 `%g` 作用相同,唯一区别在以指数形态打印时会选择 `%E` 格式。

字符及字符串

`%c` 整型数的参数会被转成 `unsigned char` 型打印出;

`%s` 指向字符串的参数会被逐字输出,直到出现 `NULL` 字符为止;

`%p` 如果是参数是“`void *`”型指针则使用十六进制格式显示。

prec 有几种情况

- a) 正整数的最小位数。
- b) 在浮点型数中代表小数位数
- c) 在 %g 格式代表有效位数的最大值。
- d) 在 %s 格式代表字符串的最大长度。
- e) 若为 × 符号则代表下个参数值为最大长度。

width 为参数的最小长度，若此栏并非数值，而是 * 符号，则表示以下一个参数当做参数长度。

flags 有下列几种情况

#NAME?

+ 一般在打印负数时，printf () 会加印一个负号，整数则不加任何负号。此旗标会使得在打印正数前多一个正号 (+)。

此旗标会根据其后转换字符的不同而有不同含义。当在类型为 o 之前 (如 %#o)，则会在打印八进制数值前多印一个 o。

而在类型为 x 之前 (%#x) 则会在打印十六进制数前多印 '0x'，在型态为 e、E、f、g 或 G 之前则会强迫数值打印小数点。在类型为 g 或 G 之前时则同时保留小数点及小数位数末尾的零。

0 当有指定参数时，无数字的参数将补上 0。默认是关闭此旗标，所以一般会打印出空白字符。

返回值： 如果函数执行成功，则返回所打印的字符总数，如果函数执行失败，则返回一个负数。

scanf 函数

表头文件： #include<stdio.h>

函数原型： int scanf(const char * format,.....)

函数说明： scanf() 会将输入的数据根据参数 format 字符串来转换并格式化数据。Scanf()

格式转换的一般形式如下

%[*][size][l][h]type 以中括号括起来的参数为选择性参数，而 % 与 type 则是必要的。

* 代表该对应的参数数据忽略不保存。

size 为允许参数输入的数据长度。

l 输入的数据数值以 long int 或 double 型保存。

h 输入的数据数值以 short int 型保存。

下面介绍 type 的几种形式：

%d 输入的数据会被转成一有符号的十进制数字 (int)。

%i 输入的数据会被转成一有符号的十进制数字，若输入数据以 “0x” 或 “0X” 开头代表转换。十六进制数字，若以 “0” 开头则转换八进制数字，其他情况代表十进制。

%O 输入的数据会被转换成一无符号的八进制数字。

%u 输入的数据会被转换成一无符号的正整数。

%x 输入的数据为无符号的十六进制数字，转换后存于 unsigned int 型变量。

%X 同 %x。

%f 输入的数据为有符号的浮点型数，转换后存于 float 型变量。

%e 同 %f。

%E 同 %f。

%g 同 %f。

%s 输入数据为以空格字符为终止的字符串。

%c 输入数据为单一字符。

[] 读取数据但只允许括号内的字符。如 [a-z]。

[^] 读取数据但不允许中括号的 ^ 符号后的字符出现，如 [^0-9]。

返回值： 成功则返回参数数目，失败则返回-1，错误原因存于 `errno` 中。

`sprintf` 函数

表头文件： `#include<stdio.h>`

函数原型： `int sprintf( char *str,const char * format,.....)`

函数说明： `sprintf()`会根据参数 `format` 字符串来转换并格式化数据，然后将结果复制到参数 `str` 所指的字符串数组，直到出现字符串结束(‘\0’)为止。关于参数 `format` 字符串的格式请参考 `printf()`。

返回值： 如果成功，则返回写入的字符总数，不包括字符串追加在字符串末尾的空字符。如果失败，则返回一个负数。

附加说明： 使用此函数得留意堆栈溢出，或改用 `snprintf()`。

`sscanf` 函数

表头文件： `#include<stdio.h>`

函数原型： `int sscanf( const char *str,const char * format,.....)`

函数说明： `sscanf()`会将参数 `str` 的字符串根据参数 `format` 字符串来转换并格式化数据。格式转换形式请参考 `scanf()`。转换后的结果存于对应的参数内。

返回值： 函数将返回成功赋值的字段个数；返回值不包括已读取但未赋值的字段个数。 返回值为 0 表示没有将任何字段赋值。 如果在第一次读取之前到达字符串结尾，则返回 EOF

`vfprintf` 函数

表头文件： `#include<stdio.h>`

`#include<stdarg.h>`

函数原型： `int vfprintf(FILE *stream,const char * format,va_list ap)`

函数说明： `vfprintf()`会根据参数 `format` 字符串来转换并格式化数据，然后将结果输出到参数 `stream` 指定的文件中，直到出现字符串结束(‘\0’)为止。关于参数 `format` 字符串的格式请参考 `printf()`。

返回值： 成功则返回实际输出的字符数，如果遇到输出错误，则返回一个负值。

`vfscanf` 函数

表头文件： `#include<stdio.h>`

函数原型： `int vfscanf(FILE * stream,const char * format ,va_list ap)`

函数说明： `vfscanf()`会自参数 `stream` 的文件流中读取字符串，再根据参数 `format` 字符串来转换并格式化数据。格式转换形式请参考 `scanf()`。转换后的结果存于对应的参数内。

返回值： 成功则返回参数数目，失败则返回-1，错误原因存于 `errno` 中。

`vprintf` 函数

表头文件： `#include<stdio.h>`

`#include<stdarg.h>`

函数原型： `int vprintf(const char * format,va_list ap)`

函数说明： `vprintf()`作用和 `printf()`相同，参数 `format` 格式也相同。`va_list` 为不定个数的参数列。

返回值： 成功则返回实际输出的字符数，如果遇到输出错误，则返回一个负值。

`vscanf` 函数

表头文件： `#include<stdio.h>`

`#include<stdarg.h>`

函数原型: `int vscanf( const char * format,va_list ap)`

函数说明: `vscanf()`会将输入的数据根据参数 `format` 字符串来转换并格式化数据。格式转换形式请参考 `scanf()`。转换后的结果存于对应的参数内。

返回值: 成功则返回参数数目, 失败则返回-1, 错误原因存于 `errno` 中。

`vsprintf` 函数

表头文件: `#include<stdio.h>`

函数原型: `int vsprintf( char * str,const char * format,va_list ap);`

函数说明: `vsprintf()`会根据参数 `format` 字符串来转换并格式化数据, 然后将结果复制到参数 `str` 所指的字符串数组, 直到出现字符串结束(' \0')为止。关于参数 `format` 字符串的格式请参考 `printf()`。

返回值: 成功则返回参数 `str` 字符串长度, 如果遇到输出错误, 则返回一个负值。

`vsscanf` 函数

表头文件: `#include<stdio.h>`

函数原型: `int vsscanf(const char * str,const char * format,va_list ap)`

函数说明: `vsscanf()`会将参数 `str` 的字符串根据参数 `format` 字符串来转换并格式化数据。

返回值: 成功则返回参数数目, 失败则返回-1, 错误原因存于 `errno` 中。

1.16 图形编程接口

1.16.1 GTK 图形编程

`gtk_init` 函数

表头文件: `#<gtk/gtk.h>`

函数原型: `void gtk_init( int *argc , char ***argv);`

函数说明: 在每个 GTK 应用程序中都会被首先调用, 是 GTK+程序的初始化函数, 它将设置 GTK+、类系统和与窗口环境的连接, 并解析一些标准命令行选项。

返回值: 无。

`gtk_window_new` 函数

表头文件: `#<gtk/gtk.h>`

函数原型: `GtkWidget* gtk_window_new( GtkWindowType type);`

函数说明: 该函数用来创建新的 `GtkWindow`, 可以包含其他小部件的顶级窗口。

返回值: 函数 `gtk_window_new` 创建一个窗口并返回这个窗口的控件指针。

`gtk_widget_show` 函数

表头文件: `#<gtk/gtk.h>`

函数原型: `void gtk_widget_show(GtkWidget *widget);`

函数说明: 显示一个 `widget` 的本质, 就是将 `widget` 实例化, 并将其映射。

返回值: 无。

`g_signal_connect` 函数

表头文件: `#<gtk/gtk.h>`

函数原型: `g_signal_connect( gpointer *object,`

```
const gchar *name,  
GCallback func,  
gpointer func_data );
```

函数说明： 该函数用来设置信号和信号处理函数之间的连接，第一个参数是要发出信号的构件，第二个参数是想要连接的信号的名称，第三个参数是信号被捕获时所要调用的函数，第四个参数是你想传递给这个函数的数据，第三个参数指定的函数叫回调函数，回调函数有两种形式：

信号回调函数，其形式是：

```
callback_func( GtkWidget *widget,gpointer callback_data );
```

事件回调函数，其形式是：

```
callback_func( GtkWidget *widget,  
GdkEvent *event,  
gpointer callback_data );
```

返回值： 无。

附 加 说 明 ： 与 g_signal_connect 相 关 连 的 另 外 的 函 数 是

g_signal_connect_swapped(gpointer *object,const gchar *name,GCallback func, gpointer *slot_object), g_signal_connect_swapped() 和 g_signal_connect(),只是回调函数只用一个参数,一个指向 GTK 对象的指针。所以当使用这个函数连接信号时，回调函数应该是这样的形式 void callback_func(GObject *object), 因为交换参数避免了编写包装函数的需要。

gtk_box_pack_start 函数

表头文件： #<gtk/gtk.h>

```
函数原型： void gtk_box_pack_start(GtkBox *box,  
GtkWidget *child,  
gboolean expand,  
gboolean fill,  
guint padding);
```

函数说明：该函数将对象从上到下组装到纵向盒中，或者从左到右组装到横向盒中，第一个参数要把对象组装进去的盒，第二个就是该对象，expand 参数是用来控制构件在盒中是充满所有多余空间，这样盒会扩展到充满所有分配给它的空间（TURE）；还是盒收缩到仅仅符合构件的大小（FALSE）。设置 expand 为 FALSE 将允许你向左或向右对齐你的构件。否则，它们会在盒中展开。fill 参数在 gtk_box_pack 中控制多余空间是分配给对象本身（TRUE），还是让多余空间围绕在这些对象周围分布（FALSE）。它只有在 expand 参数也为 TRUE 时才会生效。

返回值： 无。

附加说明： 使用这些函数允许我们调整自己的构件向左或向右对齐，同时也可以混入一些其它的方法来达到我们想要的设计效果。被组装的对象可以是另一个容器或构件。事实上，许多构件本身就是容器，包括按钮，只不过我们通常在按钮中只放入一个标签。

gtk_box_pack_end 函数

表头文件： #<gtk/gtk.h>

函 数 原 型：

```
void gtk_box_pack_end(GtkBox *box,  
GtkWidget *child,  
gboolean expand,
```

gboolean fill,
guint padding);

函数说明： 该函数将对象从下到上组装到纵向盒中，或者从右到左组装到横向盒中，第一个参数是你要把对象组装进去的盒，第二个就是该对象，expand 参数是用来控制构件在盒中是充满所有多余空间这样盒会扩展到充满所有分配给它的空间（TURE）；还是盒收缩到仅仅符合构件的大小（FALSE）。设置 expand 为 FALSE 将允许你向左或向右对齐你的构件。否则，它们会在盒中展开。fill 参数在 gtk_box_pack 中控制多余空间是分配给对象本身（TRUE），还是让多余空间围绕在这些对象周围分布（FALSE）。它只有在 expand 参数也为 TRUE 时才会生效。

返回值： 无。

附加说明： 使用这些函数允许我们调整自己的构件向左或向右对齐，同时也可以混入一些其它的方法来达到我们想要的设计效果。被组装的对象可以是另一个容器或构件。事实上，许多构件本身就是容器，包括按钮，只不过我们通常在按钮中只放入一个标签。

gtk_hbox_new 函数

表头文件： #<gtk/gtk.h>

函数原型： GtkWidget *gtk_hbox_new (gboolean homogeneous, gint spacing);

函数说明： 该函数用来创建一个新盒，与这个函数相似的另一个函数是：gtk_vbox_new(), gtk_hbox_new() 的 homogeneous 参数（对于 gtk_vbox_new()也是一样）控制是否盒里的每个对象具有相同的大小（例如，在横向盒中等宽，或在纵向盒中等高）。若它被设置，gtk_box_pack()常规函数的 expand 参数就被忽略了，它本质上总被开启。

spacing（当盒被创建时设置）和 padding（当元素被组装时设置）有什么区别呢？Spacing 是加在对象之间，而 padding 加在对象的每一边。

返回值： 该函数返回一个 GtkWidget 的结构体指针。

附加说明： 无。

gtk_table_new 函数

表头文件： #<gtk/gtk.h>

函数原型： GtkWidget *gtk_table_new(guint rows,
guint columns,
gboolean homogeneous);

函数说明： 该函数通过建立表来组装构件。第一个参数是表中要安排的行的数量，而第二个，显然，就是列的数量。homogeneous 参数跟表格框(table's boxes)的大小处理有关。如果 homogeneous 是 TRUE，所有表格框的大小都将调整为表中最大构件的大小。如果

homogeneous 为 FALSE，每个表格框将会按照同行中最高的构件，与同列中最宽的构件来决定自身的大小。

返回值： 该函数返回一个 GtkWidget 的结构体指针。

附加说明： 行与列为从 0 到 n 编号，而 n 是我们在调用 gtk_table_new 时所指定的值。

gtk_table_attach 函数

表头文件： #<gtk/gtk.h>

函数原型： void gtk_table_attach(GtkTable *table,
GtkWidget *child,
guint left_attach,
guint right_attach,
guint top_attach,

```
guint bottom_attach,  
GtkAttachOptions xoptions,  
GtkAttachOptions yoptions,  
guint xpadding, guint ypadding );
```

函数说明： 该函数用来向表格框中放置构件，第一个参数（"table"）是已经创建的表，第二个参数（"child"）是想放进表里的构件。left_attach 和 right_attach 参数指定构件放置的位置，并使用多少框来放。如果想在 2x2 的表中的右下表项(table entry)处放入一个按钮，并且想让它只充满这个项，则 left_attach 应该为 = 1, right_attach = 2, top_attach = 1, bottom_attach = 2。如果想让一个构件占据这个 2x2 表的整个顶行，就用 left_attach= 0, right_attach = 2, top_attach = 0, bottom_attach = 1。xoptions 及 yoptions 是用来指定组装时的选项，可以通过使用“位或”运算以允许多重选项。这些选项是：

GTK_FILL

如果表框大于构件，同时 GTK_FILL 被指定，该构件会扩展开以使用所有可用的空间。

GTK_SHRINK

如果表构件分配到的空间比需求的小（通常是用户在改变窗口大小的时候），那么构件将会推到窗口的底部以外的区域，无法看见。如果 GTK_SHRINK 被指定了，构件将和表一起缩小。

GTK_EXPAND

这会导致表扩展以用完窗口中所有的保留空间。

Padding 和在盒(boxes)中的一样，在构件的周围产生一个指定像素的空白区域。

返回值： 无。

附加说明： 无。

1.17 JDK 编程接口

getApplet 方法

表头文件： import java.applet. AppletContext

函数原型： Applet getApplet(String name)

函数说明： 通过给定名称找到并返回此 applet 上下文表示的文档中的 applet。名称可以在 HTML 标记中通过设置 name 属性来设置，参数 name 为 applet 名称；

返回值： 具有给定名称的 applet；如果未找到，则返回 null；

附加说明： 无

getApplets 方法

表头文件： import java.applet. AppletContext

函数原型： Enumeration<Applet> getApplets()

函数说明： 找到此 applet 上下文表示的文档中的所有 applet

返回值： 此 applet 上下文表示的文档中所有 applet 的枚举

附加说明： 无

showDocument 方法

表头文件： import java.applet. AppletContext

函数原型： void showDocument(URL url)

函数说明： 请求浏览器或 applet viewer 显示 url 参数指示的 Web 页面。浏览器或 applet viewer 确定使用哪个窗口或窗体来显示 Web 页面。非浏览器的 applet 上下文可以忽略此方法；

返回值: url – 给出文档位置的绝对 URL

附加说明: 无

showDocument 方法

表头文件: import java.applet. AppletContext

函数原型: void showDocument(URL url, String target)

函数说明: 请求浏览器或 applet viewer 显示由 url 参数所指示的 Web 页。target 参数指示在哪个 HTML 窗体中显示文档。target 参数的解释如下:

"_self" 在包含 applet 的窗口和窗体中显示。

"_parent" 在 applet 的父窗体中显示。如果 applet 的窗体没有父窗体, 执行与"_self"相同的动作。

"_top" 在 applet 窗口的顶层窗体中显示。如果 applet 窗体是顶层窗体, 执行与"_self"相同的动作。

"_blank" 在一个新的、未命名的顶层窗口中显示。

name 在名为 name 的窗体或窗口中显示。如果名为 name 的 target 尚未存在, 将创建一个具有指定名称的新顶层窗口, 文档将在该窗口中显示。

applet viewer 或浏览器可以随意地忽略 showDocument。

返回值: 返回空

附加说明: 无

showStatus 方法

表头文件: import java.applet. AppletContext

函数原型: void showStatus(String status)

函数说明: 请求参数字符串显示在“状态窗口”中。很多浏览器和 applet viewer 提供这种窗口, 应用程序可以通过该窗口将其当前状态通知给用户, status 表示在状态窗口中显示的字符串;

返回值: 返回空值

附加说明: 无

setStream 方法

表头文件: import java.applet. AppletContext

函数原型: void setStream(String key, InputStream stream) throws IOException

函数说明: 在此 applet 上下文中用指定的键关联指定的流。如果 applet 上下文以前包含了一个此键的映射关系, 那么将替换旧值。出于安全性考虑, 每个代码基都存在流和键的映射。换句话说, 一个代码基中的 applet 不能访问不同代码基中的 applet 创建的流。

key – 指定值要关联的键。stream – 指定键要关联的流。如果此参数为 null, 则在此 applet 上下文中将移除指定键;

返回值: 返回空值

附加说明: 无

getStream 方法

表头文件: import java.applet. AppletContext

函数原型: InputStream getStream(String key)

函数说明: 返回此 applet 上下文中指定键所关联的流。如果 applet 上下文中不包含此键对应的流, 则返

回 null。由于安全性考虑，每个代码基都存在流和键的映射。换句话说，一个代码基中的 applet 不能访问不同代码基中的 applet 创建的流。key - 要返回其关联流的键；

返回值：此 applet 上下文将键映射到的流

附加说明：无

getStreamKeys 方法

表头文件： import java.applet.*

函数原型： Iterator<String> getStreamKeys()

函数说明： 找到此 applet 上下文中所有流对应的键。由于安全性考虑，每个代码基都存在流和键的映射。换句话说，一个代码基中的 applet 不能访问不同代码基中的 applet 创建的流；

返回值：返回此 applet 上下文中所有流名称的迭代器

附加说明：无

isActive 方法

表头文件： import java.applet. AppletStub

函数原型： boolean isActive()

函数说明：确定 applet 是否处于激活状态。仅在 applet 的 start 方法被调用前，applet 才处于激活状态。仅在 applet 的 stop 方法被调用前，applet 才变成非激活状态；

返回值：如果 applet 处于激活状态，则返回 true；否则返回 false；

附加说明：无

getDocumentBase 方法

表头文件： import java.applet. AppletStub

函数原型： URL getDocumentBase()

函数说明：获取嵌入 applet 的文档的 URL。例如，假定 applet 包含在下面的文档中：

<http://java.sun.com/products/jdk/1.2/index.html>

则该文档基于：<http://java.sun.com/products/jdk/1.2/index.html>

返回值：包含 applet 的文档的 URL

附加说明：无

getParameter 方法

表头文件： import java.applet. AppletStub

函数原型： String getParameter(String name)

函数说明：返回 HTML 标记中命名参数的值。例如，如果 applet 被指定为：

<applet code="Clock" width=50 height=50>

<param name="Color" value="blue">

</applet>

则对 getParameter("Color") 的调用将返回值"blue"；

返回值：命名参数的值，如果没有设置，则为 null

附加说明：无

appletResize 方法

表头文件： import java.applet. AppletStub

函数原型： void appletResize(int width, int height)

函数说明：当 applet 想要重新调整大小时调用，width - 为 applet 新请求的宽度，height - 为 applet 新请求的高度；

返回值：返回空值

附加说明：无

2 用户操作接口说明

2.1 终端命令

arch

功能说明：显示系统体系结构

语法：arch

at

功能说明：在指定时间执行命令

语法：at [-IV] [-d <作业编号>] [-f <文件>][-q <队列>][日期/时间]

atd

功能说明：系统执行定时任务的 DAEMON

语法：atd

atrm

功能说明：删除待执行的作业

语法：atrm[-V] [作业编号]

atq

功能说明：显示待执行的作业

语法:atq [-V][-q<队列>]

basename

功能说明：从全路径中提取文件名

语法：basename[<全路径>]

batch

功能说明：在系统负载许可时立即执行批处理指令

语法：batch [-V][-f<文件>][-q<队列>]

bunzip2

功能说明：bz2 文件的解压缩程序

语法：bunzip2 [-fkvsVL] [文件名...]

bzcat

功能说明：浏览.bz2 文件中包含的内容

语法：bzcat [-s] [文件名]

bzgrep

功能说明：搜索.bz2 文件

语法：bzgrep [文件名]

bzip2

功能说明：.bz2 文件的压缩程序

语法：bzip2 [-cdfkqstvzVL123456789] [文件名...]

bzip2recover

功能说明：修复损坏的.bz2 文件

语法：bzip2recover <文件名>

cal

功能说明：显示日历

语法：cal [-my][<月份> 年份]

cat

功能说明：合并多个文件，并将它们的内容输出到标准设备

语法：cat [-AbeEnstTuv] [--help] [--version] [文件……]

chfn

功能说明：改变 finger 指令显示的信息

语法：chfn [-f<真实姓名>][-h<家中电话>][-o<办公地址>][-p<办公电话>][-l<帐号名称>]

chgrp

功能说明：变更文件或目录的所属组

语法：chgrp [-cfhRv] [--help] [--version] [所属组] [文件或目录...] chgrp [-cfhRv] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]

chmod

功能说明：变更文件或目录的权限

语法：chmod [-cfhRv] [--help] [--version] [<权限范围>+/-/=<权限设置>,...] [文件或目录...] chmod [-cfhRv] [--help] [--version] [数字代号] [文件或目录...] chmod [-cfhRv] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]

chown

功能说明：变更文件或目录的所有者或所属组

语法：chown [-cfhRv] [--defeference] [--help] [--version] [所有者.<所属组>] [文件或目录...] chown [-cfhRv] [--defeference] [--help] [--version] [.所属组] [文件或目录...]

chown [-cfhRv] [--defeference] [--help] [--reference=<参考文件或目录>] [--version] [文件或目录...]

cksum

功能说明：检查文件的 CRC 是否正确

语法：cksum [--help] [--version] [文件]

col

功能说明：过滤控制字符

语法：col[-bfx]

colcrt

功能说明：过滤控制字符显示至标准输出

语法：colcrt[-bfx]

colrm

功能说明：滤掉指定的列

语法：colrm[开始列数编号 <结束列数编号>]

comm

功能说明：比较两个已排过序的文件

语法：comm[-123][--help][--version][file1][file2]

cp

功能说明：复制文件或目录

语 法：cp [-abdfilpPrRsuvx][-S< 备 份 字 尾 字 符 串 >][-V< 备 份 方 式 >][--help][--sparse=< 使用 时 机>][--version][原文件或目录][目的文件或目录]

crontab

功能说明：设置定时器维护系统中各用户的 crontab

语法：crontab [-u<用户名称>][设置文件]

csplit

功能说明：分割文件

语法：csplit [-kqsz][-b<输出格式>][--version][文件][模板样式]

cut

功能说明：指定欲显示的文件内容范围，并将它们输出到标准输出设备

语法：cut [-b<输出范围>][-c<输出范围>][-n][-d<分界字符>][-f<输出范围>][-s][--help]

dd

功能说明：读取、转换并输出

语法：dd [bs=<字节数>] [cbs=<字节数>] [conv=<关键词>] [count=<块数>][ibs=<字节数>]
[if=<文件>] [obs=<字节数>] [of=<文件>] [seek=<块数>] [skip=<块数>][--help] [--version]

ddate

功能说明：转换时间格式

语法：ddate [+格式] [时间]

df

功能说明：显示磁盘的文件系统与使用情况

语法：df [-ahHiklmPT] [-t<文件系统类型>] [-x<文件系统类型>] [--block-size=<块大小>]

[--help] [--no-sync] [--sunc] [--version] [文件或设备]

dir

功能说明：列出目录内容

语法：dir[目录]

dircolors

功能说明：设置 ls 指令在显示目录或文件时所用的色彩

语法：dircolors[色彩设置文件]或 dircolors[-bcp][--help][--version]

du

功能说明：显示目录或文件的大小

语法：du [-abcDhHklmsSx][-L<符号链接>][-x<文件>][--block-size=<块大小>][--exclude=<目录或文件>][--help][--max-depth=<目录层数>][--version][目录或文件]

echo

功能说明：显示文件回显一行正文

语法：echo [-n][<字符串>]

egrep

功能说明：在文件中查找指定的字符串

语法：egrep[模板样式][文件或目录]

expand

功能说明：将 tab 转换为空格

语法：expand [-i] [-t] [文件]

false

功能说明：返回 FALSE

语法：false [<忽略的命令指令>]

find

功能说明：查找文件路径

语法：Find [路径][表达式]

常用参数：-daystart, -follow, -maxdepth levels, -mindepth levels, -mount, -noleaf, -version, -amin n, -anewer file, -cmin n, -cnewer file, -empty, -false, -fstype type, -gid n, -iname pattern, -inum n, name, -mmin n, -perm +mode, -true, -type c, -uid n, -used, -xtypec, -fls file, -fprint file, -fprint0 file, -fprintf file format, -print0, -printf format, -ls

fmt

功能说明：编排文本文件

语法：fmt [-cstu] [-p <行起始字符>] [-w <每行字符数>] [--help] [--version] [文件]

fold

功能说明：限制文件行宽

语法：fold [-bs][-w<每行行数>][--help][--version][文件]

free

功能说明：检查内存状况

语法：free [-bkmot][-s <间隔秒数>]

ftp

功能说明：文件传输协议网络文件传输程序

语法：ftp[-dpginv] [主机名称或 IP 地址]

grep

功能说明：查找文件里符合条件的字符串

语法：grep [-abcEFGhHiILnqrsVwxy][-A<显示行数>][-B<显示列数>][-C<显示列数>][-d<进行动作>][-e<模板>][-f<模板文件>][--help][模板][文件或目录...]

gunzip

功能说明：解压缩文件

语法：gunzip [-aclnqt] [文件或目录]

gzexe

功能说明：压缩可执行文件

语法：gzexe [-d] [执行文件...]

gzip

功能说明：压缩文件

语法：gzip [-lcdflnNr][-<压缩率>] [文件或目录]

head

功能说明：输出文件内容的最前面部分

语法：head [-qv] [-c <显示数目>] [-n <显示行数>] [--help] [--version] [文件]

hexdump

功能说明：以各种格式显示数据以参数指定格式显示文件

语法：hexdump - bcdovx[-e 格式模式] [-f 格式文件] [-n 长度] [<文件名>...]

info

功能说明：显示说明

语法：info [--directory=<目录>] [--dribble=<文件>] [--file=<文件>] [--help][--index-seach=<字符串>]

[--node=< 主题>] [--output=< 文件>] [--restore=< 文件>] [--show-options] [--subnodes] [--usage] [--version]
[info 文件 <主题>]

join

功能说明：将两个文件中指定字段内容相同的行连接起来

语法：join [-i][-a<1 或 2>][-e< 字符串>][-o< 格式>][-t< 字符>][-v<1 或 2>][-1< 栏位>][-2< 栏位>][--help][--version][文件 1][文件 2]

kill

功能说明：中止执行中的程序或进程

语法：kill [-s<信号名称或编号>][程序...]

kill[-l<信号编号>]

ln

功能说明：链接文件或目录

语法：ln [-bdfinsv][-S<词尾备份字符串>][-V<备份方式>][--help][--version][原文件或目录][目标文件或目录]

login

功能说明：登录系统

语法：login look

ls

功能说明：列出目录内容

语法：ls [-laAbBcCdDfFgGhHikILmnNopqQrRsStuUvX] [-I<模板样式>] [-T<跳格字数>] [-w 每行字符数>]
[--block-size=< 扇区大小>] [--color=< 使用时机>] [--format=< 列表格式>] [--full-time][--help]
[--indicator-style=< 标注样式>] [--quoting-style=< 引号样式>][--show-control-chars]
[--sort=<排序方式>] [--time=<时间戳>] [--version] [文件或目录...]

man

功能说明：在线查询命令

语法：man [-acdfFhkKtwW] [--path] [-m <系统>] [-p <预处理程序>] [-C <配置文件>] [-M <路径>] [-P <浏览方式>] [-S <区段清单>] [区段名称] ...

man2html

功能说明：以 html 方式显示帮助手册

语法：man2html [选项] [文件]

manpath

功能说明：显示帮助文件的搜索路径

语法：manpath

md5sum

功能说明：计算并检查 MD5 函数值

语法：md5sum [-b] [-c <文件>] [文件]

mkdir

功能说明：创建目录

语法：mkdir [-p] dirName

mktemp

功能说明：建立临时文件

语法：mktemp [-qu] [文件名参数]

more

功能说明：一页一页显示文件

语法：more [-dpcs][-<行数>][+<字符串>][+<行数>][文件]

mv

功能说明：移动或重命名现有的文件或目录

语法：mv [-bfiuv][<-S附加字尾>][<-V方法>][--help][--version][原文件或目录][目标文件或目录]

namei

功能说明：分析路径

语法：namei [<路径名>]

newgrp

功能说明：登录另一个组语法：newgrp [<组名>]

nice

功能说明：设置优先权

语法：nice [-n <优先等级>] [执行指令]

nl

功能说明：给文件加上行号

语法：nl [--help] [--version] [文件]

nologin

功能说明：阻止登录

语法：nologin

od

功能说明：输出文件内容

语法：od [-w <每行字符数>] [--traditional] [文件]

passwd

功能说明：设置或更改用户密码

语法：passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]

paste

功能说明：合并文件的行

语法：paste [--help] [--version] [文件]

pftp

功能说明：文件传输协议网络文件传输程序

语法：pftp[-dginv][主机名称或 IP 地址]

pgrep

功能说明：查找进程

语法：pgrep [-flnvx][-d<分隔符>][-P<父进程 id 号>][-g<组号>][-u<effective id>][-U<real usr id>][-G<real groupid>][-t<终端号>][模式]

pkill

功能说明：向指定进程发信号

语法：pkill[-<信号>] [-flnvx][-d<分隔符>][-P<父进程 id 号>][-g<组号>][-u<effective id>][-U<real usr id>][-G<real groupid>][-t<终端号>][模式]

pr

功能说明：将文件格式化编排，以便打印

语法：语法：pr [-PcJ] [-N <列数编号>] [-s <间隔字符串>] [-T] [-v] [-w <每行字符数>][--help] [--version] [文件]

ps

功能说明：报告程序状况显示系统进程状态

语法：ps [acehvx] [o<显示列>] [-al] [-g<组名称>] [-p <程序识别码>] [p <程序识别码>] [-t<终端编号>] [t<终端编号>] [-u<用户识别码>]

ptx

功能说明：产生文件内容的交换索引

语法：ptx [-AbCGfFgiMoOrRS{TwW}]... [输入]...

pwd

功能说明：显示工作目录

语法：pwd

rename

功能说明：重命名文件

语法：rename [<原名>] [<目标名>] [<文件名>...]

renice

功能说明：调整优先权

语法：renice [优先等级] [-p <程序识别码>][-u <用户名>]

rm

功能说明：删除文件或目录

语法：rm [-dfrv] [--help] [--version] [文件或目录]

rmdir

功能说明：删除目录

语法：rmdir [-p] [--help] [--ignore-fail-on-empty] [--verbose] [--version] [目录]

shasum

功能说明：计算并检查 SHA1 信息分类

语法：shasum [-bct] [--help] [--version] [文件]

skill

功能说明：向指定进程发信号

语法：skill [<信号>] [<进程选择条件>]

setsid

功能说明：在窗口新建一个程序执行，在一个新的对话中运行进程

语法：setsid [<执行命令程序>]

setterm

功能说明：设置终端显示

语法：setterm [<显示选项>]

sort

功能说明：将文本文件内容加以排序

语法：sort [-gMs] [-T <目录>] [--version] [--help] [文件]

split

功能说明：分割文件

语法：split [--bytes=字节] [-C 字节数] [--lines] [--verbose] [--version] [要分割的文件] [输出文件名]

stty

功能说明：改变和显示终端行设置

语法：stty [-F DEVICE] [SETTING]

stty [-F DEVICE] [-a | --all]

stty [-F DEVICE] [-g | --save]

su

功能说明：变更用户身份

语法：su [-mpl] [-c <指令>] [用户帐号]

sum

功能说明：计算文件的校验和块数

语法: sum [-s][--version][文件]

sync

功能说明: 将内存缓冲区内的数据写入磁盘

语法: sync[--help][--version]

tac

功能说明: 合并多个文件, 并将它们的内容反序输出到标准输出设备

语法: tac[-br][-s<分隔符>][--help][--version][文件]

tail

功能说明: 输出文件内容的最后部分

语法: tail [-fqv][--retry][--bytes=显示数目][--lines=显示行数][+NUMBER][文件]

tar

功能说明: 打包文件

语法: tar [-cvhxz] [-f <打包文件>]

tload

功能说明: 显示负载状况

语法: tload [-V] [-d <间隔秒数>] [-s<刻度大小>] [终端编号]

top

功能说明: 显示, 管理执行中的程序

语法: top

touch

功能说明: 改变文件或目录时间

语法: touch [-acfm] [-d<日期时间>] [-r<参考文件或目录>] [-t<日期时间>][--help][--version] 文件或目录...

tr

功能说明: 转译字符

语法: tr [--complement] [--delete] [--squeeze-repeats] [-t] [--version] [第 1 字符集][第 2 字符集]

true

功能说明: 返回 TRUE

语法: true[<忽略的命令指令>]

tsort

功能说明: 完成拓扑排序

语法: tsort [--help][--version] [文件]

unexpand

功能说明: 将空格转换为 Tab

语法：unexpand [--tabs]

uniq uptime

功能说明：告诉系统开机后运行时间和负载情况

语法：uptime

vdir

功能说明：列出目录内容

语法：vdir [-laAbBcCdDfGgHhIiLlmnNopqQrRsStuUvxX][-I<模板样式>][-T<跳格字数>][-w<每行字符数>][--block-size=<扇区大小>][--color=<使用时机>][--format=<列表格式>][--indicator=<标注格式>][--show-control-chars][--sort=<排序方式>][--time=<时间戳>][--version][文件或目录]

vigr

功能说明：编辑 group 文件

语法：vigr

vipw

功能说明：编辑 passwd 文件

语法：vipw

watch

功能说明：周期性的执行一个程序

语法：watch [-dhv] [-n<时间间隔>] <命令>

wc

功能说明：打印文件的字节数，字数，行数

语法：wc [--bytes] [--lines] [-L] [--words] [--version] [文件]

whatis

功能说明：查询命令帮助

语法：whatis <关键字>

whereis

功能说明：查找文件

语法：whereis [-bfmsu] [-B <目录>...] [-M[目录]...] [文件...]

which

功能说明：显示命令文件的全路径

语法：which [--all][--read-alias][--skip-alias][--tty-only][--help][--version][命令]

write

功能说明：传送消息

语法：write [用户名称][终端编号]

xargs

功能说明：通过输入执行命令行

语法：xargs[**--max-lines**=<参数行数>][**--max-args**<参数个数>][**--interactive**][**--max-chars**]

zcat

功能说明：合并多个压缩文件，并将它们的内容输出到标准输出设备

语法：zcat [文件名]

zcmp

功能说明：比较压缩文件

语法：zcmp [文件名 1] [文件名 2]

zdiff

功能说明：比较压缩文件

语法：zdiff [文件名 1] [文件名 2]

zegrep

功能说明：在压缩文件中查找

语法：zegrep [模板样式] [文件或目录]

zfgrep

功能说明：在压缩文件中查找

语法：zfgrep [模板样式] [文件或目录]

zgrep

功能说明：在压缩文件中查找

语法：zgrep [模板样式] [文件或目录]

zless

功能说明：显示压缩文件内容

语法：zless [压缩文件]

zmore

功能说明：显示压缩文件内容

语法：zmore [压缩文件]

3 第三方软件约束要求

3.1 第三方应用移植约束要求

3.1.1 数据结构

银河麒麟 64 位操作系统环境下基础数据类型的大小(sizeof 函数计算的大小),如表所示:

表 3.1 银河麒麟 64 位操作系统基础数据类型

数据类型	大小
int	4
unsigned int	4
short	2
short int	2
unsigned short	2
long	8
unsigned long	8
long long	8
float	4
double	8
long double	16
char	1
char []	1
void *	8

基础数据类型的首地址对齐要求,也就是说默认情况下系统要求访问它时首地址必须是 2 字节、4 字节、8 字节或 16 字节对齐。

银河麒麟 64 位操作系统环境下基础数据类型的首地址对齐要求,如表所示:

表 3.2 64 位操作系统基础数据类型首地址对齐要求

数据类型	首地址对齐要求
int	4
unsigned int	4
short	2
short int	2
unsigned short	2
long	8
unsigned long	8
long long	8
float	4
double	8
long double	16
char	1
char []	1
void *	8

3.1.2 大小端

a)大端模式

大端模式是指数据的高位保存在内存的低地址中,而数据的低位保存在内存的高地址中。数据 0x12345678 使用大端模式在内存的存放形式如表:

表 3.3 大端模式数据存放形式

内存地址(0x80000100)	+0	+1	+2	+3	+4	+5
数据	0x12	0x34	0x56	0x78		

b)小端模式

小端模式是指数据的高位保存在内存的高地址中,而数据的低位保存在内存的低地址中。数据 0x12345678 使用小端模式在内存的存放形式如表:

表 3.4 小端模式数据存放形式

内存地址(0x80000100)	+0	+1	+2	+3	+4	+5
数据	0x78	0x56	0x34	0x12		

国产处理器平台的数据对齐方式存在差异,大端对齐和小端对齐的方式均存在,所以在移植第三方应用软件时应用需要针对实际情况采用不同的函数转换方式将变量进行大小端转换。

4 错误码定义

错误码的定义参见下表。

表 4.1 错误码列表

错误码	描述	备注
E2BIG	参数列表过长; 或者输出缓冲空间不够用; 或者参数超过了系统定义的最大值	新进程使用的参数列表或环境变量列表的字节总数超过系统限制(ARG_MAX)限制的字节。
EACCESS	权限拒绝	试图以禁止的文件访问权限访问文件。
EADDRINUSE	地址正在使用	指定的地址正在使用中。
EADDRNOTAVAIL	地址无效	对于本地系统而言,指定的地址无效。
EAFNOSUPPORT	不支持的地址族 (Address Family)	实现不支持指定的地址 。或对指定的 socket 地址而言,指定的地址无效。
EAGAIN	资源暂时无效	这是一个临时条件,稍后的调用可能正常完成。
EALREADY	连接在处理中	指定 socket 所需的连接在处理中。
EBADFD	文件描述符错误	文件描述符参数超出了范围,指向了未打开的文件,或者对以只写 (只读) 方式打开的文件进行读 (写) 操作。
EBADMSG	消息错误	在调用 read() 、getmsg() 、getpmsg() 或以 L_RECVFD 调用 ioctl() 请求一个流 (STREAM) 设备过程中,对接收消息的函数而言,到达流头部的消息是错误的 (inappropriate) : read() – 正在等待读取的流消息不是数据消息 (data message) ; getmsg()或 getpmsg() – 收到的是文件描述符而不是控制消息 (control message) ; ioctl() – 当指定参数 L_RECVFD 时,收到的是数据消息或控制消息,而不是文件描述符。或实现检测到错误 (corrupted) 消息。
EBUSY	资源忙	试图使用系统内当前无效的系统资源,即由于资

ECANCELED	操作取消	源被其他进程使用，且使用方式在一定程度上与本进程的请求冲突。
ECHILD	无孩子（child）进程	相关的异步操作在完成之前被取消。
ECONNABORTED	连接中断	wait() 或 waitpid() 被不存在的或未等待子进程的进程调用，ReWorks 系统中暂不支持。
ECONNREFUSED	连接拒绝	连接被中断。
ECONNRESET	连接复位	由于没有进程（listen），或连接队列已满且所用协议不支持重发，因而导致连接 socket 被拒绝。
EDEADLK	避免发生资源死锁	连接被对方强制关闭。
EDESTADDRREQ	需要目的地址	试图锁定可能导致死锁的系统资源。
EDOM	定义域错误	未绑定（bind）地址。
EDQUOT	超过磁盘配额	输入参数超过了数学函数（ISO C 标准库中定义）定义的域限制。
EEXIST	文件已存在	保留。
EFAULT	地址错误	已存在的文件在一个不合的上下文中引用。例如作为 link()函数的新 link 名称。
EFBIG	文件太大	在试图使用调用的参数时，系统检测到无效的地址。不能保证该错误一定可“地”被检测到，如果没有检测到将导致向进程发送一个表示地址越界（violation）的信号。
EHOSTUNREACH	无法达到主机	文件大小超过了实现的文件最大值或文件描述符中定义的文件偏移最大值。
EIDRM	标识符已被移除	目的主机无法达到（可能由于远程主机已关闭，或远程路由无法达到主机）。
EILSEQ	非法的字节序列	如果标识符已从系统中移除，则 XSI 进程间通信将返回该错误。
EINPROGRESS	操作进行中	检测到宽字符编码不能对应到有效的字符，或字节序列不能构成有效的宽字符编码（ISO C 标准中定义）。
EINTR	函数调用被中断	该编码用于表示异步操作未完成。或对 socket 文件描述符设置 O_NONBLOCK，但没有立即建立连接。
EINVAL	参数无效	在执行可中断函数过程中，进程捕获到异步信号。如果信号处理程序正常返回，则被中断的函数将返回被中断的点继续执行。
EIO	输入输出错误	调用接口时提供了无效的参数，例如在 single()或 kill()函数中指定了未定义的信号。
EISCONN	socket 已连接	发生了一些物理的输入输出错误。该错误可能在同一个文件描述符的后续操作中上报。在同一个文件描述符上的任何导致其他错误的操作将导致 EIO 错误 K 失。
EISDIR	这是一个目录	指定的 socket 已连接。
ELOOP	符号链接循环	试图以写模式打开一个目录。
		在路径解析的符号链接过程中发生循环。如果在

EMFILE	打开了太多文件	路径解析过程中，符号链接超过 SYMLOOP_MAX 也导致该错误。
EMLINK	太多连接	试图打开超过文件描述符最大值限制的文件。
EMSGSIZE	消息长度过长	单个文件的连接（link count）超过了 LINK_MAX。
EMULTIHOP	无	传输者发送的消息长度超过了内部消息缓冲或其他一些网络的限制。或消息缓冲的长度设置错误。保留。
ENAMETOOLONG	文件名（filename）太长	路径名长度超过了 PATH_MAX 或路径组件超过了 NAME_MAX。
ENETDOWN	网络关闭	用于到达目的端的本地网络接口关闭。
ENETRESET	网络终止了连接	无。
ENETUNREACH	网络不可到达	没有提供到达网络的路径。
ENFILE	系统中打开了太多文件	当前在系统中打开了太多文件。系统已经达到同时打开文件的预定义限制，临时不能接受打开另一个文件的请求。
ENOBUFS	没有有效的缓冲空间	系统中缺乏缓冲资源用以执行 socket 操作。
ENODATA	没有有效的数据	STREAM 头读队列中没有有效的数据。
ENODEV	无此设备	试图对设备调用不合适的函数，例如试图读一个只写设备（打印机）。
ENOENT	无此文件或目录	指定的路径组件不存在，或路径为空字符串。
ENOEXEC	可执行文件格式错误	请求的文件格式不是可执行文件实现所需的格式，尽管权限许可。
ENOLCK	没有有效的锁	同时存在的文件和记录锁已到达系统限制，并且当前没有有效的。
ENOLINK	无	预留。
ENOMEM	没有足够的空间	进程需要比硬件或系统内存管理约束所允许的更多的内存。
ENOMSG	没有期望类型的消息	在 XSI 进程间通信过程中，消息队列不包含所期望类型的消息。
ENOPROTOPT	协议无效	setsockopt()调用中指定的协议选项不支持。
ENOSPC	设备上无剩余空间	在对常规文件执行 write() 操作或对目录执行扩展操作过程中，设备上没有空 空间。
ENOSR	没有 STREAM 资源	执行 STREAM 相关函数的 STREAM 内存资源不够。这是一个临时条件，它可其他进程释放的资源中回收。
ENOSTR	不是流(STREAM)	在一个与流设备无关的文件描述符上执行流函数。
ENOSYS	函数未实现	试图使用当前实现中无效的函数。
ENOTCONN	socket 未连接	
ENOTDIR	不是目录	当需要一个目录时，指定路径的组件存在，但它并不是目录。
ENOTEMPTY	目录不为空	当期望一个空目录时，提供的目录非空。
ENOTSOCK	不是一个 socket	文件描述符引用的不是一个 socket。

ENOTSUP	操作不支持	当前实现不支持实时可选项的特征。
ENOTTY	不合的 I/O 控制操作	试图对文件或操作不用的特定文件执行控制函数。
ENXIO	无此设备或地址	特定文件的 I/O 操作指向了不存在的设备, 或超过了设备能力范围。
EOPNOTSUPP	socket 不支持的操作	socket 类型(地址族或协议)不支持所请求的操作。
E_OVERFLOW	溢出	试图执行一个操作, 而操作所生成的值超出了相关数据类型定义的范围的值。
EPERM	操作被禁止	试图执行受限于合权限的进程或文件/ 其他资源限制的操作。
EPIPE	坏 (broken) 管道	对没有进程读数据的 socket 、管道、FIFO 执行写操作。
EPROTO	协议错误	该错误与设备相关, 但通常不是硬件错误。
EPROTONOSUPPORT	协议不支持	地址族不支持指定的协议, 或当前实现不支持指定的协议。
EPROTOTYPE	socket 不支持的协议类型	协议不支持指定的 socket 类型。
ERANGE	结果太大或太小	对有效的空间而言, 函数的结果太大 (溢出, overflow) 或太小 (下溢, underflow)。(ISOC 标准定义)
EROFS	只读文件系统	试图在只读文件系统上修改文件或目录。
ESPIPE	定位无效	试图访问与管道或 FIFO 关联的文件偏移。
ESRCH	无此进程	没有指定的进程 ID 对应的进程。
ESTALE	已过时的文件句柄	预留。
ETIME	ioctl()流操作超时	为 ioctl()流操作设置的定时器到期。导致该错误的原因与具体设备相关, 可能与硬件或软件失败相关,
	连接超时或操作超时	也可能特定的操作的超时时间太小。ioctl()操作的状态未定义。
ETIMEDOUT		与远程机器的连接超时。如果在执行上报该错误的函数过程中发生超时 (相对于函数调用前超时), 是否正确完成该函数相关的部分或全部行为是未定义的。
ETXTBSY	文本文件	或在操作完成前超过了时间限制。
		试图执行当前用于写的打开操作的纯过程程序, 或者对一个正在执行的纯过程程序文件执行 write() 模式的打开操作。
EWouldBlock	操作可能阻塞	在标记为非阻塞 (non-blocking) 的 socket 上的操作遇到因数据无效而使函数挂起执行的情况。
EXDEV	不恰当的连接	试图对另一个文件系统上的文件执行 link 操作。

5 sp3 2303 和 sp2 的 API 接口差异

增加 pthread.h

```
extern int pthread_cond_clockwait (pthread_cond_t *__restrict __cond,  
                                   pthread_mutex_t *__restrict __mutex,  
                                   __clockid_t __clock_id,  
                                   const struct timespec *__restrict __abstime)
```

减少 string.h

```
extern void *mesa_memcpy void *__restrict __dest, const void *__restrict __src, size_t __n)  
extern void *mesa_memmove (void *__dest, const void *__src, size_t __n)  
extern void *mesa_memset (void *__s, int __c, size_t __n)
```

kernel-devel

【 变动超过一万行，省略 】

postgresql-devel

减少：

```
extern bool pq_buffer_has_data(void);
```

rpm-devel

增加 rpmdb.h

```
rpmdbIndexIterator rpmdbIndexKeyIteratorInit(rpmdb db, rpmDbiTag rpmtag);
```

ndctl-devel

增加 libdaxctl.h

```
unsigned long long daxctl_dev_get_resource(struct daxctl_dev *dev);  
struct daxctl_ctx *daxctl_dev_get_ctx(struct daxctl_dev *dev);  
int daxctl_dev_is_enabled(struct daxctl_dev *dev);  
int daxctl_dev_disable(struct daxctl_dev *dev);  
int daxctl_dev_enable_devdax(struct daxctl_dev *dev);  
int daxctl_dev_enable_ram(struct daxctl_dev *dev);  
int daxctl_dev_get_target_node(struct daxctl_dev *dev);  
struct daxctl_memory;  
struct daxctl_memory *daxctl_dev_get_memory(struct daxctl_dev *dev);  
struct daxctl_dev *daxctl_memory_get_dev(struct daxctl_memory *mem);  
const char *daxctl_memory_get_node_path(struct daxctl_memory *mem);  
unsigned long daxctl_memory_get_block_size(struct daxctl_memory *mem);  
int daxctl_memory_online(struct daxctl_memory *mem);  
int daxctl_memory_offline(struct daxctl_memory *mem);  
int daxctl_memory_is_online(struct daxctl_memory *mem);  
int daxctl_memory_num_sections(struct daxctl_memory *mem);  
int daxctl_memory_is_movable(struct daxctl_memory *mem);  
int daxctl_memory_online_no_movable(struct daxctl_memory *mem);
```

增加 libndctl.h

```
int ndctl_bus_poll_scrub_completion(struct ndctl_bus *bus,  
                                   unsigned int poll_interval, unsigned int timeout);  
int ndctl_dimm_zero_label_extent(struct ndctl_dimm *dimm,
```



```

        unsigned int len, unsigned int offset);
struct ndctl_cmd *ndctl_dimm_read_label_index(struct ndctl_dimm *dimm);
struct ndctl_cmd *ndctl_dimm_read_label_extent(struct ndctl_dimm *dimm,
        unsigned int len, unsigned int offset);
int ndctl_cmd_cfg_read_set_extent(struct ndctl_cmd *cfg_read,
        unsigned int len, unsigned int offset);
int ndctl_cmd_cfg_write_set_extent(struct ndctl_cmd *cfg_write,
        unsigned int len, unsigned int offset);
unsigned long ndctl_region_get_align(struct ndctl_region *region);
int ndctl_region_set_align(struct ndctl_region *region, unsigned long align);
int ndctl_region_get_target_node(struct ndctl_region *region);
        NDCTL_NS_MODE_SECTOR = NDCTL_NS_MODE_SAFE,
int ndctl_namespace_is_configuration_idle(struct ndctl_namespace *ndns);
int ndctl_namespace_get_target_node(struct ndctl_namespace *ndns);
int ndctl_pfn_get_num_alignments(struct ndctl_pfn *pfn);
unsigned long ndctl_pfn_get_supported_alignment(struct ndctl_pfn *pfn, int i);
int ndctl_dax_get_num_alignments(struct ndctl_dax *dax);
unsigned long ndctl_dax_get_supported_alignment(struct ndctl_dax *dax, int i);
int ndctl_cmd_xlat_firmware_status(struct ndctl_cmd *cmd);
int ndctl_cmd_submit_xlat(struct ndctl_cmd *cmd);
enum ndctl_security_state {
        NDCTL_SECURITY_INVALID = -1,
        NDCTL_SECURITY_DISABLED = 0,
        NDCTL_SECURITY_UNLOCKED,
        NDCTL_SECURITY_LOCKED,
        NDCTL_SECURITY_FROZEN,
        NDCTL_SECURITY_OVERWRITE,
};
enum ndctl_security_state ndctl_dimm_get_security(struct ndctl_dimm *dimm);
bool ndctl_dimm_security_is_frozen(struct ndctl_dimm *dimm);
int ndctl_dimm_update_passphrase(struct ndctl_dimm *dimm,
        long ckey, long nkey);
int ndctl_dimm_disable_passphrase(struct ndctl_dimm *dimm, long key);
int ndctl_dimm_freeze_security(struct ndctl_dimm *dimm);
int ndctl_dimm_secure_erase(struct ndctl_dimm *dimm, long key);
int ndctl_dimm_overwrite(struct ndctl_dimm *dimm, long key);
int ndctl_dimm_wait_overwrite(struct ndctl_dimm *dimm);
int ndctl_dimm_update_master_passphrase(struct ndctl_dimm *dimm,
        long ckey, long nkey);
int ndctl_dimm_master_secure_erase(struct ndctl_dimm *dimm, long key);

```

tpm2-tss-devel

改变 tss2_esys.h

TPMI_RH_HIERARCHY 变为 ESYS_TR

增加 tss2_fapi.h

```
TSS2_RC Fapi_GetEsysBlob(
    FAPI_CONTEXT    *context,
    char            const *path,
    uint8_t         *type,
    uint8_t         **data,
    size_t          *length);
TSS2_RC Fapi_GetEsysBlob_Async(
    FAPI_CONTEXT    *context,
    char            const *path);
TSS2_RC Fapi_GetEsysBlob_Finish(
    FAPI_CONTEXT    *context,
    uint8_t         *type,
    uint8_t         **data,
    size_t          *length);
```

改变 tss2_sys.h

```
TSS2_RC Tss2_Sys_ReadClock(      TSS2_RC Tss2_Sys_ReadClock(
    TSS2_SYS_CONTEXT *sysContext,  TSS2_SYS_CONTEXT *sysContext,
    TPMS_TIME_INFO *currentTime); |    TSS2L_SYS_AUTH_COMMAND const *cmdAuthsArray,
                                   >    TPMS_TIME_INFO *currentTime,
                                   >    TSS2L_SYS_AUTH_RESPONSE *rspAuthsArray);
```

增加 tss2_tcti_cmd.h

增加 tss2_tcti_swtpm.h

gdk-pixbuf2-devel

增加 gdk-pixbuf-io.h

```
gboolean gdk_pixbuf_init_modules const char *path,GError **error);
```

libreport-devel

减少 internal_libreport_gtk.h、libreport_curl.h

evolution-data-server-devel

减少 camel-stream-buffer.h

```
void camel_stream_buffer_discard_cache(CamelStreamBuffer *sbf);
```

openssl-devel

增加 ec.h

```
void *EC_KEY_get_key_method_data(EC_KEY *key, void *(*dup_func) (void *), void (*free_func) (void *), void
(*clear_free_func) (void *));
```

```
void *EC_KEY_insert_key_method_data(EC_KEY *key, void *data, void *(*dup_func) (void *), void (*free_func)
(void *), void (*clear_free_func) (void *));
```

增加 evp.h

```
EVP_PKEY_CTX *EVP_PKEY_CTX_new_pkey_id(EVP_PKEY *pkey, int id, ENGINE *e);
```

增加 sm2.h

```

int SM2_compute_key(void *out, size_t outlen, int initiator,
    const char *peer_uid, int peer_uid_len, const char *self_uid, int self_uid_len,
    const EC_KEY *peer_ecdhe_key, const EC_KEY *self_ecdhe_key, const EC_KEY *peer_pub_key, const
EC_KEY *self_eckey,
    const EVP_MD *md);
int SM2_KAP_CTX_init(SM2_KAP_CTX *ctx,
    EC_KEY *ec_key, const char *id, size_t idlen,
    EC_KEY *remote_pubkey, const char *rid, size_t ridlen,
    int is_initiator, int do_checksum);
int SM2_KAP_prepare(SM2_KAP_CTX *ctx, unsigned char *ephem_point,
    size_t *ephem_point_len);
int SM2_KAP_compute_key(SM2_KAP_CTX *ctx, const unsigned char *remote_ephem_point,
    size_t remote_ephem_point_len, unsigned char *key, size_t keylen,
    unsigned char *checksum, size_t *checksumlen);
int SM2_KAP_final_check(SM2_KAP_CTX *ctx, const unsigned char *checksum,
    size_t checksumlen);
void SM2_KAP_CTX_cleanup(SM2_KAP_CTX *ctx);

```

增加 sm4.h

```

void sm4_cbc_encrypt(const unsigned char *in, unsigned char *out, size_t len,
    const SM4_KEY *key, unsigned char *ivec, const int enc);
void sm4_cfb128_encrypt(const unsigned char *in, unsigned char *out, size_t length,
    const SM4_KEY *key, unsigned char *ivec, int *num,
const int enc);
void sm4_ecb_encrypt(const unsigned char *in, unsigned char *out,
    const SM4_KEY *key, const int enc);
void sm4_ofb128_encrypt(const unsigned char *in, unsigned char *out, size_t length,
    const SM4_KEY *key, unsigned char *ivec, int *num);
void sm4_ctr128_encrypt(const unsigned char *in, unsigned char* out, size_t len,
    const SM4_KEY* key, unsigned char* ivec,
    unsigned char ecount_buf[SM4_BLOCK_SIZE], unsigned int *num);

```

gobject-introspection-devel

增加 girepository.h

```

void g_irepository_get_object_gtype_interfaces (GIRepository *repository,
    GType gtype,
    guint *n_interfaces_out,
    GIInterfaceInfo ***interfaces_out);

```

增加 gtypeinfo.h

```

GTypeTag g_type_info_get_storage_type (GTypeInfo *info);
void g_type_info_argument_from_hash_pointer (GTypeInfo *info,
    gpointer hash_pointer,
    GIArgument *arg);
gpointer g_type_info_hash_pointer_from_argument (GTypeInfo *info,
    GIArgument *arg);

```

增加 giversion.h

mariadb-devel

【 变动较大，省略 】

librsvg2-devel

增加 rsvg.h

```
gboolean rsvg_handle_set_stylesheet (RsvgHandle *handle,
                                     const guint8 *css,
                                     gsize css_len,
                                     GError **error);
```

libldb-devel

减少 ldb.h

```
int ldb_msg_element_add_value(TALLOC_CTX *mem_ctx,
                              struct ldb_message_element *el,
                              const struct ldb_val *val);

int ldb_msg_add_string_flags(struct ldb_message *msg,
                             const char *attr_name, const char *str,
                             int flags);

int ldb_msg_append_value(struct ldb_message *msg,
                         const char *attr_name,
                         const struct ldb_val *val,
                         int flags);

int ldb_msg_append_steal_value(struct ldb_message *msg,
                              const char *attr_name,
                              struct ldb_val *val,
                              int flags);

int ldb_msg_append_steal_string(struct ldb_message *msg,
                                const char *attr_name, char *str,
                                int flags);

int ldb_msg_append_string(struct ldb_message *msg,
                          const char *attr_name, const char *str,
                          int flags);

int ldb_msg_append_linearized_dn(struct ldb_message *msg, const char *attr_name,
                                 struct ldb_dn *dn, int flags);

int ldb_msg_append_fmt(struct ldb_message *msg, int flags,
                      const char *attr_name, const char *fmt, ...) PRINTF_ATTRIBUTE(4,5);
```

glib2-devel

增加 garray.h

```
GLIB_AVAILABLE_IN_2_64
gpointer g_array_steal (GArray *array,
                       gpointer *array,
```

```

                                gsize          *len);

GLIB_AVAILABLE_IN_2_64
gpointer* g_ptr_array_steal (GPtrArray *array,
                              gsize      *len);

GLIB_AVAILABLE_IN_2_64
guint8* g_byte_array_steal (GByteArray *array,
                             gsize      *len);

改变 gbookmarkfile.h
88c88
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_set_application_info)
96c96,104
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_AVAILABLE_IN_2_66
> gboolean g_bookmark_file_set_application_info (GBookmarkFile *bookmark,
>
>                                const char *uri,
>                                const char *name,
>                                const char *exec,
>                                int count,
>                                GDateTime *stamp,
>                                GError **error);
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_get_application_info)
123c139
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_set_added_date_time)
127c143,147
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_AVAILABLE_IN_2_66
> void g_bookmark_file_set_added_date_time (GBookmarkFile *bookmark,
>
>                                const char *uri,
>                                GDateTime *added);
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_get_added_date_time)
131c151,155
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_AVAILABLE_IN_2_66
> GDateTime *g_bookmark_file_get_added_date_time (GBookmarkFile *bookmark,
>
>                                const char *uri,
>                                GError **error);

```

```

> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_set_modified_date_time)
135c159,163
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_AVAILABLE_IN_2_66
> void          g_bookmark_file_set_modified_date_time (GBookmarkFile  *bookmark,
>
>                  const char      *uri,
>                  GDateTime       *modified);
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_get_modified_date_time)
139c167,171
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_AVAILABLE_IN_2_66
> GDateTime     *g_bookmark_file_get_modified_date_time (GBookmarkFile  *bookmark,
>
>                  const char      *uri,
>                  GError          **error);
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_set_visited_date_time)
143c175,179
< GLIB_AVAILABLE_IN_ALL
---
> GLIB_AVAILABLE_IN_2_66
> void          g_bookmark_file_set_visited_date_time (GBookmarkFile  *bookmark,
>
>                  const char      *uri,
>                  GDateTime       *visited);
> GLIB_DEPRECATED_IN_2_66_FOR(g_bookmark_file_get_visited_date_time)

```

增加 gbookmarkfile.h

```

GLIB_AVAILABLE_IN_2_66
gboolean      g_bookmark_file_get_application_info (GBookmarkFile  *bookmark,
                                                    const char      *uri,
                                                    const char      *name,
                                                    char           **exec,
                                                    unsigned int    *count,
                                                    GDateTime     **stamp,
                                                    GError         **error);

GLIB_AVAILABLE_IN_2_66
GDateTime     *g_bookmark_file_get_visited_date_time (GBookmarkFile  *bookmark,
                                                    const char      *uri,
                                                    GError         **error);

```

增加 gdtlsconnection.h

```

G_GNUC_BEGIN_IGNORE_DEPRECATED
GLIB_AVAILABLE_IN_2_66

```

```

gboolean          g_dtls_connection_get_channel_binding_data (GDTLSConnection *conn,
                                                             GTLSChannelBindingType
type,
                                                             GByteArray
*data,
                                                             GError
**error);
G_GNUC_END_IGNORE_DEPRECATIONS

```

减少 gfileutils.h

```

GLIB_AVAILABLE_IN_ALL
GQuark      g_file_error_quark      void);
GLIB_AVAILABLE_IN_ALL
GFileError g_file_error_from_errno (gint err_no);
GLIB_AVAILABLE_IN_ALL
gboolean g_file_test      (const gchar *filename,
                           GFileTest   test);

GLIB_AVAILABLE_IN_ALL
gboolean g_file_get_contents (const gchar *filename,
                             gchar       **contents,
                             gsize       *length,
                             GError      **error);

GLIB_AVAILABLE_IN_ALL
gboolean g_file_set_contents (const gchar *filename,
                             const gchar *contents,
                             gssize       length,
                             GError      **error);

GLIB_AVAILABLE_IN_ALL
gchar *g_file_read_link      (const gchar *filename,
                              GError      **error);

GLIB_AVAILABLE_IN_2_30
gchar *g_mkdtmp      (gchar *tmpl);
GLIB_AVAILABLE_IN_2_30
gchar *g_mkdtmp_full (gchar *tmpl,
                      gint    mode);

GLIB_AVAILABLE_IN_ALL
gint    g_mkstemp      (gchar *tmpl);
GLIB_AVAILABLE_IN_ALL
gint    g_mkstemp_full (gchar *tmpl,
                        gint    flags,
                        gint    mode);

GLIB_AVAILABLE_IN_ALL
gint    g_file_open_tmp (const gchar *tmpl,
                          gchar       **name_used,

```

```

                                GError      **error);

GLIB_AVAILABLE_IN_2_30
gchar      *g_dir_make_tmp      (const gchar *tmpl,
                                GError      **error);

GLIB_AVAILABLE_IN_ALL
gchar      *g_build_path        (const gchar *separator,
                                const gchar *first_element,
                                ...) G_GNUC_MALLOC G_GNUC_NULL_TERMINATED;

GLIB_AVAILABLE_IN_ALL
gchar      *g_build_pathv       (const gchar *separator,
                                gchar        **args) G_GNUC_MALLOC;

GLIB_AVAILABLE_IN_ALL
gchar      *g_build_filename    (const gchar *first_element,
                                ...) G_GNUC_MALLOC G_GNUC_NULL_TERMINATED;

GLIB_AVAILABLE_IN_ALL
gchar      *g_build_filenamev   (gchar        **args) G_GNUC_MALLOC;
GLIB_AVAILABLE_IN_2_56
gchar      *g_build_filename_valist (const gchar *first_element,
                                va_list      *args) G_GNUC_MALLOC;

GLIB_AVAILABLE_IN_ALL
gint        g_mkdir_with_parents (const gchar *pathname,
                                gint          mode);

GLIB_AVAILABLE_IN_ALL
gboolean     g_path_is_absolute (const gchar *file_name);
GLIB_AVAILABLE_IN_ALL
const gchar *g_path_skip_root   (const gchar *file_name);
GLIB_DEPRECATED_FOR(g_path_get_basename)
const gchar *g_basename         (const gchar *file_name);
GLIB_AVAILABLE_IN_ALL
gchar      *g_get_current_dir   (void);
GLIB_AVAILABLE_IN_ALL
gchar      *g_path_get_basename (const gchar *file_name) G_GNUC_MALLOC;
GLIB_AVAILABLE_IN_ALL
gchar      *g_path_get_dirname   (const gchar *file_name) G_GNUC_MALLOC;
GLIB_AVAILABLE_IN_2_58
gchar      *g_canonicalize_filename (const gchar *filename,
                                const gchar *relative_to) G_GNUC_MALLOC;

G_END_DECLS

```

增加 gioenumtypes.h

```

GLIB_AVAILABLE_IN_ALL GType g_tls_channel_binding_type_get_type (void) G_GNUC_CONST;
GLIB_AVAILABLE_IN_ALL GType g_tls_channel_binding_error_get_type (void) G_GNUC_CONST;
GLIB_AVAILABLE_IN_ALL GType g_memory_monitor_warning_level_get_type (void) G_GNUC_CONST;

```


增加 glib-autocleanups.h

```
G_GNUC_BEGIN_IGNORE_DEPRECATIONS
G_DEFINE_AUTO_PTR_CLEANUP_FUNC(GDate, g_date_free)
G_DEFINE_AUTO_PTR_CLEANUP_FUNC(GMainContextPusher, g_main_context_pusher_free)
G_DEFINE_AUTO_PTR_CLEANUP_FUNC(GUri, g_uri_unref)
G_GNUC_END_IGNORE_DEPRECATIONS
```

增加 glib-types.h

```
GLIB_AVAILABLE_IN_2_66
GType    g_uri_get_type          (void) G_GNUC_CONST;
```

增加 glib-unix.h

```
GLIB_AVAILABLE_IN_2_64
struct passwd *g_unix_get_passwd_entry (const gchar *user_name,
                                         GError **error);
```

增加 glist.h

```
GLIB_AVAILABLE_IN_2_64
void    g_clear_list              (GList **list_ptr,
                                   GDestroyNotify destroy);
```

增加 gliststore.h

```
GLIB_AVAILABLE_IN_2_64
gboolean    g_list_store_find      (GListStore *store,
                                     gpointer item,
                                     guint *position);

GLIB_AVAILABLE_IN_2_64
gboolean    g_list_store_find_with_equal_func (GListStore *store,
                                               gpointer item,
                                               GEqualFunc
equal_func,
                                               guint *position);
```

增加 gmain.h

```
G_GNUC_BEGIN_IGNORE_DEPRECATIONS
GLIB_AVAILABLE_STATIC_INLINE_IN_2_64
static inline GMainContextPusher *
g_main_context_pusher_new (GMainContext *main_context)
{
    g_main_context_push_thread_default (main_context);
    return (GMainContextPusher *) main_context;
}
G_GNUC_END_IGNORE_DEPRECATIONS
G_GNUC_BEGIN_IGNORE_DEPRECATIONS
```

```

GLIB_AVAILABLE_STATIC_INLINE_IN_2_64
static inline void
g_main_context_pusher_free (GMainContextPusher *pusher)
{
    g_main_context_pop_thread_default ((GMainContext *) pusher);
}
G_GNUC_END_IGNORE_DEPRECATIONS
G_GNUC_BEGIN_IGNORE_DEPRECATIONS
GLIB_AVAILABLE_IN_2_64
void      g_source_set_dispose_function (GSource      *source,
                                         GSourceDisposeFunc  dispose);
G_GNUC_END_IGNORE_DEPRECATIONS

```

改变 gparam.h

```

gboolean      g_param_value_defaults      (GParamSpec *pspec,
                                           GValue      *value);  ->      const GValue
*value);

```

增加 gparam.h

```

GLIB_AVAILABLE_IN_2_66
gboolean g_param_spec_is_valid_name      (const gchar *name);

```

增加 gsignal.h

```

GLIB_AVAILABLE_IN_2_66
gboolean      g_signal_is_valid_name (const gchar *name);

```

增加 gclist.h

```

GLIB_AVAILABLE_IN_2_64
void      g_clear_slist      (GSList      **slist_ptr,
                              GDestroyNotify  destroy);

```

增加 gstdio.h

```

GLIB_AVAILABLE_IN_2_64
gint g_fsync      (gint fd);

```

减少 gstrfuncs.h

```

gpointer      g_memdup2      (gconstpointer mem,
                              gsize          byte_size) G_GNUC_ALLOC_SIZE(2);

```

增加 gtask.h

```

GLIB_AVAILABLE_IN_2_64
void      g_task_return_value      (GTask      *task,
                                    GValue      *result);
GError    **error);

```

GLIB_AVAILABLE_IN_2_64

gboolean g_task_propagate_value (GTask *task,
GValue *value,

增加 gtlconnection.h

GLIB_AVAILABLE_IN_2_66

gboolean g_tls_connection_get_channel_binding_data (GtlsConnection *conn,
GtlsChannelBindingType
type,

GByteArray

*data,

GError

**error);

GLIB_AVAILABLE_IN_2_66

GQuark g_tls_channel_binding_error_quark (void);

增加 gunixmounts.h

GLIB_AVAILABLE_IN_2_66

GUnixMountPoint *g_unix_mount_point_at (const char *mount_path,
guint64 *time_read);

减少 gurifuncs.h

增加 guri.h

增加 gutils.h

GLIB_AVAILABLE_IN_2_64

gchar * g_get_os_info (const gchar *key_name);

增加 gvaluetypes.h

GLIB_AVAILABLE_IN_2_66

void g_value_set_interned_string (GValue *value,
> const gchar *v_string);

rdma-core-devel

增加 efadv.h

EFADV_DEVICE_ATTR_CAPS_RNR_RETRY = 1 << 1,

增加 mad.h

IB_PORT_EXT_NDR_FEC_MODE_SUPPORTED_F,

IB_PORT_EXT_NDR_FEC_MODE_ENABLED_F,

IB_PORT_EXT_NDR_FEC_MODE_LAST_F,

增加 mlx5dv.h

void *mlx5dv_dm_map_op_addr(struct ibv_dm *dm, uint8_t op);

int _mlx5dv_query_port(struct ibv_context *context,

```

        uint32_t port_num,
        struct mlx5dv_port *info,
        size_t info_len);

static inline int mlx5dv_query_port(struct ibv_context *context,
        uint32_t port_num,
        struct mlx5dv_port *info)

struct mlx5dv_devx_umem *
mlx5dv_devx_umem_reg_ex(struct ibv_context *ctx, struct mlx5dv_devx_umem_in *umem_in);
void mlx5dv_dr_domain_set_reclaim_device_memory(struct mlx5dv_dr_domain *dmn,
        bool enable);

struct mlx5dv_dr_action *
mlx5dv_dr_action_create_dest_devx_tir(struct mlx5dv_devx_obj *devx_obj);
struct mlx5dv_dr_action *
mlx5dv_dr_action_create_dest_array(struct mlx5dv_dr_domain *domain,
        size_t num_dest,
        struct mlx5dv_dr_action_dest_attr *dests[]);
struct mlx5dv_dr_action *mlx5dv_dr_action_create_default_miss(void);
struct mlx5dv_dr_action *
mlx5dv_dr_action_create_aso(struct mlx5dv_dr_domain *domain,
        struct mlx5dv_devx_obj *devx_obj,
        uint32_t offset,
        uint32_t flags,
        uint8_t return_reg_c);
int mlx5dv_dr_action_modify_aso(struct mlx5dv_dr_action *action,
        uint32_t offset,
        uint32_t flags,
        uint8_t return_reg_c);

struct mlx5dv_dr_action *
mlx5dv_dr_action_create_flow_sampler(struct mlx5dv_dr_flow_sampler_attr *attr);
struct mlx5dv_dr_action *
mlx5dv_dr_action_create_pop_vlan(void);
struct mlx5dv_dr_action *
mlx5dv_dr_action_create_push_vlan(struct mlx5dv_dr_domain *domain,
        __be32 vlan_hdr);
struct mlx5dv_pp *mlx5dv_pp_alloc(struct ibv_context *context,
        size_t pp_context_sz,
        const void *pp_context,
        uint32_t flags);
void mlx5dv_pp_free(struct mlx5dv_pp *pp);
int mlx5dv_query_qp_lag_port(struct ibv_qp *qp,
        uint8_t *port_num,
        uint8_t *active_port_num);
int mlx5dv_modify_qp_lag_port(struct ibv_qp *qp, uint8_t port_num);
int mlx5dv_modify_qp_udp_sport(struct ibv_qp *qp, uint16_t udp_sport);

```

```

struct mlx5dv_sched_node *
mlx5dv_sched_node_create(struct ibv_context *context,
                        const struct mlx5dv_sched_attr *sched_attr);
struct mlx5dv_sched_leaf *
mlx5dv_sched_leaf_create(struct ibv_context *context,
                        const struct mlx5dv_sched_attr *sched_attr);
int mlx5dv_sched_node_modify(struct mlx5dv_sched_node *node,
                        const struct mlx5dv_sched_attr *sched_attr);
int mlx5dv_sched_leaf_modify(struct mlx5dv_sched_leaf *leaf,
                        const struct mlx5dv_sched_attr *sched_attr);
int mlx5dv_sched_node_destroy(struct mlx5dv_sched_node *node);
int mlx5dv_sched_leaf_destroy(struct mlx5dv_sched_leaf *leaf);
int mlx5dv_modify_qp_sched_elem(struct ibv_qp *qp,
                        const struct mlx5dv_sched_leaf *requestor,
                        const struct mlx5dv_sched_leaf *responder);
int mlx5dv_reserved_qpn_alloc(struct ibv_context *ctx, uint32_t *qpn);
int mlx5dv_reserved_qpn_dealloc(struct ibv_context *ctx, uint32_t qpn);
增加 rdma_cma.h
int rdma_reject_ece(struct rdma_cm_id *id, const void *private_data,
                    uint8_t private_data_len);
int rdma_set_local_ece(struct rdma_cm_id *id, struct ibv_ece *ece);
int rdma_get_remote_ece(struct rdma_cm_id *id, struct ibv_ece *ece);
增加 umad.h
int umad_sort_ca_device_list(struct umad_device_node **head, size_t size);
改变 verbs.h
1139c1142,1143
<    void (*_compat_query_device)(void);
____
>    int (*_compat_query_device)(struct ibv_context *context,
>                                struct ibv_device_attr *device_attr);
增加 verbs.h
int ibv_get_device_index(struct ibv_device *device);
struct ibv_context *ibv_import_device(int cmd_fd);
struct ibv_pd *ibv_import_pd(struct ibv_context *context,
                            uint32_t pd_handle);
void ibv_unimport_pd(struct ibv_pd *pd);
struct ibv_mr *ibv_import_mr(struct ibv_pd *pd, uint32_t mr_handle);
void ibv_unimport_mr(struct ibv_mr *mr);
struct ibv_dm *ibv_import_dm(struct ibv_context *context, uint32_t dm_handle);
void ibv_unimport_dm(struct ibv_dm *dm);
int _ibv_query_gid_ex(struct ibv_context *context, uint32_t port_num,
                    uint32_t gid_index, struct ibv_gid_entry *entry,
                    uint32_t flags, size_t entry_size);
static inline int ibv_query_gid_ex(struct ibv_context *context,

```

```

        uint32_t port_num, uint32_t gid_index,
        struct ibv_gid_entry *entry, uint32_t flags)
ssize_t _ibv_query_gid_table(struct ibv_context *context,
        struct ibv_gid_entry *entries, size_t max_entries,
        uint32_t flags, size_t entry_size);
static inline ssize_t ibv_query_gid_table(struct ibv_context *context,
        struct ibv_gid_entry *entries,
        size_t max_entries, uint32_t flags)
struct ibv_mr *ibv_reg_dmabuf_mr(struct ibv_pd *pd, uint64_t offset, size_t length,
        uint64_t iova, int fd, int access);
enum ibv_fork_status ibv_is_fork_initialized(void);
static inline uint16_t ibv_flow_label_to_udp_sport(uint32_t fl)
int ibv_set_ece(struct ibv_qp *qp, struct ibv_ece *ece);
int ibv_query_ece(struct ibv_qp *qp, struct ibv_ece *ece);

```

avahi-devel

减少 glib-malloc.h、glib-watch.h

增加 qt-watch.h

libcap-ng-devel

增加 cap-ng.h

```

int capng_get_rootid(void);
int capng_set_rootid(int rootid);

```

libbpf-devel

改变 bpf.h

```

LIBBPF_API int bpf_load_btf(void *btf, __u32 btf_size, char *log_buf,
LIBBPF_API int bpf_load_btf(const void *btf, __u32 btf_size, char *log_buf,

```

改变 bpf_helper_defs.h

```

< static long (*bpf_sk_release)(struct bpf_sock *sock) = (void *) 86;
____
> static long (*bpf_sk_release)(void *sock) = (void *) 86;
128c135
< static long (*bpf_tcp_check_syncookie)(struct bpf_sock *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32
th_len) = (void *) 100;
____
> static long (*bpf_tcp_check_syncookie)(void *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len) =
(void *) 100;
135,136c142,143
< static void (*bpf_sk_storage_get)(void *map, struct bpf_sock *sk, void *value, __u64 flags) = (void *) 107;
< static long (*bpf_sk_storage_delete)(void *map, struct bpf_sock *sk) = (void *) 108;
____
> static void (*bpf_sk_storage_get)(void *map, void *sk, void *value, __u64 flags) = (void *) 107;
> static long (*bpf_sk_storage_delete)(void *map, void *sk) = (void *) 108;

```

138c145

```
< static __s64 (*bpf_tcp_gen_syncookie)(struct bpf_sock *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len) = (void *) 110;
```

```
> static __s64 (*bpf_tcp_gen_syncookie)(void *sk, void *iph, __u32 iph_len, struct tcphdr *th, __u32 th_len) = (void *) 110;
```

152c159

```
< static long (*bpf_sk_assign)(void *ctx, struct bpf_sock *sk, __u64 flags) = (void *) 124;
```

```
> static long (*bpf_sk_assign)(void *ctx, void *sk, __u64 flags) = (void *) 124;
```

156,157c163,164

```
< static __u64 (*bpf_sk_cgroup_id)(struct bpf_sock *sk) = (void *) 128;
```

```
< static __u64 (*bpf_sk_ancestor_cgroup_id)(struct bpf_sock *sk, int ancestor_level) = (void *) 129;
```

```
> static __u64 (*bpf_sk_cgroup_id)(void *sk) = (void *) 128;
```

```
> static __u64 (*bpf_sk_ancestor_cgroup_id)(void *sk, int ancestor_level) = (void *) 129;
```

增加 bpf_helper_defs.h

```
static long (*bpf_load_hdr_opt)(struct bpf_sock_ops *skops, void *searchby_res, __u32 len, __u64 flags) = (void *) 142;
```

```
static long (*bpf_store_hdr_opt)(struct bpf_sock_ops *skops, const void *from, __u32 len, __u64 flags) = (void *) 143;
```

```
static long (*bpf_reserve_hdr_opt)(struct bpf_sock_ops *skops, __u32 len, __u64 flags) = (void *) 144;
```

```
static void (*bpf_inode_storage_get)(void *map, void *inode, void *value, __u64 flags) = (void *) 145;
```

```
static int (*bpf_inode_storage_delete)(void *map, void *inode) = (void *) 146;
```

```
static long (*bpf_d_path)(struct path *path, char *buf, __u32 sz) = (void *) 147;
```

```
static long (*bpf_copy_from_user)(void *dst, __u32 size, const void *user_ptr) = (void *) 148;
```

```
static long (*bpf_snprintf_btf)(char *str, __u32 str_size, struct btf_ptr *ptr, __u32 btf_ptr_size, __u64 flags) = (void *) 149;
```

```
static long (*bpf_seq_printf_btf)(struct seq_file *m, struct btf_ptr *ptr, __u32 ptr_size, __u64 flags) = (void *) 150;
```

```
static __u64 (*bpf_skb_cgroup_classid)(struct __sk_buff *skb) = (void *) 151;
```

```
static long (*bpf_redirect_neigh)(__u32 ifindex, struct bpf_redir_neigh *params, int plen, __u64 flags) = (void *) 152;
```

```
static void (*bpf_per_cpu_ptr)(const void *percpu_ptr, __u32 cpu) = (void *) 153;
```

```
static void (*bpf_this_cpu_ptr)(const void *percpu_ptr) = (void *) 154;
```

```
static long (*bpf_redirect_peer)(__u32 ifindex, __u64 flags) = (void *) 155;
```

```
static void (*bpf_task_storage_get)(void *map, struct task_struct *task, void *value, __u64 flags) = (void *) 156;
```

```
static long (*bpf_task_storage_delete)(void *map, struct task_struct *task) = (void *) 157;
```

```
static struct task_struct (*bpf_get_current_task_btf)(void) = (void *) 158;
```

```
static long (*bpf_bprm_opts_set)(struct linux_binprm *bprm, __u64 flags) = (void *) 159;
```

```
static __u64 (*bpf_ktime_get_coarse_ns)(void) = (void *) 160;
```

```
static long (*bpf_ima_inode_hash)(struct inode *inode, void *dst, __u32 size) = (void *) 161;
```

```
static struct socket (*bpf_sock_from_file)(struct file *file) = (void *) 162;
```

```
static long (*bpf_check_mtu)(void *ctx, __u32 ifindex, __u32 *mtu_len, __s32 len_diff, __u64 flags) = (void *) 163;
```

```
static long (*bpf_for_each_map_elem)(void *map, void *callback_fn, void *callback_ctx, __u64 flags) = (void *)
164;
```

增加 bpf_helpers.h

增加 btf.h


```
static inline bool btf_is_float(const struct btf_type *t)
```

改变 btf.h

38,45c48,57

```
< LIBBPF_API int btf_ext__reloc_func_info(const struct btf *btf,  
<                                     const struct btf_ext *btf_ext,  
<                                     const char *sec_name, __u32 insns_cnt,  
<                                     void **func_info, __u32 *cnt);  
< LIBBPF_API int btf_ext__reloc_line_info(const struct btf *btf,  
<                                     const struct btf_ext *btf_ext,  
<                                     const char *sec_name, __u32 insns_cnt,  
<                                     void **line_info, __u32 *cnt);  
____
```

```
> LIBBPF_API LIBBPF_DEPRECATED("btf_ext__reloc_func_info was never meant as a public API and has wrong  
assumptions embedded in it; it will be removed in the future libbpf versions")
```

```
> int btf_ext__reloc_func_info(const struct btf *btf,  
>                             const struct btf_ext *btf_ext,  
>                             const char *sec_name, __u32 insns_cnt,  
>                             void **func_info, __u32 *cnt);
```

```
> LIBBPF_API LIBBPF_DEPRECATED("btf_ext__reloc_line_info was never meant as a public API and has wrong  
assumptions embedded in it; it will be removed in the future libbpf versions")
```

```
> int btf_ext__reloc_line_info(const struct btf *btf,  
>                             const struct btf_ext *btf_ext,  
>                             const char *sec_name, __u32 insns_cnt,  
>                             void **line_info, __u32 *cnt);
```

改变 libbpf.h

88,89c89,91

```
< LIBBPF_API const char *bpf_program__title(const struct bpf_program *prog,  
<                                     bool needs_copy);  
____
```

```
> LIBBPF_API const char *bpf_program__section_name(const struct bpf_program *prog);
```

```
> LIBBPF_API LIBBPF_DEPRECATED("BPF program title is confusing term; please use  
bpf_program__section_name() instead")
```

```
> const char *bpf_program__title(const struct bpf_program *prog, bool needs_copy);
```

176c181

```
< LIBBPF_API enum bpf_prog_type bpf_program__get_type(struct bpf_program *prog);  
____
```

```
> LIBBPF_API enum bpf_prog_type bpf_program__get_type(const struct bpf_program *prog);
```

180c185

```
< bpf_program__get_expected_attach_type(struct bpf_program *prog);  
____
```

```
> bpf_program__get_expected_attach_type(const struct bpf_program *prog);
```

增加 libbpf.h

```
LIBBPF_API int bpf_object__set_kversion(struct bpf_object *obj, __u32 kern_version);
```

```

LIBBPF_API struct bpf_link *
bpf_program__attach_freplace(struct bpf_program *prog,
                             int target_fd, const char *attach_func_name);
LIBBPF_API struct bpf_map *bpf_map__inner_map(struct bpf_map *map);
LIBBPF_API int bpf_tc_hook_create(struct bpf_tc_hook *hook);
LIBBPF_API int bpf_tc_hook_destroy(struct bpf_tc_hook *hook);
LIBBPF_API int bpf_tc_attach(const struct bpf_tc_hook *hook,
                             struct bpf_tc_opts *opts);
LIBBPF_API int bpf_tc_detach(const struct bpf_tc_hook *hook,
                             const struct bpf_tc_opts *opts);
LIBBPF_API int bpf_tc_query(const struct bpf_tc_hook *hook,
                             struct bpf_tc_opts *opts);
LIBBPF_API int ring_buffer__epoll_fd(const struct ring_buffer *rb);
LIBBPF_API int perf_buffer__epoll_fd(const struct perf_buffer *pb);
LIBBPF_API int perf_buffer__consume_buffer(struct perf_buffer *pb, size_t buf_idx);
LIBBPF_API size_t perf_buffer__buffer_cnt(const struct perf_buffer *pb);
LIBBPF_API int perf_buffer__buffer_fd(const struct perf_buffer *pb, size_t buf_idx);
LIBBPF_API struct bpf_linker *bpf_linker__new(const char *filename, struct bpf_linker_opts *opts);
LIBBPF_API int bpf_linker__add_file(struct bpf_linker *linker,
                                     const char *filename,
                                     const struct bpf_linker_file_opts *opts);
LIBBPF_API int bpf_linker__finalize(struct bpf_linker *linker);
LIBBPF_API void bpf_linker__free(struct bpf_linker *linker);
减少 libbpf_util.h
改变 xsk.h
51,52c52
< static inline size_t xsk_ring_prod__reserve(struct xsk_ring_prod *prod,
<
size_t nb, __u32 *idx)
——
> static inline __u32 xsk_ring_prod__reserve(struct xsk_ring_prod *prod, __u32 nb, __u32 *idx)
60c60
< static inline void xsk_ring_prod__submit(struct xsk_ring_prod *prod, size_t nb)
——
> static inline void xsk_ring_prod__submit(struct xsk_ring_prod *prod, __u32 nb)
65,66c64
< static inline size_t xsk_ring_cons__peek(struct xsk_ring_cons *cons,
<
size_t nb, __u32 *idx)
——
> static inline __u32 xsk_ring_cons__peek(struct xsk_ring_cons *cons, __u32 nb, __u32 *idx)
76c73
< static inline void xsk_ring_cons__release(struct xsk_ring_cons *cons, size_t nb)
——
> static inline void xsk_ring_cons__cancel(struct xsk_ring_cons *cons, __u32 nb)
> static inline void xsk_ring_cons__release(struct xsk_ring_cons *cons, __u32 nb)

```

增加 xsk.h

```
LIBBPF_API int xsk_setup_xdp_prog(int ifindex,
                                   int *xsks_map_fd);

LIBBPF_API int xsk_socket__update_xskmap(struct xsk_socket *xsk,
                                           int xsks_map_fd);

LIBBPF_API int
xsk_socket__create_shared(struct xsk_socket **xsk_ptr,
                          const char *ifname,
                          __u32 queue_id, struct xsk_umem *umem,
                          struct xsk_ring_cons *rx,
                          struct xsk_ring_prod *tx,
                          struct xsk_ring_prod *fill,
                          struct xsk_ring_cons *comp,
                          const struct xsk_socket_config *config);
```

libtar-devel

减少 libtar.h

```
void free_longlink_longname(struct tar_header th_buf);
```

audit-devel

增加 libaudit.h

```
extern char *audit_format_signal_info(char *buf, int len, char *op, struct audit_reply *rep, char *res);

int audit_reset_backlog_wait_time_actual(int fd);

extern struct audit_rule_data *audit_rule_create_data(void);

extern void audit_rule_init_data(struct audit_rule_data *rule);
```

subunit-devel

减少 SubunitTestProgressListener.h

libsodium-devel

增加 crypto_core_ed25519.h

```
size_t crypto_core_ed25519_hashbytes(void);
SODIUM_EXPORT
size_t crypto_core_ed25519_scalarbytes(void);
SODIUM_EXPORT
size_t crypto_core_ed25519_nonreducedscalarbytes(void);
SODIUM_EXPORT
int crypto_core_ed25519_from_hash(unsigned char *p, const unsigned char *h)
    __attribute__((nonnull));

SODIUM_EXPORT
void crypto_core_ed25519_random(unsigned char *p)
    __attribute__((nonnull));

SODIUM_EXPORT
```


改变 ibusutil.h

```
1c1
< const gchar *      ibus_get_untranslated_language_name
-----
> gchar *            ibus_get_untranslated_language_name
3c3
< const gchar *      ibus_get_language_name          (const gchar * _locale);
-----
> gchar *            ibus_get_language_name          (const gchar * _locale);
```

clibcni-devel

增加 api.h、types.h、version.h

lcr-devel

增加 container_state.h

增加 cri_terminal_size.h

增加 imagetool_image_summary.h

增加 isulad_daemon_constants.h

增加 shim_client_runtime_stats.h

```
void free_shim_client_runtime_stats_data_memory_raw(shim_client_runtime_stats_data_memory_raw *ptr);
shim_client_runtime_stats_data_memory_raw *make_shim_client_runtime_stats_data_memory_raw(yajl_val tree,
const struct parser_context *ctx, parser_error *err);
yajl_gen_status gen_shim_client_runtime_stats_data_memory_raw(yajl_gen g, const
shim_client_runtime_stats_data_memory_raw *ptr, const struct parser_context *ctx, parser_error *err);
```

增加 tmpfs_options.h

daxctl-devel

增加 libdaxctl.h

```
unsigned long long daxctl_dev_get_resource(struct daxctl_dev *dev);
struct daxctl_ctx *daxctl_dev_get_ctx(struct daxctl_dev *dev);
int daxctl_dev_is_enabled(struct daxctl_dev *dev);
int daxctl_dev_disable(struct daxctl_dev *dev);
int daxctl_dev_enable_devdax(struct daxctl_dev *dev);
int daxctl_dev_enable_ram(struct daxctl_dev *dev);
int daxctl_dev_get_target_node(struct daxctl_dev *dev);
struct daxctl_memory;
struct daxctl_memory *daxctl_dev_get_memory(struct daxctl_dev *dev);
struct daxctl_dev *daxctl_memory_get_dev(struct daxctl_memory *mem);
const char *daxctl_memory_get_node_path(struct daxctl_memory *mem);
unsigned long daxctl_memory_get_block_size(struct daxctl_memory *mem);
int daxctl_memory_online(struct daxctl_memory *mem);
int daxctl_memory_offline(struct daxctl_memory *mem);
int daxctl_memory_is_online(struct daxctl_memory *mem);
int daxctl_memory_num_sections(struct daxctl_memory *mem);
```

```
int daxctl_memory_is_movable(struct daxctl_memory *mem);
int daxctl_memory_online_no_movable(struct daxctl_memory *mem);
```

libfastjson-devel

增加 json_object.h

```
extern void fjson_object_array_del_idx(struct fjson_object *jso, int idx);
extern uint32_t fjson_object_get_uint(struct fjson_object *obj);
```

e2fsprogs-devel

减少 ext2fs.h

```
extern int ext2fs_current_level_get(ext2_extent_handle_t handle);
```

libbox1-devel

减少 box.h

```
int box_add_share(const char *box_name, mode_t perm, id_t id, int id_type);
int box_remove_share(const char *box_name, id_t id, int id_type);
int box_clear_share(const char *box_name);
int box_check_owner(const char *ownername);
int box_crypto_destroy(struct box_crypto *crypt);
int box_info_destroy(struct box_info *info);
int box_passphrase_destroy(struct passphrase *password);
```

改变 box.h

```
int box_crypto_create(const char *box_name, struct box_crypto *crypto);
int box_crypto_create(const char *box_name, const char *passwd, int passwd_len);
int box_crypto_change_passphrase(const char *box_name, const struct passphrase *old_password, struct box_crypto
*new_crypto);
int box_crypto_change_passphrase(const char *box_name, char *old_passwd, int old_passwd_len, char
*new_passwd, int new_passwd_len);
int box_crypto_decrypt(const char *box_name, const struct passphrase *password);
int box_crypto_decrypt(const char *box_name, char *passwd, int passwd_len);
int box_crypto_verify_passphrase(const char *box_name, const struct passphrase *password);
bool box_crypto_verify_passphrase(const char *box_name, const char *passwd, int passwd_len);
int box_crypto_mount(const char *ownername, const char *box_name, const struct passphrase *password);
int box_crypto_mount(const char *box_name, char *passwd, int passwd_len);
int box_crypto_umount(const char *ownername, const char *box_name);
int box_crypto_umount(const char *box_name);
int box_get_info(const char *ownername, const char *box_name, struct box_info **info);
int box_get_info(const char *box_name, struct box_info *info);
```

增加 box.h

```
int box_crypto_umount_all();
int box_crypto_verify(const char *box_name);
bool box_global_key_is_exist();
int box_create_global_key(unsigned char *key, int *len);
bool box_verify_global_key(unsigned char *key, int len);
```

```

int box_crypto_change_passphrase_by_global_key(const char *box_name, unsigned char *key, unsigned int
key_len, const char *passwd, int passwd_len);
int box_crypto_set_resetkey_by_passphrase(const char *box_name, const char *passwd, unsigned int passws_len);
bool box_resetkey_status();
int box_setattr(const char *box_name);

```

harfbuzz-devel

增加 hb-buffer.h

```

HB_EXTERN hb_bool_t
hb_buffer_has_positions (hb_buffer_t *buffer);
HB_EXTERN unsigned int
hb_buffer_serialize_unicode (hb_buffer_t *buffer,
                             unsigned int start,
                             unsigned int end,
                             char *buf,
                             unsigned int buf_size,
                             unsigned int *buf_consumed,
                             hb_buffer_serialize_format_t format,
                             hb_buffer_serialize_flags_t flags);
HB_EXTERN unsigned int
hb_buffer_serialize (hb_buffer_t *buffer,
                    unsigned int start,
                    unsigned int end,
                    char *buf,
                    unsigned int buf_size,
                    unsigned int *buf_consumed,
                    hb_font_t *font,
                    hb_buffer_serialize_format_t format,
                    hb_buffer_serialize_flags_t flags);
HB_EXTERN hb_bool_t
hb_buffer_deserialize_unicode (hb_buffer_t *buffer,
                              const char *buf,
                              int buf_len,
                              const char **end_ptr,
                              hb_buffer_serialize_format_t format);

```

opencc-devel

增加 DictConverter.hpp

增加 opencc.h

```

OPENCCEXPORT opencc_t opencc_open_w(const wchar_t* configFileName);

```

增加 UTF8Util.hpp

```

static std::wstring GetPlatformString(const std::string& str)
static std::string GetPlatformString(const std::string& str)
static std::string U16ToU8(const std::wstring& wstr)

```

```
static std::wstring U8ToU16(const std::string& str)
```

systemd-devel

减少 sd-event.h

```
int sd_event_source_set_ratelimit(sd_event_source *s, uint64_t interval_usec, unsigned burst);
int sd_event_source_get_ratelimit(sd_event_source *s, uint64_t *ret_interval_usec, unsigned *ret_burst);
int sd_event_source_is_ratelimited(sd_event_source *s);
```

vala-devel

【差异很大，省略】

efl-devel

【差异很大，省略】

dpdk-devel

增加 rte_eal.h

```
> char *rte_eal_sec_get_runtime_dir(const int sec_idx);
> struct rte_config *rte_eal_sec_get_configuration(const int sec_idx);
> struct internal_config *rte_eal_sec_get_internal_config(const int sec_idx);
> int rte_eal_sec_attach(int argc, char **argv);
> int rte_eal_sec_detach(const char *file_prefix, int length);
```

增加 rte_fbararray.h

```
> int
> rte_sec_fbararray_attach(struct rte_fbararray *arr,
>                          const int switch_pri_and_sec, const int sec_idx);
> int
> rte_sec_fbararray_destroy(struct rte_fbararray *arr,
>                          const int sec_idx);
```

增加 rte_ring.h

```
> static __rte_always_inline unsigned
> rte_ring_cn_dequeue_burst(struct rte_ring *r, void **obj_table, unsigned int n)
> static __rte_always_inline void
> rte_ring_cn_enqueue(struct rte_ring *r)
> static __rte_always_inline unsigned
> rte_ring_en_dequeue_burst(struct rte_ring *r, void **obj_table, unsigned int n)
> static __rte_always_inline unsigned
> rte_ring_en_enqueue_bulk(struct rte_ring *r, void **obj_table, unsigned int n)
```

变动包清单

1. audit-devel
2. avahi-devel
3. bash-devel
4. boost-devel

5. clibcni-devel
6. daxctl-devel
7. dpdk-devel
8. e2fsprogs-devel
9. efl-devel
10. evolution-data-server-devel
11. gdk-pixbuf2-devel
12. glib2-devel
13. glibc-devel
14. glusterfs-devel
15. gobject-introspection-devel
16. harfbuzz-devel
17. httpd-devel
18. ibus-devel
19. kernel-devel
20. lcr-devel
21. libbox1-devel
22. libbpf-devel
23. libcap-ng-devel
24. libfastjson-devel
25. libgcrypt-devel
26. libkysec-devel
27. libkysecwhlist-devel
28. libldb-devel
29. libreport-devel
30. librsvg2-devel
31. libsodium-devel
32. libstdc++-devel
33. libtar-devel
34. mariadb-devel
35. ndctl-devel
36. opencc-devel
37. open-isns-devel
38. openssl-devel
39. postgresql-devel
40. rdma-core-devel
41. rpm-devel
42. subunit-devel
43. systemd-devel
44. tpm2-tss-devel
45. vala-devel

6 sp3 2403 和 sp3 2303 的 API 接口差异

libssh-devel

改变 libsshpp.hpp

211c211

```
<      const char *sourcehost=NULL, int localport=0){  
_____  
>      const char *sourcehost, int localport=0){
```

rpm-devel

改变 rpmts.h

33a34,35

```
> int rpmtsCheckUninstallPermissions(rpmts te);  
> int rpmtsCheckInstallPermissions(rpmts te);
```

改变 rpmts.h

9c9

```
< int rpmtsInitDB(rpmts ts, int dbmode);  
_____  
> int rpmtsInitDB(rpmts ts, int perms);
```

libkysec-devel

改变 libkysec.h

6a7

```
> int kysec_cmdline_get_by_pid(pid_t pid, char **cmdline);
```

改变 status.h

1,11d0

```
< static const char *func_entries[] = {  
<   KYSEC_FUNC_DIR "exectl",  
<   KYSEC_FUNC_DIR "netctl",  
<   KYSEC_FUNC_DIR "fpro",  
<   KYSEC_FUNC_DIR "kmod",  
<   KYSEC_FUNC_DIR "3adm",  
<   KYSEC_FUNC_DIR "ppro",  
<   KYSEC_FUNC_DIR "devctl",  
<   KYSEC_FUNC_DIR "iptables",  
<   KYSEC_FUNC_DIR "kysec_tpm"  
< };
```

nagios-devel

改变 nagios.h

301a302

```
> time_t reschedule_within_timeperiod(time_t, timeperiod*, time_t);
```

poppler-glib-devel

改变 poppler-action.h

7a14,19

> POPPLER_PUBLIC

```
> char *poppler_named_dest_from_bytestring(const guint8 *data,  
                                             gsize length);
```

> POPPLER_PUBLIC

```
> guint8 *poppler_named_dest_to_bytestring (const char *name,
>                                           gsize *length);
```

改变 poppler-annot.h

100a168

```
> POPPLER_PUBLIC
```

改变 poppler-attachment.h

1a2,4

```
> G_GNUC_BEGIN_IGNORE_DEPRECATIONS
```

```
> G_GNUC_END_IGNORE_DEPRECATIONS
```

```
> POPPLER_PUBLIC
```

2a6

```
> POPPLER_PUBLIC
```

5a10

```
> POPPLER_PUBLIC
```

改变 poppler-date.h

1a2

```
> POPPLER_PUBLIC
```

改变 poppler-document.h

9a13,17

> POPPLER_PUBLIC

```
> PopplerDocument    *poppler_document_new_from_bytes      (GBytes      *bytes,
>
>                                const char    *password,
>                                GError        **error);
```

64a102,110

```
> POPPLER_PUBLIC
```

```
> gchar *poppler_document_get_pdf_subtype_string (PopplerDocument *document);
```

```
> POPPLER_PUBLIC
```

```
> PopplerPDFSubtype poppler_document_get_pdf_subtype (PopplerDocument *document);
```

```
> POPPLER_PUBLIC
```

```
> PopplerPDFPart      poppler_document_get_pdf_part      (PopplerDocument *document);
```

```
> POPPLER_PUBLIC
```

```
> PopplerPDFConformance poppler_document_get_pdf_conformance (PopplerDocument *document);
```

```
> POPPLER_PUBLIC
```

65a112,121

```

> POPPLER_PUBLIC
> PopplerPrintScaling poppler_document_get_print_scaling      (PopplerDocument *document);
> POPPLER_PUBLIC
> PopplerPrintDuplex poppler_document_get_print_duplex        (PopplerDocument *document);
> POPPLER_PUBLIC
> gint                poppler_document_get_print_n_copies      (PopplerDocument *document);
> POPPLER_PUBLIC
> PopplerPageRange    *poppler_document_get_print_page_ranges  (PopplerDocument *document,
>                                                                int                *n_ranges)
G_GNUC_MALLOC;
> POPPLER_PUBLIC
66a123
> POPPLER_PUBLIC
67a125
> POPPLER_PUBLIC
68a127
> POPPLER_PUBLIC
70a130,132
> POPPLER_PUBLIC
> GTree                *poppler_document_create_dests_tree      (PopplerDocument *document);
> POPPLER_PUBLIC
72a135,141
> POPPLER_PUBLIC
> void                poppler_document_reset_form              (PopplerDocument *document,
>                                                                GList                *fields,
>                                                                gboolean              exclude_fields);
> POPPLER_PUBLIC
> gboolean              poppler_document_has_javascript          (PopplerDocument *document);
> POPPLER_PUBLIC
> GType poppler_print_scaling_get_type (void) G_GNUC_CONST;
> POPPLER_PUBLIC
> GType poppler_print_duplex_get_type (void) G_GNUC_CONST;
> POPPLER_PUBLIC
16a36,42
> POPPLER_PUBLIC
> GType poppler_pdf_subtype_get_type (void) G_GNUC_CONST;
> POPPLER_PUBLIC
> GType poppler_pdf_part_get_type (void) G_GNUC_CONST;
> POPPLER_PUBLIC
> GType poppler_pdf_conformance_get_type (void) G_GNUC_CONST;
20a50,52
> POPPLER_PUBLIC
> GType poppler_additional_action_type_get_type (void) G_GNUC_CONST;
44a100

```

```

> POPPLER_PUBLIC
改变 poppler-form-field.h
10a20,25
> POPPLER_PUBLIC
> PopplerAction      *poppler_form_field_get_additional_action    (PopplerFormField *field,
>                                                              PopplerAdditionalActionType type);
> POPPLER_PUBLIC
> gchar              *poppler_form_field_get_alternate_ui_name    (PopplerFormField *field);
> POPPLER_PUBLIC
改变 poppler.h
3a6
> POPPLER_PUBLIC
增加头文件: poppler-macros.h
改变 poppler-movie.h
6a12,25
> POPPLER_PUBLIC
> gboolean           poppler_movie_is_synchronous (PopplerMovie *poppler_movie);
> POPPLER_PUBLIC
> gdouble            poppler_movie_get_volume (PopplerMovie *poppler_movie);
> POPPLER_PUBLIC
> gdouble            poppler_movie_get_rate (PopplerMovie *poppler_movie);
> POPPLER_PUBLIC
> gushort            poppler_movie_get_rotation_angle (PopplerMovie *poppler_movie);
> POPPLER_PUBLIC
> guint64            poppler_movie_get_start (PopplerMovie *poppler_movie);
> POPPLER_PUBLIC
> guint64            poppler_movie_get_duration (PopplerMovie *poppler_movie);
> POPPLER_PUBLIC
> void               poppler_movie_get_aspect (PopplerMovie *poppler_movie, gint *width, gint *height);
改变 poppler-page.h
65a99,102
> POPPLER_PUBLIC
> gboolean           poppler_page_get_bounding_box (PopplerPage      *page,
>                                                    PopplerRectangle *rect);

```

postgresql-devel

```

改变 dependency.h
31a32
> extern void checkMembershipInCurrentExtension(const ObjectAddress *object);
改变 libpq.h
18a19
> extern bool pq_buffer_has_data(void);

```

check-devel

改变 check.h

11,14c11,13

```
< CK_DLL_EXP void CK_EXPORT _tcase_add_test(TCase * tc, TFunc tf,
<                                     const char *fname, int _signal,
<                                     int allowed_exit_value, int start,
<                                     int end);
---
> CK_DLL_EXP void CK_EXPORT _tcase_add_test(TCase * tc, const TTest * ttest,
>                                     int _signal, int allowed_exit_value,
>                                     int start, int end);
```

libabigail-devel

变动较大，略

poppler-qt5-devel

改变 poppler-annotation.h

94,95c94,95

```
<     Annotation( AnnotationPrivate &dd, const QDomNode &description );
<     void storeBaseAnnotationProperties( QDomNode & parentNode, QDomDocument & document ) const;
---
>     Annotation( AnnotationPrivate &dd, const QDomNode &annNode );
>     void storeBaseAnnotationProperties( QDomNode & annNode, QDomDocument & document ) const;
105c105
<     ~TextAnnotation();
---
>     ~TextAnnotation() override;
111a112,113
>     QColor textColor() const;
>     void setTextColor( const QColor &color );
131c133
<     ~GeomAnnotation();
---
>     ~GeomAnnotation() override;
139c141
<     ~HighlightAnnotation();
---
>     ~HighlightAnnotation() override;
152c154
<     ~StampAnnotation();
---
>     ~StampAnnotation() override;
```

163c165

```
<    ~InkAnnotation();
```

```
>    ~InkAnnotation() override;
```

173c175

```
<    ~LinkAnnotation();
```

```
>    ~LinkAnnotation() override;
```

181c183

```
<    ~CaretAnnotation();
```

```
>    ~CaretAnnotation() override;
```

187c189

```
<    ~FileAttachmentAnnotation();
```

```
>    ~FileAttachmentAnnotation() override;
```

198c200

```
<    ~SoundAnnotation();
```

```
>    ~SoundAnnotation() override;
```

203c205

```
<    void setSound( SoundObject *ef );
```

```
>    void setSound( SoundObject *s );
```

209c211

```
<    ~MovieAnnotation();
```

```
>    ~MovieAnnotation() override;
```

220c222

```
<    ~ScreenAnnotation();
```

```
>    ~ScreenAnnotation() override;
```

232c234

```
<    ~WidgetAnnotation();
```

```
>    ~WidgetAnnotation() override;
```

240c242

```
<    ~RichMediaAnnotation();
```

```
>    ~RichMediaAnnotation() override;
```

改变 poppler-form.h

10a11,19

```
>    class FormFieldIconData;
```

```
>    class POPPLER_QT5_EXPORT FormFieldIcon {
```

```

>     friend class FormFieldIconData;
>     public:
>         FormFieldIcon(FormFieldIconData *data);
>         FormFieldIcon(const FormFieldIcon &ffIcon);
>         ~FormFieldIcon();
>         FormFieldIcon& operator=(const FormFieldIcon &ffIcon);
>     };
11a21
>     friend class FormFieldData;
24a35,36
>     bool isPrintable() const;
>     void setPrintable(bool value);
29,30c41,42
<     FormField(FormFieldData &dd);
<     FormFieldData *m_formData;
---
>     FormField(std::unique_ptr<FormFieldData> dd);
>     std::unique_ptr<FormFieldData> m_formData;
36c48
<     ~FormFieldButton();
---
>     ~FormFieldButton() override;
39a52,53
>     FormFieldIcon icon() const;
>     void setIcon(const FormFieldIcon &icon);
48c62
<     ~FormFieldText();
---
>     ~FormFieldText() override;
52a67
>     void setAppearanceText( const QString& text );
65c80
<     ~FormFieldChoice();
---
>     ~FormFieldChoice() override;
68a84
>     QVector<QPair<QString,QString>> choicesWithExportValues() const;
78a95,117
>     class CertificateInfoPrivate;
>     class POPPLER_QT5_EXPORT CertificateInfo {
>     public:
>         Q_DECLARE_FLAGS(KeyUsageExtensions, KeyUsageExtension)
>         CertificateInfo(CertificateInfoPrivate *priv);
>         ~CertificateInfo();

```



```

>         bool isNull() const;
>         int version() const;
>         QByteArray serialNumber() const;
>         QString issuerInfo(EntityInfoKey key) const;
>         QString subjectInfo(EntityInfoKey key) const;
>         QDateTime validityStart() const;
>         QDateTime validityEnd() const;
>         KeyUsageExtensions keyUsageExtensions() const;
>         QByteArray publicKey() const;
>         PublicKeyType publicKeyType() const;
>         int publicKeyStrength() const;
>         bool isSelfSigned() const;
>         QByteArray certificateData() const;
>         CertificateInfo(const CertificateInfo &other);
>         CertificateInfo &operator=(const CertificateInfo &other);
>     };
>     Q_DECLARE_OPERATORS_FOR_FLAGS(CertificateInfo::KeyUsageExtensions)
87a127,128
>     QString location() const;
>     QString reason() const;
92a134
>     CertificateInfo certificateInfo() const;
101c143
<     ~FormFieldSignature();
——
>     ~FormFieldSignature() override;
改变 poppler-link.h
3,4d2
< class MovieAnnotation;
< class ScreenAnnotation;
19a18,19
> class MovieAnnotation;
> class ScreenAnnotation;
44d43
<     friend class OptContentModel;
60c59
<         ~LinkGoto();
——
>         ~LinkGoto() override;
72c71
<         ~LinkExecute();
——
>         ~LinkExecute() override;
80c79

```

```

<      ~LinkBrowse();
-----
>      ~LinkBrowse() override;
88c87
<      ~LinkAction();
-----
>      ~LinkAction() override;
95c94
<      ~LinkSound();
-----
>      ~LinkSound() override;
107c106
<      ~LinkRendition();
-----
>      ~LinkRendition() override;
118c117
<      ~LinkJavaScript();
-----
>      ~LinkJavaScript() override;
126c125
<      ~LinkMovie();
-----
>      ~LinkMovie() override;
132a132
>      friend class OptContentModel;
135c135
<      ~LinkOCGState();
-----
>      ~LinkOCGState() override;
142c142
<      ~LinkHide();
-----
>      ~LinkHide() override;
改变 poppler-optcontent.h
11c11
<      ~OptContentModel();
-----
>      ~OptContentModel() override;
改变 poppler-qt5.h
101c104
<          bool search(const QString &text, double &rectLeft, double &rectTop, double &rectRight, double
&rectBottom, SearchDirection direction, SearchFlags flags = NoSearchFlags, Rotation rotate = Rotate0) const;
-----
>          bool search(const QString &text, double &sLeft, double &sTop, double &sRight, double &sBottom,

```

SearchDirection direction, SearchFlags flags = NoSearchFlags, Rotation rotate = Rotate0) const;

119a123

> int index() const;

123a128,145

```
> class POPPLER_QT5_EXPORT OutlineItem {
>     friend class Document;
> public:
>     OutlineItem();
>     ~OutlineItem();
>     OutlineItem(const OutlineItem &other);
>     OutlineItem &operator=(const OutlineItem &other);
>     OutlineItem(OutlineItem &&other) noexcept;
>     OutlineItem &operator=(OutlineItem &&other) noexcept;
>     bool isNull() const;
>     QString name() const;
>     bool isOpen() const;
>     QSharedPointer<const LinkDestination> destination() const;
>     QString externalFileName() const;
>     QString uri() const;
>     bool hasChildren() const;
>     QVector<OutlineItem> children() const;
> };
```

135a158,160

```
> static Document *load(QIODevice *device,
>                        const QByteArray &ownerPassword=QByteArray(),
>                        const QByteArray &userPassword=QByteArray());
```

147c172

< QDateTime date(const QString & data) const;

> QDateTime date(const QString & type) const;

153c178

< QString info(const QString & data) const;

> QString info(const QString & type) const;

183c208

< QByteArray fontData(const FontInfo &font) const;

> QByteArray fontData(const FontInfo &fi) const;

186a212

> QVector<OutlineItem> outline() const;

203a230

> QVector<FormFieldSignature*> signatures() const;

231c258

< ~PSConverter();

```

-----
> ~PSConverter() override;
255c282
< ~PDFConverter();
-----
> ~PDFConverter() override;
增加头文件: poppler-version.h

```

libtommath-devel

改变 tommath.h

```

2,5c2,5
< extern int KARATSUBA_MUL_CUTOFF,
< KARATSUBA_SQR_CUTOFF,
< TOOM_MUL_CUTOFF,
< TOOM_SQR_CUTOFF;
-----
> extern int KARATSUBA_MUL_CUTOFF,
> KARATSUBA_SQR_CUTOFF,
> TOOM_MUL_CUTOFF,
> TOOM_SQR_CUTOFF;
16a17
> int mp_set_double(mp_int *a, double b);
20,26c21,28
< unsigned long mp_get_int(mp_int *a);
< unsigned long mp_get_long(mp_int *a);
< unsigned long long mp_get_long_long(mp_int *a);
< int mp_init_set (mp_int *a, mp_digit b);
< int mp_init_set_int (mp_int *a, unsigned long b);
< int mp_copy(mp_int *a, mp_int *b);
< int mp_init_copy(mp_int *a, mp_int *b);
-----
> double mp_get_double(const mp_int *a);
> unsigned long mp_get_int(const mp_int *a);
> unsigned long mp_get_long(const mp_int *a);
> unsigned long long mp_get_long_long(const mp_int *a);
> int mp_init_set(mp_int *a, mp_digit b);
> int mp_init_set_int(mp_int *a, unsigned long b);
> int mp_copy(const mp_int *a, mp_int *b);
> int mp_init_copy(mp_int *a, const mp_int *b);
28,29c30,31
< int mp_import(mp_int* rop, size_t count, int order, size_t size, int endian, size_t nails, const void* op);
< int mp_export(void* rop, size_t* countp, int order, size_t size, int endian, size_t nails, mp_int* op);
-----

```

```

> int mp_import(mp_int *rop, size_t count, int order, size_t size, int endian, size_t nails, const void *op);
> int mp_export(void *rop, size_t *countp, int order, size_t size, int endian, size_t nails, const mp_int *op);
32,36c34,38
< int mp_div_2d(mp_int *a, int b, mp_int *c, mp_int *d);
< int mp_div_2(mp_int *a, mp_int *b);
< int mp_mul_2d(mp_int *a, int b, mp_int *c);
< int mp_mul_2(mp_int *a, mp_int *b);
< int mp_mod_2d(mp_int *a, int b, mp_int *c);
----
> int mp_div_2d(const mp_int *a, int b, mp_int *c, mp_int *d);
> int mp_div_2(const mp_int *a, mp_int *b);
> int mp_mul_2d(const mp_int *a, int b, mp_int *c);
> int mp_mul_2(const mp_int *a, mp_int *b);
> int mp_mod_2d(const mp_int *a, int b, mp_int *c);
38c40
< int mp_cnt_lsb(mp_int *a);
----
> int mp_cnt_lsb(const mp_int *a);
40,90c42,102
< int mp_xor(mp_int *a, mp_int *b, mp_int *c);
< int mp_or(mp_int *a, mp_int *b, mp_int *c);
< int mp_and(mp_int *a, mp_int *b, mp_int *c);
< int mp_neg(mp_int *a, mp_int *b);
< int mp_abs(mp_int *a, mp_int *b);
< int mp_cmp(mp_int *a, mp_int *b);
< int mp_cmp_mag(mp_int *a, mp_int *b);
< int mp_add(mp_int *a, mp_int *b, mp_int *c);
< int mp_sub(mp_int *a, mp_int *b, mp_int *c);
< int mp_mul(mp_int *a, mp_int *b, mp_int *c);
< int mp_sqr(mp_int *a, mp_int *b);
< int mp_div(mp_int *a, mp_int *b, mp_int *c, mp_int *d);
< int mp_mod(mp_int *a, mp_int *b, mp_int *c);
< int mp_cmp_d(mp_int *a, mp_digit b);
< int mp_add_d(mp_int *a, mp_digit b, mp_int *c);
< int mp_sub_d(mp_int *a, mp_digit b, mp_int *c);
< int mp_mul_d(mp_int *a, mp_digit b, mp_int *c);
< int mp_div_d(mp_int *a, mp_digit b, mp_int *c, mp_digit *d);
< int mp_div_3(mp_int *a, mp_int *c, mp_digit *d);
< int mp_expt_d(mp_int *a, mp_digit b, mp_int *c);
< int mp_expt_d_ex (mp_int * a, mp_digit b, mp_int * c, int fast);
< int mp_mod_d(mp_int *a, mp_digit b, mp_digit *c);
< int mp_addmod(mp_int *a, mp_int *b, mp_int *c, mp_int *d);
< int mp_submod(mp_int *a, mp_int *b, mp_int *c, mp_int *d);
< int mp_mulmod(mp_int *a, mp_int *b, mp_int *c, mp_int *d);

```

```

< int mp_sqrmod(mp_int *a, mp_int *b, mp_int *c);
< int mp_invmod(mp_int *a, mp_int *b, mp_int *c);
< int mp_gcd(mp_int *a, mp_int *b, mp_int *c);
< int mp_exteuclid(mp_int *a, mp_int *b, mp_int *U1, mp_int *U2, mp_int *U3);
< int mp_lcm(mp_int *a, mp_int *b, mp_int *c);
< int mp_n_root(mp_int *a, mp_digit b, mp_int *c);
< int mp_n_root_ex (mp_int * a, mp_digit b, mp_int * c, int fast);
< int mp_sqrt(mp_int *arg, mp_int *ret);
< int mp_sqrtmod_prime(mp_int *arg, mp_int *prime, mp_int *ret);
< int mp_is_square(mp_int *arg, int *ret);
< int mp_jacobi(mp_int *a, mp_int *n, int *c);
< int mp_reduce_setup(mp_int *a, mp_int *b);
< int mp_reduce(mp_int *a, mp_int *b, mp_int *c);
< int mp_montgomery_setup(mp_int *a, mp_digit *mp);
< int mp_montgomery_calc_normalization(mp_int *a, mp_int *b);
< int mp_montgomery_reduce(mp_int *a, mp_int *m, mp_digit mp);
< int mp_dr_is_modulus(mp_int *a);
< void mp_dr_setup(mp_int *a, mp_digit *d);
< int mp_dr_reduce(mp_int *a, mp_int *b, mp_digit mp);
< int mp_reduce_is_2k(mp_int *a);
< int mp_reduce_2k_setup(mp_int *a, mp_digit *d);
< int mp_reduce_2k(mp_int *a, mp_int *n, mp_digit d);
< int mp_reduce_is_2k_l(mp_int *a);
< int mp_reduce_2k_setup_l(mp_int *a, mp_int *d);
< int mp_reduce_2k_l(mp_int *a, mp_int *n, mp_int *d);
< int mp_exptmod(mp_int *a, mp_int *b, mp_int *c, mp_int *d);
-----
> int mp_rand_digit(mp_digit *r);
> extern unsigned long (*ltn_rng)(unsigned char *out, unsigned long outlen, void (*callback)(void));
> extern void (*ltn_rng_callback)(void);
> int mp_xor(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_or(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_and(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_get_bit(const mp_int *a, int b);
> int mp_tc_xor(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_tc_or(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_tc_and(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_tc_div_2d(const mp_int *a, int b, mp_int *c);
> int mp_complement(const mp_int *a, mp_int *b);
> int mp_neg(const mp_int *a, mp_int *b);
> int mp_abs(const mp_int *a, mp_int *b);
> int mp_cmp(const mp_int *a, const mp_int *b);
> int mp_cmp_mag(const mp_int *a, const mp_int *b);
> int mp_add(const mp_int *a, const mp_int *b, mp_int *c);

```

```
> int mp_sub(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_mul(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_sqr(const mp_int *a, mp_int *b);
> int mp_div(const mp_int *a, const mp_int *b, mp_int *c, mp_int *d);
> int mp_mod(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_cmp_d(const mp_int *a, mp_digit b);
> int mp_add_d(const mp_int *a, mp_digit b, mp_int *c);
> int mp_sub_d(const mp_int *a, mp_digit b, mp_int *c);
> int mp_mul_d(const mp_int *a, mp_digit b, mp_int *c);
> int mp_div_d(const mp_int *a, mp_digit b, mp_int *c, mp_digit *d);
> int mp_div_3(const mp_int *a, mp_int *c, mp_digit *d);
> int mp_expt_d(const mp_int *a, mp_digit b, mp_int *c);
> int mp_expt_d_ex(const mp_int *a, mp_digit b, mp_int *c, int fast);
> int mp_mod_d(const mp_int *a, mp_digit b, mp_digit *c);
> int mp_addmod(const mp_int *a, const mp_int *b, const mp_int *c, mp_int *d);
> int mp_submod(const mp_int *a, const mp_int *b, const mp_int *c, mp_int *d);
> int mp_mulmod(const mp_int *a, const mp_int *b, const mp_int *c, mp_int *d);
> int mp_sqrmod(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_invmod(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_gcd(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_exteuclid(const mp_int *a, const mp_int *b, mp_int *U1, mp_int *U2, mp_int *U3);
> int mp_lcm(const mp_int *a, const mp_int *b, mp_int *c);
> int mp_n_root(const mp_int *a, mp_digit b, mp_int *c);
> int mp_n_root_ex(const mp_int *a, mp_digit b, mp_int *c, int fast);
> int mp_sqrt(const mp_int *arg, mp_int *ret);
> int mp_sqrtmod_prime(const mp_int *n, const mp_int *prime, mp_int *ret);
> int mp_is_square(const mp_int *arg, int *ret);
> int mp_jacobi(const mp_int *a, const mp_int *n, int *c);
> int mp_kronecker(const mp_int *a, const mp_int *p, int *c);
> int mp_reduce_setup(mp_int *a, const mp_int *b);
> int mp_reduce(mp_int *x, const mp_int *m, const mp_int *mu);
> int mp_montgomery_setup(const mp_int *n, mp_digit *rho);
> int mp_montgomery_calc_normalization(mp_int *a, const mp_int *b);
> int mp_montgomery_reduce(mp_int *x, const mp_int *n, mp_digit rho);
> int mp_dr_is_modulus(const mp_int *a);
> void mp_dr_setup(const mp_int *a, mp_digit *d);
> int mp_dr_reduce(mp_int *x, const mp_int *n, mp_digit k);
> int mp_reduce_is_2k(const mp_int *a);
> int mp_reduce_2k_setup(const mp_int *a, mp_digit *d);
> int mp_reduce_2k(mp_int *a, const mp_int *n, mp_digit d);
> int mp_reduce_is_2k_l(const mp_int *a);
> int mp_reduce_2k_setup_l(const mp_int *a, mp_int *d);
> int mp_reduce_2k_l(mp_int *a, const mp_int *n, const mp_int *d);
> int mp_exptmod(const mp_int *G, const mp_int *X, const mp_int *P, mp_int *Y);
```

92,94c104,106

```
< int mp_prime_is_divisible(mp_int *a, int *result);  
< int mp_prime_fermat(mp_int *a, mp_int *b, int *result);  
< int mp_prime_miller_rabin(mp_int *a, mp_int *b, int *result);  
----
```

```
> int mp_prime_is_divisible(const mp_int *a, int *result);  
> int mp_prime_fermat(const mp_int *a, const mp_int *b, int *result);  
> int mp_prime_miller_rabin(const mp_int *a, const mp_int *b, int *result);
```

96c108,110

```
< int mp_prime_is_prime(mp_int *a, int t, int *result);  
----
```

```
> int mp_prime_strong_lucas_selfridge(const mp_int *a, int *result);  
> int mp_prime_frobenius_underwood(const mp_int *N, int *result);  
> int mp_prime_is_prime(const mp_int *a, int t, int *result);
```

99,100c113,114

```
< int mp_count_bits(mp_int *a);  
< int mp_unsigned_bin_size(mp_int *a);  
----
```

```
> int mp_count_bits(const mp_int *a);  
> int mp_unsigned_bin_size(const mp_int *a);
```

102,104c116,118

```
< int mp_to_unsigned_bin(mp_int *a, unsigned char *b);  
< int mp_to_unsigned_bin_n (mp_int * a, unsigned char *b, unsigned long *outlen);  
< int mp_signed_bin_size(mp_int *a);  
----
```

```
> int mp_to_unsigned_bin(const mp_int *a, unsigned char *b);  
> int mp_to_unsigned_bin_n(const mp_int *a, unsigned char *b, unsigned long *outlen);  
> int mp_signed_bin_size(const mp_int *a);
```

106,107c120,121

```
< int mp_to_signed_bin(mp_int *a, unsigned char *b);  
< int mp_to_signed_bin_n (mp_int * a, unsigned char *b, unsigned long *outlen);  
----
```

```
> int mp_to_signed_bin(const mp_int *a, unsigned char *b);  
> int mp_to_signed_bin_n(const mp_int *a, unsigned char *b, unsigned long *outlen);
```

109,111c123,125

```
< int mp_toradix(mp_int *a, char *str, int radix);  
< int mp_toradix_n(mp_int * a, char *str, int radix, int maxlen);  
< int mp_radix_size(mp_int *a, int radix, int *size);  
----
```

```
> int mp_toradix(const mp_int *a, char *str, int radix);  
> int mp_toradix_n(const mp_int *a, char *str, int radix, int maxlen);  
> int mp_radix_size(const mp_int *a, int radix, int *size);
```

113,114c127,128

```
< int mp_fwrite(mp_int *a, int radix, FILE *stream);
```



```
<    }
```

```
——
```

```
> int mp_fwrite(const mp_int *a, int radix, FILE *stream);
```

```
> }
```

nss-softokn-devel

改变 blapi.h

517a518,532

```
> extern SM3Context *SM3_NewContext(void);
```

```
> extern void SM3_DestroyContext(SM3Context *cx, PRBool freeit);
```

```
> extern void SM3_Begin(SM3Context *cx);
```

```
> extern void SM3_Update(SM3Context *cx, const unsigned char *input,
```

```
>                                unsigned int inputLen);
```

```
> extern void SM3_End(SM3Context *cx, unsigned char *digest,
```

```
>                                unsigned int *digestLen, unsigned int maxDigestLen);
```

```
> extern SECStatus SM3_HashBuf(unsigned char *dest, const unsigned char *src,
```

```
>                                PRUint32 src_length);
```

```
> extern SECStatus SM3_Hash(unsigned char *dest, const char *src);
```

```
> extern void SM3_TraceState(SM3Context *cx);
```

```
> extern unsigned int SM3_FlattenSize(SM3Context *cx);
```

```
> extern SECStatus SM3_Flatten(SM3Context *cx, unsigned char *space);
```

```
> extern SM3Context *SM3_Resurrect(unsigned char *space, void *arg);
```

```
> extern void SM3_Clone(SM3Context *dest, SM3Context *src);
```

改变 blapit.h

11a12

```
> struct SM3ContextStr;
```

bind-devel

改变 adb.h

84c84

```
< dns_adbentry_overquota(dns_adbentry_t *entry);
```

```
——
```

```
> dns_adbentry_overquota(dns_adb_t *adb, dns_adbentry_t *entry);
```

改变 message.h

7c7,9

```
< dns_message_destroy(dns_message_t **msgp);
```

```
——
```

```
> dns_message_attach(dns_message_t *source, dns_message_t **target);
```

```
> void
```

```
> dns_message_detach(dns_message_t **messagep);
```

改变 refcount.h

1a2

```
> unsigned int isc_refcount_current(isc_refcount_t *rp);
```

kmod-devel

改变 libkmod.h

52a53

```
> KMOD_REMOVE_REPLACE = 0x1000,
```

kysec-common-devel

改变 db.h

19a20,27

```
> int sqlite_prepare_stmt(sqlite3 *sqlite, const char *sql, sqlite3_stmt **stmt);
```

```
> int sqlite_bind_int_stmt(sqlite3_stmt* stmt, int index, int value);
```

```
> int sqlite_bind_text_stmt(sqlite3_stmt* stmt, int index, const char* value);
```

```
> int sqlite_column_int_stmt(sqlite3_stmt*, int column);
```

```
> const unsigned char *sqlite_column_text_stmt(sqlite3_stmt*, int column);
```

```
> int sqlite_reset_stmt(sqlite3_stmt *stmt);
```

```
> int sqlite_step_stmt(sqlite3_stmt *stmt);
```

```
> int sqlite_finalize_stmt(sqlite3_stmt *stmt);
```

改变 kysec_log.h

1,4c1

```
< extern "C" {
```

```
< void kysec_log(int type, char *op, char *obj, char *format, ...);
```

```
< }
```

```
<
```

```
----
```

```
> }
```

增加头文件: kysec_xz_decompress.h

espeak-ng-devel

改变 espeak_ng.h

31a32,33

```
> espeak_ng_SetVoiceByFile(const char *filename);
```

```
> ESPEAK_NG_API espeak_ng_STATUS
```

82a85,88

```
> ESPEAK_NG_API espeak_ng_STATUS
```

```
> espeak_ng_SetOutputHooks(espeak_ng_OUTPUT_HOOKS* hooks);
```

```
> ESPEAK_NG_API espeak_ng_STATUS
```

```
> espeak_ng_SetConstF0(int f0);
```

改变 speak_lib.h

7a8,9

```
> ESPEAK_API void espeak_SetPhonemeCallback(int (*PhonemeCallback)(const char *));
```

```
> extern "C"
41a44,45
> extern "C"
> ESPEAK_API espeak_ERROR espeak_SetVoiceByFile(const char *filename);
```

grpc-devel

减少头文件: channel_test_peer.h
减少头文件: default_reactor_test_peer.h
减少头文件: mock_stream.h
减少头文件: server_context_test_spouse.h

hsdimm-lite-devel

改变 libdimm.h
5a6,9
> extern int mem_dev_get_driver_version();
> extern int mem_dev_check_virtualized_env();
> extern __attribute__((constructor)) void mem_dev_init(void);
> extern void mem_dev_init(void);

evolution-data-server-devel

改变 camel-stream-buffer.h
12a13,14
> void camel_stream_buffer_discard_cache
> (CamelStreamBuffer *sbf);

openssl-devel

改变 engine.h
9c9
< void ENGINE_load_builtin_engines(void);
——
> void ENGINE_load_builtin_engines(void);
改变 evp.h
455a456
> int EVP_PKEY_is_sm2(EVP_PKEY *pkey);
651d651
< EVP_PKEY_CTX *EVP_PKEY_CTX_new_pkey_id(EVP_PKEY *pkey, int id, ENGINE *e);
改变 ssl.h
219a220,222
> __owur int SSL_use_gm_PrivateKey(SSL *ssl, EVP_PKEY *pkey, int usage);
> __owur int SSL_use_gm_PrivateKey_ASN1(int pk, SSL *ssl, const unsigned char *d,

```

> long len, int usage);
223c226,230
< __owur int SSL_use_certificate(SSL *ssl, X509 *x);
---
> __owur int SSL_use_gm_certificate(SSL *ssl, X509 *x, int usage);
> __owur int SSL_use_gm_certificate_ASN1(SSL *ssl, const unsigned char *d, int len, int usage);
> __owur int SSL_use_gm_cert_and_key(SSL *ssl, X509 *x509, EVP_PKEY *privatekey,
> STACK_OF(X509) *chain, int override, int usage);
> __owur int SSL_use_certificate(SSL *ssl, X509 *x);
233a241,242
> __owur int SSL_use_gm_PrivateKey_file(SSL *ssl, const char *file, int type, int usage);
> __owur int SSL_use_gm_certificate_file(SSL *ssl, const char *file, int type, int usage);
237a247,250
> __owur int SSL_CTX_use_gm_PrivateKey_file(SSL_CTX *ctx, const char *file,
> int type, int usage);
> __owur int SSL_CTX_use_gm_certificate_file(SSL_CTX *ctx, const char *file,
> int type, int usage);
322c335,343
< __owur int SSL_CTX_use_PrivateKey(SSL_CTX *ctx, EVP_PKEY *pkey);
---
> __owur int SSL_CTX_use_gm_PrivateKey(SSL_CTX *ctx, EVP_PKEY *pkey, int usage);
> __owur int SSL_CTX_use_gm_PrivateKey_ASN1(int pk, SSL_CTX *ctx,
> const unsigned char *d, long len, int usage);
> __owur int SSL_CTX_use_gm_certificate(SSL_CTX *ctx, X509 *x, int usage);
> __owur int SSL_CTX_use_gm_certificate_ASN1(SSL_CTX *ctx, int len,
> const unsigned char *d, int usage);
> __owur int SSL_CTX_use_gm_cert_and_key(SSL_CTX *ctx, X509 *x509, EVP_PKEY *privatekey,
> STACK_OF(X509) *chain, int override, int usage);
> __owur int SSL_CTX_use_PrivateKey(SSL_CTX *ctx, EVP_PKEY *pkey);
438a460,462
> __owur const SSL_METHOD *TLCP_method(void);
> __owur const SSL_METHOD *TLCP_server_method(void);
> __owur const SSL_METHOD *TLCP_client_method(void);

```

~~mariadb-devel~~

变动很大，略

~~proftpd-devel~~

增加头文件：acconfig.h

改变 auth.h

31a32,33

```

> char *pr_auth_bcrypt(pool *p, const char *key, const char *salt,
> size_t *hashed_len);

```

改变 bindings.h

25c25,26

```
< int pr_namebind_open(const char *name, const pr_netaddr_t *addr);
```

```
> int pr_namebind_open(const char *name, const pr_netaddr_t *addr,
```

```
>   unsigned int port);
```

改变 class.h

1c1

```
< const pr_class_t *pr_class_find(const char *);
```

```
> const pr_class_t *pr_class_find(const char *name);
```

3,4c3,6

```
< const pr_class_t *pr_class_match_addr(const pr_netaddr_t *);
```

```
< int pr_class_open(pool *, const char *);
```

```
> int pr_class_satisfied(pool *p, const pr_class_t *cls,
```

```
>   const pr_netaddr_t *addr);
```

```
> const pr_class_t *pr_class_match_addr(const pr_netaddr_t *addr);
```

```
> int pr_class_open(pool *p, const char *name);
```

6,8c8,10

```
< int pr_class_add_acl(const pr_netacl_t *);
```

```
< int pr_class_set_satisfy(int);
```

```
< int pr_class_add_note(const char *, void *, size_t);
```

```
> int pr_class_add_acl(const pr_netacl_t *acl);
```

```
> int pr_class_set_satisfy(int flags);
```

```
> int pr_class_add_note(const char *key, void *val, size_t valsz);
```

改变 conf.h

1,4d0

```
<         char *strchr(),*strrchr());
```

```
< off_t lseek(int, off_t, int);
```

```
< extern int optopt;
```

```
<                                     extern char *alloca();
```

改变 data.h

5a6

```
> void pr_data_close2(void);
```

增加头文件: error.h

改变 fsio.h

2d1

```
< int pr_fsio_stat_canon(const char *, struct stat *);
```

5d3

```
< int pr_fsio_lstat_canon(const char *, struct stat *);
```

7d4

```
< int pr_fsio_readlink_canon(const char *, char *, size_t);
```

16d12

```
< int pr_fsio_rename_canon(const char *, const char *);
```

19d14

```
< int pr_fsio_unlink_canon(const char *);
```

23a19

```
> ssize_t pr_fsio_pread(pr_fh_t *, void *, size_t, off_t);
```

24a21

```
> ssize_t pr_fsio_pwrite(pr_fh_t *, const void *, size_t, off_t);
```

26d22

```
< int pr_fsio_link_canon(const char *, const char *);
```

28d23

```
< int pr_fsio_symlink_canon(const char *, const char *);
```

31d25

```
< int pr_fsio_truncate_canon(const char *, off_t);
```

34d27

```
< int pr_fsio_chmod_canon(const char *, mode_t);
```

38d30

```
< int pr_fsio_chown_canon(const char *, uid_t, gid_t);
```

58a51,78

```
> int pr_fsio_chmod_with_error(pool *p, const char *path, mode_t mode,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_chown_with_error(pool *p, const char *path, uid_t uid, gid_t gid,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_chroot_with_error(pool *p, const char *path, pr_error_t **err);
```

```
> int pr_fsio_close_with_error(pool *p, pr_fh_t *fh, pr_error_t **err);
```

```
> int pr_fsio_fchmod_with_error(pool *p, pr_fh_t *fh, mode_t mode,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_fchown_with_error(pool *p, pr_fh_t *fh, uid_t uid, gid_t gid,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_lchown_with_error(pool *p, const char *path, uid_t uid, gid_t gid,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_lstat_with_error(pool *p, const char *path, struct stat *st,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_mkdir_with_error(pool *p, const char *path, mode_t mode,
```

```
>   pr_error_t **err);
```

```
> pr_fh_t *pr_fsio_open_with_error(pool *p, const char *path, int flags,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_read_with_error(pool *p, pr_fh_t *fh, char *buf, size_t sz,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_rename_with_error(pool *p, const char *from, const char *to,
```

```
>   pr_error_t **err);
```

```
> int pr_fsio_rmdir_with_error(pool *p, const char *path, pr_error_t **err);
```

```
> int pr_fsio_stat_with_error(pool *p, const char *path, struct stat *st,
```

```
>   pr_error_t **err);
```

```

> int pr_fsio_unlink_with_error(pool *p, const char *path, pr_error_t **err);
> int pr_fsio_write_with_error(pool *p, pr_fh_t *fh, const char *buf, size_t sz,
>   pr_error_t **err);
103a124
> void pr_fs_close_extra_fds(void);
增加头文件: jot.h
改变 json.h
61a62
> const char *pr_json_type_name(unsigned int json_type);
改变 log.h
31a32
> int pr_log_set_options(unsigned long log_opts);
改变 mod_quotatab.h
1c1
< quota_deltas_t quotatab_deltas;
----
> extern quota_deltas_t quotatab_deltas;
增加头文件: openbsd-blowfish.h
增加头文件: os.h
改变 pool.h
11a12
> const char *pr_pool_get_tag(struct pool_rec *);
25a27
> void register_cleanup2(pool *, void *, void (*)(void *));
改变 proftpd.h
0a1
> struct class_struc;
增加头文件: random.h
改变 redis.h
1c1
< pr_redis_t *pr_redis_conn_get(pool *p);
----
> pr_redis_t *pr_redis_conn_get(pool *p, unsigned long flags);
7a8
> int pr_redis_select(pr_redis_t *redis, const char *db_idx);
202c203,209
< int redis_set_server(const char *server, int port, const char *password);
----
> int pr_redis_sentinel_get_master_addr(pool *p, pr_redis_t *redis,
>   const char *name, pr_netaddr_t **addr);
> int pr_redis_sentinel_get_masters(pool *p, pr_redis_t *redis,
>   array_header **masters);
> int redis_set_server(const char *server, int port, unsigned long flags,
>   const char *password, const char *db_idx);

```

```

> int redis_set_sentinels(array_header *sentinels, const char *name);
改变 response.h
7a8
> int pr_response_blocked(void);
改变 str.h
4,5c4,9
< char *pdircat(pool *, ...);
< char *pstrcat(pool *, ...);
——
> char *pdircat(pool *, ...)
>     __attribute__((sentinel));
>     ;
> char *pstrcat(pool *, ...)
>     __attribute__((sentinel));
>     ;
改变 support.h
30c30,31
< int check_shutmsg(const char *, time_t *, time_t *, time_t *, char *, size_t);
——
> int check_shutmsg(pool *, const char *, time_t *, time_t *, time_t *, char *,
>     size_t);
36a38
> const char *pr_strtime3(pool *, time_t, int);
38a41,50
> int pr_snprintf(char *buf, size_t bufsz, const char *fmt, ...)
>     __attribute__((format (printf, 3, 4)));
>     ;
> int pr_snprintf1(const char *file, int lineno, char *buf, size_t bufsz,
>     const char *fmt, ...)
>     __attribute__((format (printf, 5, 6)));
>     ;
> int pr_vsnprintf(char *buf, size_t bufsz, const char *fmt, va_list msg);
> int pr_vsnprintf1(const char *file, int lineno, char *buf, size_t bufsz,
>     const char *fmt, va_list msg);
改变 version.h
3a4
>

```

openvswitch-devel

```

改变 classifier.h
3a4
> struct mf_field;
改变 dp-packet.h

```


11a12,13

> static inline void

>

改变 meta-flow.h

92a93,94

> void mf_set_mask_l3_prereqs(const struct mf_field *, const struct flow *,

> struct flow_wildcards *);

改变 odp-util.h

21,25c21,26

< size_t odp_put_userspace_action(uint32_t pid,

< const void *userdata, size_t userdata_size,

< odp_port_t tunnel_out_port,

< bool include_actions,

< struct ofpbuf *odp_actions);

> int odp_put_userspace_action(uint32_t pid,

> const void *userdata, size_t userdata_size,

> odp_port_t tunnel_out_port,

> bool include_actions,

> struct ofpbuf *odp_actions,

> size_t *odp_actions_ofs);

减少头文件: rte_flow.h

改变 sha1.h

2c2

< void sha1_update(struct sha1_ctx *, const void *, size_t);

> void sha1_update(struct sha1_ctx *, const void *, uint32_t size);

4c4

< void sha1_bytes(const void *, size_t, uint8_t digest[SHA1_DIGEST_SIZE]);

> void sha1_bytes(const void *, uint32_t size, uint8_t digest[SHA1_DIGEST_SIZE]);

libwinpr-devel

增加头文件: assert.h

改变 crypto.h

1,96d0

< *PCRYPT_OBJID_BLOB, CERT_NAME_BLOB, *PCERT_NAME_BLOB, CERT_RDN_VALUE_BLOB,

< *PCERT_RDN_VALUE_BLOB, CERT_BLOB, *PCERT_BLOB, CRL_BLOB, *PCRL_BLOB, DATA_BLOB,
*PDATA_BLOB,

< CRYPT_DATA_BLOB, *PCRYPT_DATA_BLOB, CRYPT_HASH_BLOB, *PCRYPT_HASH_BLOB,
CRYPT_DIGEST_BLOB,

< *PCRYPT_DIGEST_BLOB, CRYPT_DER_BLOB, *PCRYPT_DER_BLOB, CRYPT_ATTR_BLOB,
*PCRYPT_ATTR_BLOB;

```

< HCERTSTORE CertOpenStore(LPCSTR lpszStoreProvider, DWORD dwMsgAndCertEncodingType,
<
<         HCRYPTPROV_LEGACY hCryptProv, DWORD dwFlags, const void* pvPara);
<  WINPR_API  HCERTSTORE  CertOpenSystemStoreW(HCRYPTPROV_LEGACY  hProv,  LPCWSTR
szSubsystemProtocol);
<  WINPR_API  HCERTSTORE  CertOpenSystemStoreA(HCRYPTPROV_LEGACY  hProv,  LPCSTR
szSubsystemProtocol);
< WINPR_API BOOL CertCloseStore(HCERTSTORE hCertStore, DWORD dwFlags);
< extern "C"
< {
<  WINPR_API PCCERT_CONTEXT CertFindCertificateInStore(HCERTSTORE hCertStore,
<
<         DWORD dwCertEncodingType, DWORD
dwFindFlags,
<
<         DWORD dwFindType, const void*
pvFindPara,
<
<         PCCERT_CONTEXT pPrevCertContext);
<  WINPR_API PCCERT_CONTEXT CertEnumCertificatesInStore(HCERTSTORE hCertStore,
<
<         PCCERT_CONTEXT pPrevCertContext);
<  WINPR_API DWORD CertGetNameStringW(PCCERT_CONTEXT pCertContext, DWORD dwType, DWORD
dwFlags,
<
<         void* pvTypePara, LPWSTR pszNameString, DWORD
cchNameString);
<  WINPR_API DWORD CertGetNameStringA(PCCERT_CONTEXT pCertContext, DWORD dwType, DWORD
dwFlags,
<
<         void* pvTypePara, LPSTR pszNameString, DWORD
cchNameString);
< }
< extern "C"
< {
<  WINPR_API BOOL CryptProtectMemory(LPVOID pData, DWORD cbData, DWORD dwFlags);
<  WINPR_API BOOL CryptUnprotectMemory(LPVOID pData, DWORD cbData, DWORD dwFlags);
<  WINPR_API BOOL CryptProtectData(DATA_BLOB* pDataIn, LPCWSTR szDataDescr,
<
<         DATA_BLOB* pOptionalEntropy, PVOID pvReserved,
<
<         CRYPTPROTECT_PROMPTSTRUCT* pPromptStruct, DWORD
dwFlags,
<
<         DATA_BLOB* pDataOut);
<  WINPR_API BOOL CryptUnprotectData(DATA_BLOB* pDataIn, LPWSTR* ppszDataDescr,
<
<         DATA_BLOB* pOptionalEntropy, PVOID pvReserved,
<
<         CRYPTPROTECT_PROMPTSTRUCT* pPromptStruct, DWORD
dwFlags,
<
<         DATA_BLOB* pDataOut);
< }
< BOOL CryptStringToBinaryW(LPCWSTR pszString, DWORD cchString, DWORD dwFlags, BYTE* pbBinary,
<
<         DWORD* pcbBinary, DWORD* pdwSkip, DWORD* pdwFlags);
< BOOL CryptStringToBinaryA(LPCSTR pszString, DWORD cchString, DWORD dwFlags, BYTE* pbBinary,

```

```

<                                DWORD* pcbBinary, DWORD* pdwSkip, DWORD* pdwFlags);
< BOOL CryptBinaryToStringW(CONST BYTE* pbBinary, DWORD cbBinary, DWORD dwFlags, LPWSTR
pszString,
<                                DWORD* pcchString);
< BOOL CryptBinaryToStringA(CONST BYTE* pbBinary, DWORD cbBinary, DWORD dwFlags, LPSTR pszString,
<                                DWORD* pcchString);
< extern "C"
< {
<     WINPR_API WINPR_MD_TYPE winpr_md_type_from_string(const char* name);
<     WINPR_API const char* winpr_md_type_to_string(WINPR_MD_TYPE md);
<     WINPR_API WINPR_HMAC_CTX* winpr_HMAC_New(void);
<     WINPR_API BOOL winpr_HMAC_Init(WINPR_HMAC_CTX* ctx, WINPR_MD_TYPE md, const BYTE*
key,
<                                size_t keylen);
<     WINPR_API BOOL winpr_HMAC_Update(WINPR_HMAC_CTX* ctx, const BYTE* input, size_t ilen);
<     WINPR_API BOOL winpr_HMAC_Final(WINPR_HMAC_CTX* ctx, BYTE* output, size_t ilen);
<     WINPR_API void winpr_HMAC_Free(WINPR_HMAC_CTX* ctx);
<     WINPR_API BOOL winpr_HMAC(WINPR_MD_TYPE md, const BYTE* key, size_t keylen, const BYTE*
input,
<                                size_t ilen, BYTE* output, size_t olen);
< }
< extern "C"
< {
<     WINPR_API WINPR_DIGEST_CTX* winpr_Digest_New(void);
<     WINPR_API BOOL winpr_Digest_Init_Allow_FIPS(WINPR_DIGEST_CTX* ctx, WINPR_MD_TYPE md);
<     WINPR_API BOOL winpr_Digest_Init(WINPR_DIGEST_CTX* ctx, WINPR_MD_TYPE md);
<     WINPR_API BOOL winpr_Digest_Update(WINPR_DIGEST_CTX* ctx, const BYTE* input, size_t ilen);
<     WINPR_API BOOL winpr_Digest_Final(WINPR_DIGEST_CTX* ctx, BYTE* output, size_t ilen);
<     WINPR_API void winpr_Digest_Free(WINPR_DIGEST_CTX* ctx);
<     WINPR_API BOOL winpr_Digest_Allow_FIPS(WINPR_MD_TYPE md, const BYTE* input, size_t ilen,
<                                BYTE* output, size_t olen);
<     WINPR_API BOOL winpr_Digest(WINPR_MD_TYPE md, const BYTE* input, size_t ilen, BYTE* output,
<                                size_t olen);
< }
< extern "C"
< {
<     WINPR_API int winpr_RAND(BYTE* output, size_t len);
<     WINPR_API int winpr_RAND_pseudo(BYTE* output, size_t len);
< }
< extern "C"
< {
<     WINPR_API WINPR_RC4_CTX* winpr_RC4_New_Allow_FIPS(const BYTE* key, size_t keylen);
<     WINPR_API WINPR_RC4_CTX* winpr_RC4_New(const BYTE* key, size_t keylen);
<     WINPR_API BOOL winpr_RC4_Update(WINPR_RC4_CTX* ctx, size_t length, const BYTE* input,

```

```

<                                     BYTE* output);
<   WINPR_API void winpr_RC4_Free(WINPR_RC4_CTX* ctx);
<}
< extern "C"
< {
<   WINPR_API WINPR_CIPHER_CTX* winpr_Cipher_New(int cipher, int op, const BYTE* key,
<                                     const BYTE* iv);
<   WINPR_API BOOL winpr_Cipher_Update(WINPR_CIPHER_CTX* ctx, const BYTE* input, size_t ilen,
<                                     BYTE* output, size_t* olen);
<   WINPR_API BOOL winpr_Cipher_Final(WINPR_CIPHER_CTX* ctx, BYTE* output, size_t* olen);
<   WINPR_API void winpr_Cipher_Free(WINPR_CIPHER_CTX* ctx);
<}
< extern "C"
< {
<   WINPR_API int winpr_Cipher_BytesToKey(int cipher, int md, const BYTE* salt, const BYTE* data,
<                                     int datal, int count, BYTE* key, BYTE* iv);
<}

```

增加头文件: custom-crypto.h

改变 locale.h

3,8c3,8

```

<   DWORD WINAPI FormatMessageA(DWORD dwFlags, LPCVOID lpSource, DWORD dwMessageId,
<                                     DWORD dwLanguageId, LPSTR lpBuffer, DWORD nSize,
<                                     va_list* Arguments);
<   DWORD WINAPI FormatMessageW(DWORD dwFlags, LPCVOID lpSource, DWORD dwMessageId,
<                                     DWORD dwLanguageId, LPWSTR lpBuffer, DWORD nSize,
<                                     va_list* Arguments);
——
>   WINPR_API  DWORD  WINAPI  FormatMessageA(DWORD  dwFlags,  LPCVOID  lpSource,  DWORD
dwMessageId,
>                                     DWORD dwLanguageId, LPSTR lpBuffer, DWORD nSize,
>                                     va_list* Arguments);
>   WINPR_API  DWORD  WINAPI  FormatMessageW(DWORD  dwFlags,  LPCVOID  lpSource,  DWORD
dwMessageId,
>                                     DWORD dwLanguageId, LPWSTR lpBuffer, DWORD nSize,
>                                     va_list* Arguments);

```

改变 smartcard.h

3,5c3,5

```

<   extern const SCARD_IO_REQUEST g_rgSCardT0Pci;
<   extern const SCARD_IO_REQUEST g_rgSCardT1Pci;
<   extern const SCARD_IO_REQUEST g_rgSCardRawPci;
——
>   WINPR_API extern const SCARD_IO_REQUEST g_rgSCardT0Pci;
>   WINPR_API extern const SCARD_IO_REQUEST g_rgSCardT1Pci;
>   WINPR_API extern const SCARD_IO_REQUEST g_rgSCardRawPci;

```

改变 stream.h

7a8,16

```
> WINPR_API BOOL Stream_CheckAndLogRequiredLengthEx(const char* tag, DWORD level, wStream* s,
>
>                                     UINT64 len, const char* fmt, ...);
> WINPR_API BOOL Stream_CheckAndLogRequiredLengthExVa(const char* tag, DWORD level, wStream* s,
>
>                                     UINT64 len, const char* fmt, va_list args);
> WINPR_API BOOL Stream_CheckAndLogRequiredLengthWLogEx(wLog* log, DWORD level, wStream* s,
>
>                                     UINT64 len, const char* fmt, ...);
> WINPR_API BOOL Stream_CheckAndLogRequiredLengthWLogExVa(wLog* log, DWORD level, wStream* s,
>
>                                     UINT64 len, const char* fmt,
>                                     va_list args);
```

改变 string.h

2a3,4

```
> WINPR_API BOOL winpr_str_append(const char* what, char* buffer, size_t size,
>
>                                     const char* separator);
```

7a10

```
> WINPR_API int _wcsncmp(const WCHAR* string1, const WCHAR* string2, size_t count);
```

9a13

```
> WINPR_API WCHAR* _wcsstr(const WCHAR* str, const WCHAR* strSearch);
```

改变 thread.h

66c66

```
< WINPR_API DECLSPEC_NORETURN VOID ExitThread(DWORD dwExitCode);
```

——

```
> WINPR_API VOID ExitThread(DWORD dwExitCode);
```

增加头文件: wincrypt.h

freerdp-devel

增加头文件: ainput.h

增加头文件: cliprdr_utils.h

改变 color.h

27a28,31

```
> case PIXEL_FORMAT_BGRX32_DEPTH30:
>
>     return "PIXEL_FORMAT_BGRX32_DEPTH30";
> case PIXEL_FORMAT_RGBX32_DEPTH30:
>
>     return "PIXEL_FORMAT_RGBX32_DEPTH30";
```

339a344

```
>     UINT32 t;
```

357a363,368

```
> case PIXEL_FORMAT_RGBX32_DEPTH30:
>
>     t = (_r << 22) | (_g << 12) | (_b << 2);
>
>     return ((t & 0xff) << 24) | (((t >> 8) & 0xff) << 16) | (((t >> 16) & 0xff) << 8) | (t >> 24);
> case PIXEL_FORMAT_BGRX32_DEPTH30:
>
>     t = (_r << 22) | (_g << 12) | (_b << 2);
```

```

> return ((t & 0xff) << 24) | (((t >> 8) & 0xff) << 16) | (((t >> 16) & 0xff) << 8) | (t >> 24);
改变 crypto.h
3a4
> FREERDP_API WINPR_MD_TYPE crypto_cert_get_signature_alg(X509* xcert);
25,27c26,28
< extern const BYTE tssk_modulus[];
< extern const BYTE tssk_privateExponent[];
< extern const BYTE tssk_exponent[];
——
> WINPR_API extern const BYTE tssk_modulus[];
> WINPR_API extern const BYTE tssk_privateExponent[];
> WINPR_API extern const BYTE tssk_exponent[];
改变 freerdp.h
3c3
< FREERDP_API
BOOL freerdp_context_new(freerdp* instance);
——
>
FREERDP_API BOOL freerdp_context_new(freerdp* instance);
68a69,70
> FREERDP_API CONNECTION_STATE freerdp_get_state(rdpContext* context);
> FREERDP_API const char* freerdp_state_string(CONNECTION_STATE state);
改变 peer.h
4a5,6
> FREERDP_API const char* freerdp_peer_os_major_type_string(freerdp_peer* client);
> FREERDP_API const char* freerdp_peer_os_minor_type_string(freerdp_peer* client);
增加头文件: rdpecam-enumerator.h
增加头文件: rdpecam.h
改变 settings.h
32a33,34
> FREERDP_API BOOL freerdp_target_net_addresses_copy(rdpSettings* settings, char** addresses,
>
UINT32 count);
81a84,85
> FREERDP_API char* freerdp_rail_support_flags_to_string(UINT32 flags, char* buffer,
>
size_t length);
增加头文件: string.h
增加头文件: telemetry.h
改变 uwac.h
36a37
> UWAC_API void UwacWindowSetAppId(UwacWindow* window, const char* app_id);
改变 wtsvc.h
3c3
< FREERDP_API void WTSVirtualChannelManagerGetFileDescriptor(HANDLE hServer, void** fds,
——

```

```

> FREERDP_API void WTSVirtualChannelManagerGetFileDescriptor(HANDLE hServer, void** fds,
8a9,11
> FREERDP_API void WTSVirtualChannelManagerSetDVCCreationCallback(HANDLE hServer,
>
psDVCCreationStatusCallback cb,
> void* userdata);
18a22,24
> FREERDP_API INT64 WTSCchannelGetOptions(freerdp_peer* client, UINT16 channel_id);
> FREERDP_API char** WTSGetAcceptedChannelNames(freerdp_peer* client, size_t* count);
> FREERDP_API UINT32 WTSCchannelGetIdByHandle(HANDLE hChannelHandle);

```

poppler-cpp-devel

增加头文件: poppler-destination.h

改变 poppler-document.h

```

2a3
> class destination;
51,52c52,54
< bool save(const std::string &filename) const;
< bool save_a_copy(const std::string &filename) const;
---
> std::map<std::string, destination> create_destination_map() const;
> bool save(const std::string &file_name) const;
> bool save_a_copy(const std::string &file_name) const;

```

增加头文件: poppler-font-private.h

改变 poppler-global.h

```

6a7,9
> public:
> noncopyable(const noncopyable &) = delete;
> const noncopyable& operator=(const noncopyable &) = delete;
13a17
> POPPLER_CPP_EXPORT bool set_data_dir(const std::string &new_data_dir);

```

改变 poppler-page.h

```

8,9c8,9
< text_box(text_box&&) = default;
< text_box& operator=(text_box&&) = default;
---
> text_box(text_box&&) noexcept;
> text_box& operator=(text_box&&) noexcept;
12a13
> int rotation() const;
14a16,19
> bool has_font_info() const;
> writing_mode_enum get_wmode(int i = 0) const;

```

```

> double get_font_size() const;
> std::string get_font_name(int i = 0) const;
33a39
> std::vector<text_box> text_list(int opt_flag) const;

```

jemalloc-devel

改变 jemalloc.h

```

6c6
< void JEMALLOC_NO_THROW *je_malloc(size_t size)
----
> void JEMALLOC_SYS_NO_THROW *je_malloc(size_t size)
9c9
< void JEMALLOC_NO_THROW *je_calloc(size_t num, size_t size)
----
> void JEMALLOC_SYS_NO_THROW *je_calloc(size_t num, size_t size)
11,12c11,13
< JEMALLOC_EXPORT int JEMALLOC_NO_THROW je_posix_memalign(void **memptr,
< size_t alignment, size_t size) JEMALLOC_CXX_THROW JEMALLOC_ATTR(nonnull(1));
----
> JEMALLOC_EXPORT int JEMALLOC_SYS_NO_THROW je_posix_memalign(
> void **memptr, size_t alignment, size_t size) JEMALLOC_CXX_THROW
> JEMALLOC_ATTR(nonnull(1));
14c15
< void JEMALLOC_NO_THROW *je_aligned_alloc(size_t alignment,
----
> void JEMALLOC_SYS_NO_THROW *je_aligned_alloc(size_t alignment,
18c19
< void JEMALLOC_NO_THROW *je_realloc(void *ptr, size_t size)
----
> void JEMALLOC_SYS_NO_THROW *je_realloc(void *ptr, size_t size)
20c21
< JEMALLOC_EXPORT void JEMALLOC_NO_THROW je_free(void *ptr)
----
> JEMALLOC_EXPORT void JEMALLOC_SYS_NO_THROW je_free(void *ptr)
47a49,50
> JEMALLOC_EXPORT size_t JEMALLOC_NO_THROW je_malloc_size(
> const void *ptr);
49c52
< void JEMALLOC_NO_THROW *je_memalign(size_t alignment, size_t size)
----
> void JEMALLOC_SYS_NO_THROW *je_memalign(size_t alignment, size_t size)

```


52c55

```
< void JEMALLOC_NOTHROW *je_valloc(size_t size) JEMALLOC_CXX_THROW
```

```
> void JEMALLOC_SYS_NOTHROW *je_valloc(size_t size) JEMALLOC_CXX_THROW
```

dovecot-devel

改变 auth-client-private.h

0a1

```
> extern struct event_category event_category_auth_client;
```

改变 auth.h

0a1,2

```
> ARRAY_DEFINE_TYPE(auth, struct auth *);
```

```
> extern ARRAY_TYPE(auth) auths;
```

改变 auth-master-connection.h

3,4d2

```
< void auth_master_log_error(struct auth_master_connection *conn,
```

```
< const char *fmt, ...) ATTR_FORMAT(2, 3);
```

改变 auth-request.h

5c5

```
< struct auth_request *auth_request_new_dummy(void);
```

```
> struct auth_request *auth_request_new_dummy(struct event *parent_event);
```

37a38,39

```
> void auth_request_set_username_forced(struct auth_request *request,
```

```
> const char *username);
```

40a43,50

```
> void auth_request_set_login_username_forced(struct auth_request *request,
```

```
> const char *username);
```

```
> void auth_request_set_realm(struct auth_request *request, const char *realm);
```

```
> void auth_request_set_auth_successful(struct auth_request *request);
```

```
> void auth_request_set_password_verified(struct auth_request *request);
```

```
> void auth_request_set_delayed_credentials(struct auth_request *request,
```

```
> const unsigned char *credentials,
```

```
> size_t size);
```

51c61,62

```
< void auth_request_init_userdb_reply(struct auth_request *request);
```

```
> void auth_request_init_userdb_reply(struct auth_request *request,
```

```
> bool add_default_fields);
```

104a116,117

```
> void auth_request_default_verify_plain_continue(struct auth_request *request,
```

```

>                                verify_plain_callback_t *callback);
110a124,126
> void auth_request_master_user_login_finish(struct auth_request *request);
> const char *auth_request_get_log_prefix_db(struct auth_request *auth_request);
> void auth_request_fields_init(struct auth_request *request);
117c133
< static inline struct event *authdb_event(struct auth_request *request)
-----
> static inline struct event *authdb_event(const struct auth_request *request)
增加头文件: auth-request-handler-private.h
改变 auth-worker-client.h
1a2
> struct auth_worker_command;
改变 base64.h
9a10,11
> size_t base64_encode_get_full_space(struct base64_encoder *enc,
>                                size_t dst_space);
改变 bits.h
19c19
< static inline uint64_t
-----
> static inline uint64_t ATTR_NO_SANITIZE_INTEGER
21c21
< static inline uint32_t
-----
> static inline uint32_t ATTR_NO_SANITIZE_INTEGER
23c23
< static inline uint64_t
-----
> static inline uint64_t ATTR_NO_SANITIZE_INTEGER
25c25
< static inline uint32_t
-----
> static inline uint32_t ATTR_NO_SANITIZE_INTEGER
31c31
< static inline unsigned int ATTR_CONST
-----
> static inline unsigned int ATTR_CONST ATTR_NO_SANITIZE_INTEGER
改变 buffer.h
4a5,6
> buffer_t *buffer_create_dynamic_max(pool_t pool, size_t init_size,
>                                size_t max_size);
改变 client-common.h
12c12,13

```

```

< void client_disconnect(struct client *client, const char *reason);
——
> void client_disconnect(struct client *client, const char *reason,
>
>         bool add_disconnected_prefix);
13a15
> void client_destroy_iostream_error(struct client *client);
26,28d27
< void client_log(struct client *client, const char *msg);
< void client_log_err(struct client *client, const char *msg);
< void client_log_warn(struct client *client, const char *msg);
48a48,50
> void client_common_proxy_failed(struct client *client,
>
>         enum login_proxy_failure_type type,
>
>         const char *reason, bool reconnecting);
60d61
< void client_proxy_failed(struct client *client, bool send_line);
63c64
< void client_destroy_oldest(void);
——
> bool client_destroy_oldest(bool kill, struct timeval *created_r);
改变 compression.h
2c2,3
< const struct compression_handler *compression_lookup_handler(const char *name);
——
> int compression_lookup_handler(const char *name,
>
>         const struct compression_handler **handler_r);
5,6c6,7
< const struct compression_handler *
< compression_lookup_handler_from_ext(const char *path);
——
> int compression_lookup_handler_from_ext(const char *path,
>
>         const struct compression_handler **handler_r);
增加头文件: cpu-limit.h
减少头文件: cydir-storage.h
减少头文件: cydir-sync.h
改变 data-stack.h
14a15
> size_t data_stack_get_used_size(void);
改变 dencrypt.h
7a8
> bool dencrypt_is_initialized(void);
改变 dict-private.h
0a1,2
> struct ioloop;

```

```

> struct dict_commit_callback_ctx;
12a15,16
> void dict_pre_api_callback(struct dict *dict);
> void dict_post_api_callback(struct dict *dict);
改变 event-filter.h
2a3
> struct event_filter *event_filter_create_fragment(pool_t pool);
8a10,12
> void event_filter_merge_with_context(struct event_filter *dest,
>                                     const struct event_filter *src,
>                                     void *new_context);
10,14c14,15
< bool event_filter_import(struct event_filter *filter, const char *str,
<                             const char **error_r);
< bool event_filter_import_unescaped(struct event_filter *filter,
<                                     const char *const *args,
<                                     const char **error_r);
——
> int event_filter_parse(const char *str, struct event_filter *filter,
>                         const char **error_r);
增加头文件: event-filter-parser.h
增加头文件: event-filter-private.h
改变 fts-api.h
6a7
> extern struct event_category event_category_fts;
增加头文件: fuzzer.h
改变 hash2.h
0a1,17
> hash2_create(unsigned int initial_size, unsigned int value_size,
>              hash2_key_callback_t *key_hash_cb,
>              hash2_cmp_callback_t *key_compare_cb, void *context) ATTR_NULL(5);
> void hash2_destroy(struct hash2_table **hash);
> void hash2_clear(struct hash2_table *hash);
> void *hash2_lookup(const struct hash2_table *hash, const void *key) ATTR_PURE;
> void *hash2_iterate(const struct hash2_table *hash,
>                     unsigned int key_hash, struct hash2_iter *iter);
> void *hash2_insert(struct hash2_table *hash, const void *key);
> void *hash2_insert_hash(struct hash2_table *hash, unsigned int key_hash);
> void hash2_remove(struct hash2_table *hash, const void *key);
> void hash2_remove_iter(struct hash2_table *hash, struct hash2_iter *iter);
> unsigned int hash2_count(const struct hash2_table *hash) ATTR_PURE;
> static inline bool hash2_strcmp(const void *a, const void *b, void *ctx ATTR_UNUSED)
>
> static inline unsigned int hash2_str_hash(const void *key)

```

```

>
改变 hmac-cram-md5.h
1c1
< void hmac_md5_get_cram_context(struct hmac_context *ctx,
-----
> void hmac_md5_get_cram_context(struct orig_hmac_context *ctx,
3c3
< void hmac_md5_set_cram_context(struct hmac_context *ctx,
-----
> void hmac_md5_set_cram_context(struct orig_hmac_context *ctx,
改变 hmac.h
1c1
< void hmac_init(struct hmac_context *ctx, const unsigned char *key,
-----
> void openssl_hmac_init(struct openssl_hmac_context *ctx, const unsigned char *key,
3c3
< void hmac_final(struct hmac_context *ctx, unsigned char *digest);
-----
> void openssl_hmac_final(struct openssl_hmac_context *ctx, unsigned char *digest);
6c6,11
< buffer_t *t_hmac_data(const struct hash_method *meth,
-----
> void orig_hmac_init(struct orig_hmac_context *ctx, const unsigned char *key,
>         size_t key_len, const struct hash_method *meth);
> void orig_hmac_final(struct orig_hmac_context *ctx, unsigned char *digest);
> static inline void
>
> buffer_t *openssl_t_hmac_data(const struct hash_method *meth,
9c14
< buffer_t *t_hmac_buffer(const struct hash_method *meth,
-----
> buffer_t *openssl_t_hmac_buffer(const struct hash_method *meth,
12c17
< buffer_t *t_hmac_str(const struct hash_method *meth,
-----
> buffer_t *openssl_t_hmac_str(const struct hash_method *meth,
15c20
< void hmac_hkdf(const struct hash_method *method,
-----
> void openssl_hmac_hkdf(const struct hash_method *method,
改变 http-client-private.h
8,21c8,14
< ARRAY_DEFINE_TYPE(http_client_request,
<     struct http_client_request *);

```

```

< ARRAY_DEFINE_TYPE(http_client_connection,
<     struct http_client_connection *);
< ARRAY_DEFINE_TYPE(http_client_peer,
<     struct http_client_peer *);
< ARRAY_DEFINE_TYPE(http_client_peer_shared,
<     struct http_client_peer_shared *);
< ARRAY_DEFINE_TYPE(http_client_peer_pool,
<     struct http_client_peer_pool *);
< ARRAY_DEFINE_TYPE(http_client_queue,
<     struct http_client_queue *);
< ARRAY_DEFINE_TYPE(http_client_host,
<     struct http_client_host_shared *);
____
> ARRAY_DEFINE_TYPE(http_client_request, struct http_client_request *);
> ARRAY_DEFINE_TYPE(http_client_connection, struct http_client_connection *);
> ARRAY_DEFINE_TYPE(http_client_peer, struct http_client_peer *);
> ARRAY_DEFINE_TYPE(http_client_peer_shared, struct http_client_peer_shared *);
> ARRAY_DEFINE_TYPE(http_client_peer_pool, struct http_client_peer_pool *);
> ARRAY_DEFINE_TYPE(http_client_queue, struct http_client_queue *);
> ARRAY_DEFINE_TYPE(http_client_host, struct http_client_host_shared *);
23,26c16,19
<     const struct http_client_peer_addr *,
<     struct http_client_peer_shared *);
< HASH_TABLE_DEFINE_TYPE(http_client_host_shared,
<     const char *, struct http_client_host_shared *);
____
>         const struct http_client_peer_addr *,
>         struct http_client_peer_shared *);
> HASH_TABLE_DEFINE_TYPE(http_client_host_shared, const char *,
>         struct http_client_host_shared *);
38c31,32
<         net_ip2addr(&addr->a.tcp.ip), addr->a.tcp.port);
____
>         net_ip2addr(&addr->a.tcp.ip),
>         addr->a.tcp.port);
49,50c43
< const char *
< http_client_request_label(struct http_client_request *req);
____
> const char *http_client_request_label(struct http_client_request *req);
55c48
<     struct http_client_peer_addr *addr);
____
>         struct http_client_peer_addr *addr);

```

改变 http-server.h

4a5

> struct http_server_resource;

8c9,11

< unsigned int status, const char *reason);

> unsigned int status, const char *reason);

> void http_server_response_ref(struct http_server_response *resp);

> bool http_server_response_unref(struct http_server_response **_resp);

10c13

< const char *key, const char *value);

> const char *key, const char *value);

16c19

< time_t date);

> time_t date);

18c21

< struct istream *input);

> struct istream *input);

20c23,24

< const unsigned char *data, size_t size);

> const unsigned char *data,

> size_t size);

23c27

< bool blocking);

> size_t max_buffer_size, bool blocking);

25c29

< int *status_r, const char **reason_r);

> int *status_r, const char **reason_r);

27,31c31,34

< void http_server_response_add_auth(

< struct http_server_response *resp,

< const struct http_auth_challenge *chlng);

< void http_server_response_add_auth_basic(

< struct http_server_response *resp, const char *realm);

> void http_server_response_add_auth(struct http_server_response *resp,

> const struct http_auth_challenge *chlng);

> void http_server_response_add_auth_basic(struct http_server_response *resp,

```

>                                const char *realm);
35c38,39
<    http_server_tunnel_callback_t callback, void *context);
-----
>                                http_server_tunnel_callback_t callback,
>                                void *context);
37c41
<    const unsigned char *data, size_t size);
-----
>                                const unsigned char *data, size_t size);
45c49
<    bool close);
-----
>                                bool close);
54c58
<    bool blocking);
-----
>                                bool blocking);
56,57c60,63
<    struct ostream *output, uoff_t max_size,
<    void (*callback)(void *), void *context);
-----
>                                struct ostream *output,
>                                uoff_t max_size,
>                                void (*callback)(void *),
>                                void *context);
59,60c65,67
<    buffer_t *buffer, uoff_t max_size,
<    void (*callback)(void *), void *context);
-----
>                                buffer_t *buffer, uoff_t max_size,
>                                void (*callback)(void *),
>                                void *context);
62c69,70
<    void (*callback)(void *context), void *context);
-----
>                                void (*callback)(void *context),
>                                void *context);
64c72
<    struct http_auth_credentials *credentials);
-----
>                                struct http_auth_credentials *credentials);
66c74
<    unsigned int status, const char *reason);

```



```

-----
> unsigned int status, const char *reason);
68c76
< unsigned int status, const char *reason);
-----
> unsigned int status, const char *reason);
70,71c78,79
< unsigned int status, const char *reason, const char *format, ...)
< ATTR_FORMAT(4, 5);
-----
> unsigned int status, const char *reason,
> const char *format, ...) ATTR_FORMAT(4, 5);
73,74c81,83
< const char *reason, const struct http_auth_challenge *chlng)
< ATTR_NULL(2);
-----
> const char *reason,
> const struct http_auth_challenge *chlng)
> ATTR_NULL(2);
76,77c85,88
< const char *reason, const char *realm)
< ATTR_NULL(2);
-----
> const char *reason, const char *realm)
> ATTR_NULL(2);
> void http_server_request_fail_bad_method(struct http_server_request *req,
> const char *allow);
改变 http-server-private.h
0a1
> struct http_server_ostream;
4c5,24
< void http_server_response_free(struct http_server_response *resp);
-----
> http_server_ostream_create(struct http_server_response *resp,
> size_t max_buffer_size, bool blocking);
> bool http_server_ostream_get_size(struct http_server_ostream *hsostream,
> uoff_t *size_r);
> int http_server_ostream_continue(struct http_server_ostream *hsostream);
> void http_server_ostream_output_available(
> struct http_server_ostream *hsostream);
> void http_server_ostream_response_finished(
> struct http_server_ostream *hsostream);
> void http_server_ostream_response_destroyed(
> struct http_server_ostream *hsostream);

```

```

> struct ostream *
> http_server_ostream_get_output(struct http_server_ostream *hsostream);
> void http_server_ostream_set_error(struct http_server_ostream *hsostream,
>
>         int stream_errno, const char *stream_error);
> void http_server_response_request_free(struct http_server_response *resp);
> void http_server_response_request_destroy(struct http_server_response *resp);
> void http_server_response_request_abort(struct http_server_response *resp,
>
>         const char *reason);
> void http_server_response_request_finished(struct http_server_response *resp);
7,11c27
< static inline const char *
<
<
<     return t_strdup_printf("[Req%u: %s %s]", req->id,
<         req->req.method, req->req.target_raw);
< }
____
> int http_server_response_finish_payload_out(struct http_server_response *resp);
15a32,33
> const char *http_server_request_label(struct http_server_request *req);
> void http_server_request_update_event(struct http_server_request *req);
19c37,39
<     const char *reason) ATTR_NULL(2);
____
>
>         const char *reason) ATTR_NULL(2);
> void http_server_request_immune_ref(struct http_server_request *req);
> void http_server_request_immune_unref(struct http_server_request **_req);
20a41
> void http_server_request_callback(struct http_server_request *req);
29,31c50
<     struct http_server_payload_handler *handler);
< static inline const char *
<
____
>     struct http_server_payload_handler *handler, struct ioloop *ioloop);
38,39c57,61
< void http_server_connection_switch_ioloop(struct http_server_connection *conn);
< void http_server_connection_handle_output_error(
____
> void http_server_connection_input_set_pending(
>     struct http_server_connection *conn);
> void http_server_connection_input_halt(struct http_server_connection *conn);
> void http_server_connection_input_resume(struct http_server_connection *conn);
> void http_server_connection_start_idle_timeout(
41c63

```

```

< void http_server_connection_trigger_responses(
____
> void http_server_connection_reset_idle_timeout(
42a65,71
> void http_server_connection_stop_idle_timeout(
>     struct http_server_connection *conn);
> void http_server_connection_handle_output_error(
>     struct http_server_connection *conn);
> void http_server_connection_output_trigger(struct http_server_connection *conn);
> void http_server_connection_output_halt(struct http_server_connection *conn);
> void http_server_connection_output_resume(struct http_server_connection *conn);
46,48c75,76
<     http_server_tunnel_callback_t callback, void *context);
< int http_server_connection_discard_payload(
<     struct http_server_connection *conn);
____
>
>         http_server_tunnel_callback_t callback,
>         void *context);
50a79,82
> int http_server_resource_find(struct http_server *server, const char *path,
>
>         struct http_server_resource **res_r,
>         const char **sub_path_r) ATTR_NULL(2);
> bool http_server_resource_callback(struct http_server_request *req);
改变 http-url.h
14a15,16
> void http_url_init_authority_from(struct http_url *dest,
>
>         const struct http_url *src);
改变 imap-bodystructure.h
4,5c4,6
< void imap_bodystructure_write(const struct message_part *part,
<
<         string_t *dest, bool extended);
____
> int imap_bodystructure_write(const struct message_part *part,
>
>         string_t *dest, bool extended,
>         const char **error_r);
改变 imapc-list.h
5c5,6
< const char *imapc_list_to_remote(struct imapc_mailbox_list *list, const char *name);
____
> const char *imapc_list_storage_to_remote_name(struct imapc_mailbox_list *list,
>
>         const char *storage_name);
改变 imap-common.h
7a8
> void imap_refresh_proctitle_delayed(void);

```

改变 imap-search-args.h

12a13,14

```
> void imap_search_anysubset_to_uidset(struct client_command_context *cmd,  
> struct mail_search_args *args);
```

改变 imap-utf7.h

2a3,4

```
> int imap_escaped_utf8_to_utf7(const char *src, char escape_char, string_t *dest);  
> int imap_escaped_utf8_hex_to_char(const char *str, unsigned char *chr_r);  
3a6,7  
> void imap_utf7_to_utf8_escaped(const char *src, const char *escape_chars,  
> string_t *dest);
```

改变 iostream-openssl.h

15a16

```
> const char *set_name,  
21,22c22,23  
< bool openssl_iostream_bio_sync(struct ssl_iostream *ssl_io,  
< enum openssl_iostream_sync_type type);  
——  
> int openssl_iostream_bio_sync(struct ssl_iostream *ssl_io,  
> enum openssl_iostream_sync_type type);
```

改变 iostream-pump.h

2a3

```
> struct ioloop;
```

改变 istream.h

5,7d4

```
< struct istream *i_stream_create_mmap(int fd, size_t block_size,  
< uoff_t start_offset, uoff_t v_size,  
< bool autoclose_fd);
```

改变 istream-try.h

1d0

```
< struct istream *istream_try_create(struct istream *const input[]);
```

改变 istream-zlib.h

1,5c1,6

```
< struct istream *i_stream_create_gz(struct istream *input, bool log_errors);  
< struct istream *i_stream_create_deflate(struct istream *input, bool log_errors);  
< struct istream *i_stream_create_bz2(struct istream *input, bool log_errors);  
< struct istream *i_stream_create_lzma(struct istream *input, bool log_errors);  
< struct istream *i_stream_create_lz4(struct istream *input, bool log_errors);  
——  
> struct istream *i_stream_create_gz(struct istream *input);  
> struct istream *i_stream_create_deflate(struct istream *input);  
> struct istream *i_stream_create_bz2(struct istream *input);  
> struct istream *i_stream_create_lzma(struct istream *input);  
> struct istream *i_stream_create_lz4(struct istream *input);
```

```

> struct istream *i_stream_create_zstd(struct istream *input);
改变 lib.h
16a17,20
> static inline unsigned short i_rand_ushort(void)
>
> static inline unsigned char i_rand_uchar(void)
>
改变 lib-signals.h
8a9
> void lib_signals_clear_handlers_and_ignore(int signo);
11a13,15
> void lib_signals_set_expected(int signo, bool expected,
>                               signal_handler_t *handler, void *context);
>     ATTR_NULL(4);
13c17
<     signal_handler_t *handler, void *context);
---
>                               signal_handler_t *handler, void *context);
改变 login-common.h
2c2
< extern const struct login_binary *login_binary;
---
> extern struct login_binary *login_binary;
20c20
< int login_binary_run(const struct login_binary *binary,
---
> int login_binary_run(struct login_binary *binary,
改变 login-proxy.h
3c3
< int login_proxy_new(struct client *client,
---
> int login_proxy_new(struct client *client, struct event *event,
5c5,6
<
    proxy_callback_t *callback);
---
>
    login_proxy_input_callback_t *input_callback,
>
    login_proxy_failure_callback_t *failure_callback);
6a8,9
> bool login_proxy_failed(struct login_proxy *proxy, struct event *event,
>
    enum login_proxy_failure_type type, const char *reason);
12a16,19
> void login_proxy_append_success_log_info(struct login_proxy *proxy,
>
    string_t *str);
> struct event *login_proxy_get_event(struct login_proxy *proxy);

```

> const char *login_proxy_get_source_host(const struct login_proxy *proxy) ATTR_PURE;

改变 mailbox-list-private.h

16,17d15

< mailbox_list_escape_name(struct mailbox_list *list, const char *vname);

< const char *

21,22d18

< const char *

< mailbox_list_unescape_name(struct mailbox_list *list, const char *src);

改变 mailbox-log.h

1,13d0

< struct mailbox_log *mailbox_log_alloc(const char *path);

< void mailbox_log_free(struct mailbox_log **log);

< void mailbox_log_set_permissions(struct mailbox_log *log, mode_t mode,

< gid_t gid, const char *gid_origin);

< void mailbox_log_record_set_timestamp(struct mailbox_log_record *rec,

< time_t stamp);

< time_t mailbox_log_record_get_timestamp(const struct mailbox_log_record *rec);

< int mailbox_log_append(struct mailbox_log *log,

< const struct mailbox_log_record *rec);

< struct mailbox_log_iter *mailbox_log_iter_init(struct mailbox_log *log);

< const struct mailbox_log_record *

< mailbox_log_iter_next(struct mailbox_log_iter *iter);

< int mailbox_log_iter_deinit(struct mailbox_log_iter **iter);

改变 mail-cache.h

4d3

< struct mail_cache_compress_lock;

改变 mail-cache-private.h

2,3c2,3

< int mail_cache_try_lock(struct mail_cache *cache);

< int mail_cache_unlock(struct mail_cache *cache);

> int mail_cache_flush_and_unlock(struct mail_cache *cache);

> void mail_cache_unlock(struct mail_cache *cache);

28a29

> bool mail_cache_transactions_have_changes(struct mail_cache *cache);

30a32

> int mail_cache_map_all(struct mail_cache *cache);

32a35

> int mail_cache_sync_reset_id(struct mail_cache *cache);

35,39c38,48

< int mail_cache_expunge_handler(struct mail_index_sync_map_ctx *sync_ctx,

< uint32_t seq, const void *data,

< void **sync_context, void *context);

< void mail_cache_set_syscall_error(struct mail_cache *cache,

```

<
const char *function);
-----
> const char *mail_cache_decision_to_string(enum mail_cache_decision_type dec);
> mail_cache_decision_changed_event(struct mail_cache *cache, struct event *event,
>
unsigned int field);
> struct mail_cache_purge_drop_ctx {
>     struct mail_cache *cache;
>     time_t max_yes_downgrade_time;
>     time_t max_temp_drop_time;
> };
> void mail_cache_purge_drop_init(struct mail_cache *cache,
>
const struct mail_index_header *hdr,
>
struct mail_cache_purge_drop_ctx *ctx_r);

```

改变 mail-deliver.h

30c30,31

```

<
struct mail_storage **storage_r);
-----
>
enum mail_deliver_error *error_code_r,
>
const char **error_r);

```

改变 mail-index-private.h

5c5

```

< extern struct event_category event_category_index;
-----
> extern struct event_category event_category_mail_index;

```

20c20

```

< int mail_index_reopen_if_changed(struct mail_index *index,
-----
> int mail_index_reopen_if_changed(struct mail_index *index, bool *reopened_r,

```

22c22,23

```

< void mail_index_write(struct mail_index *index, bool want_rotate);
-----
> void mail_index_write(struct mail_index *index, bool want_rotate,
>
const char *reason);

```

改变 mail-index-sync-private.h

7,9c7,11

```

< int mail_index_sync_map(struct mail_index_map **map,
<
enum mail_index_sync_handler_type type, bool force,
<
const char *sync_reason);
-----
> bool mail_index_sync_map_want_index_reopen(struct mail_index_map *map,
>
enum mail_index_sync_handler_type type);
> int mail_index_sync_map(struct mail_index_map **_map,
>
enum mail_index_sync_handler_type type,
>
const char **reason_r);

```

改变 mail-namespace.h

1a2,3

```
> static inline bool mail_namespace_is_removable(const struct mail_namespace *ns)
```

```
>
```

55a58,59

```
> static inline bool
```

```
>
```

改变 mail-search.h

3c3

```
< struct mailbox *box, bool change_uidsets,
```

```
----
```

```
> struct mailbox *box, bool change_sets,
```

7,9c7

```
< struct mail_search_arg *arg,
```

```
< bool change_uidsets,
```

```
< const ARRAY_TYPE(seq_range) *search_saved_uidset);
```

```
----
```

```
> struct mail_search_arg *arg);
```

改变 mail-storage-private.h

38a39,40

```
> void mail_opened_event(struct mail *mail);
```

```
> void mail_expunge_requested_event(struct mail *mail);
```

改变 mbox-storage.h

4a5,10

```
> void mbox_istream_set_syscall_error(struct mbox_mailbox *mbox,
```

```
> struct istream *input,
```

```
> const char *function);
```

```
> void mbox_ostream_set_syscall_error(struct mbox_mailbox *mbox,
```

```
> struct ostream *output,
```

```
> const char *function);
```

增加头文件: mech-digest-md5-private.h

增加头文件: mech-otp-common.h

减少头文件: mech-otp-skey-common.h

改变 message-header-encode.h

2c2

```
< void message_header_encode_data(const unsigned char *input, unsigned int len,
```

```
----
```

```
> void message_header_encode_data(const unsigned char *input, size_t len,
```

4,7c4,7

```
< void message_header_encode_q(const unsigned char *input, unsigned int len,
```

```
< string_t *output, unsigned int first_line_len);
```

```
< void message_header_encode_b(const unsigned char *input, unsigned int len,
```

```
< string_t *output, unsigned int first_line_len);
```

```
----
```



```

> void message_header_encode_q(const unsigned char *input, size_t len,
>                               string_t *output, size_t first_line_len);
> void message_header_encode_b(const unsigned char *input, size_t len,
>                               string_t *output, size_t first_line_len);
改变 oauth2.h
0a1
> struct dict;
1a3
> struct oauth2_validation_key_cache;
改变 ostream.h
4a5,6
> struct ostream *o_stream_create_file(const char *path, uoff_t offset, mode_t mode,
>                                       enum ostream_create_file_flags flags);
14a17
> const char *o_stream_get_disconnect_reason(struct ostream *stream);
增加头文件: ostream-wrapper.h
改变 ostream-zlib.h
4d3
< struct ostream *o_stream_create_lzma(struct ostream *output, int level);
5a5,17
> struct ostream *o_stream_create_zstd(struct ostream *output, int level);
> int compression_get_min_level_gz(void);
> int compression_get_default_level_gz(void);
> int compression_get_max_level_gz(void);
> int compression_get_min_level_bz2(void);
> int compression_get_default_level_bz2(void);
> int compression_get_max_level_bz2(void);
> int compression_get_min_level_lz4(void);
> int compression_get_default_level_lz4(void);
> int compression_get_max_level_lz4(void);
> int compression_get_min_level_zstd(void);
> int compression_get_default_level_zstd(void);
> int compression_get_max_level_zstd(void);
改变 password-scheme.h
35d34
< void password_generate_rpa(const char *pw, unsigned char result[]);
改变 sdbox-file.h
7,8c7
< int sdbox_file_assign_uid(struct sdbox_file *file, uint32_t uid,
<                               bool ignore_if_exists);
——
> int sdbox_file_assign_uid(struct sdbox_file *file, uint32_t uid);
改变 seq-range-array.h
2c2

```

```

< static inline uint32_t ATTR_PURE seq_range_length(struct seq_range *range)
——
> static inline uint32_t ATTR_PURE seq_range_length(const struct seq_range *range)
改变 sha2.h
6a7,12
> void sha384_init(struct sha384_ctx *ctx);
> void sha384_loop(struct sha384_ctx *ctx, const void *data, size_t len);
> void sha384_result(struct sha384_ctx *ctx,
>         unsigned char digest[STATIC_ARRAY_SHA384_RESULTLEN]);
> void sha384_get_digest(const void *data, size_t size,
>         unsigned char digest[STATIC_ARRAY_SHA384_RESULTLEN]);
13a20
> extern const struct hash_method hash_method_sha384;
改变 sieve-address.h
1c1
< static inline int sieve_address_list_next_item
——
> static inline int
3c3
< static inline void sieve_address_list_reset
——
> static inline void
5c5
< static inline int sieve_address_list_get_length
——
> static inline int
7c7
< static inline void sieve_address_list_set_trace
——
> static inline void
改变 sieve-binary.h
4c4,12
< void sieve_binary_unref(struct sieve_binary **sbin);
——
> void sieve_binary_unref(struct sieve_binary **_sbin);
> void sieve_binary_close(struct sieve_binary **_sbin);
> void sieve_binary_get_resource_usage(struct sieve_binary *sbin,
>         struct sieve_resource_usage *rusage_r);
> bool sieve_binary_record_resource_usage(
>         struct sieve_binary *sbin, const struct sieve_resource_usage *rusage)
>         ATTR_NULL(1);
> void sieve_binary_set_resource_usage(struct sieve_binary *sbin,
>         const struct sieve_resource_usage *rusage);
23a32,34

```

```

> int sieve_binary_check_executable(struct sieve_binary *sbin,
>
> enum sieve_error *error_r,
> const char **client_error_r);
改变 sieve-binary-private.h
25a26,28
> bool sieve_binary_check_resource_usage(struct sieve_binary *sbin);
> int sieve_binary_file_update_resource_usage(struct sieve_binary *sbin,
>
> enum sieve_error *error_r);
改变 sieve-common.h
65a66
> extern struct event_category event_category_sieve;
82a84,85
> unsigned int max_cpu_time_secs;
> unsigned int resource_usage_timeout_secs;
改变 sieve-interpreter.h
56a57,59
> int sieve_interpreter_program_jump_to(struct sieve_interpreter *interp,
>
> sieve_size_t jmp_target,
>
> bool break_loops);
改变 sieve-result.h
58c58
< int sieve_result_execute(struct sieve_result *result, bool *keep,
——
> int sieve_result_execute(struct sieve_result *result, bool last, bool *keep,
59a60,61
> void sieve_result_finish(struct sieve_result *result,
>
> struct sieve_error_handler *ehandler, bool success);
改变 sieve-settings.h
1c1
< static inline const char *sieve_setting_get
——
> static inline const char *
3,20c3,16
< bool sieve_setting_get_uint_value
< (struct sieve_instance *svinst, const char *setting,
< unsigned long long int *value_r);
< bool sieve_setting_get_int_value
< (struct sieve_instance *svinst, const char *setting,
< long long int *value_r);
< bool sieve_setting_get_size_value
< (struct sieve_instance *svinst, const char *setting,
< size_t *value_r);
< bool sieve_setting_get_bool_value
< (struct sieve_instance *svinst, const char *setting,

```

```

<         bool *value_r);
< bool sieve_setting_get_duration_value
<     (struct sieve_instance *svinst, const char *setting,
<         sieve_number_t *value_r);
< void sieve_settings_load
<     (struct sieve_instance *svinst);
< static inline const char *sieve_environment_get_homedir
——
> bool sieve_setting_get_uint_value(struct sieve_instance *svinst,
>         const char *setting,
>         unsigned long long int *value_r);
> bool sieve_setting_get_int_value(struct sieve_instance *svinst,
>         const char *setting, long long int *value_r);
> bool sieve_setting_get_size_value(struct sieve_instance *svinst,
>         const char *setting, size_t *value_r);
> bool sieve_setting_get_bool_value(struct sieve_instance *svinst,
>         const char *setting, bool *value_r);
> bool sieve_setting_get_duration_value(struct sieve_instance *svinst,
>         const char *setting,
>         sieve_number_t *value_r);
> void sieve_settings_load(struct sieve_instance *svinst);
> static inline const char *
改变 smtp-client-connection.h
2a3
> struct smtp_client_capability_extra;
改变 smtp-params.h
4a5,13
> void smtp_params_copy(pool_t pool, ARRAY_TYPE(smtp_param) *dst,
>         const ARRAY_TYPE(smtp_param) *src) ATTR_NULL(3);
> void smtp_params_add_one(ARRAY_TYPE(smtp_param) *params, pool_t pool,
>         const char *keyword, const char *value);
> void smtp_params_add_encoded(ARRAY_TYPE(smtp_param) *params, pool_t pool,
>         const char *keyword, const unsigned char *value,
>         size_t value_len);
> bool smtp_params_drop_one(ARRAY_TYPE(smtp_param) *params, const char *keyword,
>         const char **value_r);
5a15,22
> const struct smtp_param *
> smtp_params_get_param(const ARRAY_TYPE(smtp_param) *params,
>         const char *keyword);
> int smtp_params_decode_param(const ARRAY_TYPE(smtp_param) *params,
>         const char *keyword, string_t **value_r,
>         bool allow_nul, const char **error_r);
> bool smtp_params_equal(const ARRAY_TYPE(smtp_param) *params1,

```

```

> const ARRAY_TYPE(smtp_param) *params2);
14c31
< const struct smtp_params_mail *src) ATTR_NULL(3);
-----
> const struct smtp_params_mail *src);
17a35,38
> void smtp_params_mail_encode_extra(struct smtp_params_mail *params, pool_t pool,
> const char *keyword,
> const unsigned char *value,
> size_t value_len);
22c43,44
< const struct smtp_params_mail *params);
-----
> const char *const *extra_params,
> const struct smtp_params_mail *params) ATTR_NULL(3);
25a48,50
> int smtp_params_mail_decode_extra(const struct smtp_params_mail *params,
> const char *keyword, string_t **value_r,
> bool allow_nul, const char **error_r);
39a65,68
> void smtp_params_rcpt_encode_extra(struct smtp_params_rcpt *params, pool_t pool,
> const char *keyword,
> const unsigned char *value,
> size_t value_len);
46c75,76
< const struct smtp_params_rcpt *params);
-----
> const char *const *extra_params,
> const struct smtp_params_rcpt *params) ATTR_NULL(3);
50,51c80,84
< bool smtp_params_rcpt_equals(const struct smtp_params_rcpt *params1,
< const struct smtp_params_rcpt *params2);
-----
> int smtp_params_rcpt_decode_extra(const struct smtp_params_rcpt *params,
> const char *keyword, string_t **value_r,
> bool allow_nul, const char **error_r);
> bool smtp_params_rcpt_equal(const struct smtp_params_rcpt *params1,
> const struct smtp_params_rcpt *params2);
改变 smtp-server.h
104c104,105
< smtp_server_connection_get_state(struct smtp_server_connection *conn);
-----
> smtp_server_connection_get_state(struct smtp_server_connection *conn,
> const char **args_r) ATTR_NULL(2);

```

改变 smtp-server-private.h

41,44c41,42

```
< void smtp_server_cmd_ehlo(struct smtp_server_cmd_ctx *cmd,  
<     const char *params);
```

```
< void smtp_server_cmd_helo(struct smtp_server_cmd_ctx *cmd,  
<     const char *params);
```

```
> void smtp_server_cmd_ehlo(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_helo(struct smtp_server_cmd_ctx *cmd, const char *params);
```

46c44

```
<     const char *params);
```

```
>                 const char *params);
```

48,66c46,55

```
<     const char *params);
```

```
< void smtp_server_cmd_auth(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_mail(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_rcpt(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_data(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_bdat(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_rset(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_noop(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_vrfy(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
< void smtp_server_cmd_quit(struct smtp_server_cmd_ctx *cmd,
```

```
<     const char *params);
```

```
>                 const char *params);
```

```
> void smtp_server_cmd_auth(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_mail(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_rcpt(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_data(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_bdat(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_rset(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_noop(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_vrfy(struct smtp_server_cmd_ctx *cmd, const char *params);
```

```
> void smtp_server_cmd_quit(struct smtp_server_cmd_ctx *cmd, const char *params);
```

68c57

```
<    const char *format, ...) ATTR_FORMAT(2, 3);
```

```
>                                const char *format, ...) ATTR_FORMAT(2, 3);
```

86c75

```
<    const char *fmt, ...) ATTR_FORMAT(2, 3);
```

```
>                                const char *fmt, ...) ATTR_FORMAT(2, 3);
```

96c85,86

```
<    enum smtp_server_state state);
```

```
>                                enum smtp_server_state state,
```

```
>                                const char *args) ATTR_NULL(3);
```

改变 smtp-syntax.h

2,3c2,3

```
< int smtp_string_parse(const char *string,
```

```
<    const char **value_r, const char **error_r);
```

```
> int smtp_string_parse(const char *string, const char **value_r,
```

```
>                                const char **error_r);
```

5,8c5,9

```
< int smtp_xtext_parse(const char *xtext,
```

```
<    const char **value_r, const char **error_r);
```

```
< void smtp_xtext_encode(string_t *out,
```

```
<    const unsigned char *data, size_t size);
```

```
> int smtp_xtext_decode(string_t *out, const char *xtext, bool allow_nul,
```

```
>                                const char **error_r);
```

```
> int smtp_xtext_parse(const char *xtext, const char **value_r,
```

```
>                                const char **error_r);
```

```
> void smtp_xtext_encode(string_t *out, const unsigned char *data, size_t size);
```

11,12c12,13

```
< int smtp_helo_domain_parse(const char *helo,
```

```
<    bool allow_literal, const char **domain_r);
```

```
> int smtp_helo_domain_parse(const char *helo, bool allow_literal,
```

```
>                                const char **domain_r);
```

17,19c18,19

```
< int smtp_ehlo_line_parse(const char *ehlo_line,
```

```
<    const char **key_r, const char *const **params_r,
```

```
<    const char **error_r);
```

```
> int smtp_ehlo_line_parse(const char *ehlo_line, const char **key_r,
```

```
>                                const char *const **params_r, const char **error_r);
```

改变 stats-dist.h

11c11

```
< uint64_t stats_dist_get_median(const struct stats_dist *stats);
```

```
> uint64_t stats_dist_get_median(struct stats_dist *stats);
```

13,14c13,14

```
< uint64_t stats_dist_get_percentile(const struct stats_dist *stats, double fraction);
```

```
< static inline uint64_t stats_dist_get_95th(const struct stats_dist *stats)
```

```
> uint64_t stats_dist_get_percentile(struct stats_dist *stats, double fraction);
```

```
> static inline uint64_t stats_dist_get_95th(struct stats_dist *stats)
```

改变 strfuncs.h

50a51

```
> bool str_equals_timing_almost_safe(const char *s1, const char *s2);
```

改变 str.h

15,16d14

```
< static inline void ATTR_DEPRECATED("Use str_append_max() or str_append_data() instead")
```

```
<
```

改变 test-common.h

13,14c13,22

```
< void test_assert_failed_strcmp(const char *code, const char *file, unsigned int line,
```

```
<         const char *src, const char *dst);
```

```
> void test_assert_failed_strcmp_idx(const char *code, const char *file, unsigned int line,
```

```
>         const char *src, const char *dst, long long i);
```

```
> void test_assert_failed_cmp_intmax_idx(const char *code, const char *file,
```

```
>         unsigned int line,
```

```
>         intmax_t src, intmax_t dst,
```

```
>         const char *op, long long i);
```

```
> void test_assert_failed_ucmp_intmax_idx(const char *code, const char *file,
```

```
>         unsigned int line,
```

```
>         uintmax_t src, uintmax_t dst,
```

```
>         const char *op, long long i);
```

31a40,41

```
> void test_fatal_set_callback(test_fatal_callback_t *callback, void *context);
```

```
> void test_forked_end(void);
```

增加头文件: test-mail-storage-common.h

增加头文件: test-subprocess.h

改变 time-util.h

23a24,28

```
> static inline void timeval_add(struct timeval *tv, const struct timeval *val)
```

```
>
```

```
> }
```

```
> static inline time_t timeval_round(struct timeval *tv)
```



```
>
27a33,34
> int str_to_timeval(const char *str, struct timeval *tv_r)
>     ATTR_WARN_UNUSED_RESULT;
减少头文件: userdb-vpopmail.h
```

gnutls-devel

```
改变 x509.h
711a712,716
> int gnutls_x509_privkey_export_sm2_raw(gnutls_x509_privkey_t key,
>
>                                     gnutls_ecc_curve_t *curve,
>                                     gnutls_datum_t *x,
>                                     gnutls_datum_t *y,
>                                     gnutls_datum_t *k);
```

libmtp-devel

```
改变 libmtp.h
23a24
> LIBMTP_mtpdevice_t *LIBMTP_Get_Device(int);
179a181,182
> int LIBMTP_Move_Object(LIBMTP_mtpdevice_t *, uint32_t, uint32_t, uint32_t);
> int LIBMTP_Copy_Object(LIBMTP_mtpdevice_t *, uint32_t, uint32_t, uint32_t);
```

samba-devel

```
改变 auth.h
2d1
< }}
改变 ndr_krb5pac.h
18,20d17
< void ndr_print_PAC_ATTRIBUTE_INFO_FLAGS(struct ndr_print *ndr, const char *name, uint32_t r);
< void ndr_print_PAC_ATTRIBUTES_INFO(struct ndr_print *ndr, const char *name, const struct
PAC_ATTRIBUTES_INFO *r);
< void ndr_print_PAC_REQUESTER_SID(struct ndr_print *ndr, const char *name, const struct
PAC_REQUESTER_SID *r);
```

gdm-devel

改变 gdm-client.h

2c2

```
< GType          gdm_client_get_type          (void);
```

```
> G_DECLARE_FINAL_TYPE (GdmClient, gdm_client, GDM, CLIENT, GObject)
```

60d59

```
< G_DEFINE_AUTOPTR_CLEANUP_FUNC (GdmClient, g_object_unref)
```

armadillo-devel

变动很大，略

jsoncpp-devel

改变 allocator.h

14,15c14,15

```
< void deallocate(volatile pointer p, size_type n) {
```

```
<     std::memset(p, 0, n * sizeof(T));
```

```
> void deallocate(pointer p, size_type n) {
```

```
>     memset_s(p, n * sizeof(T), 0, n * sizeof(T));
```

改变 value.h

64a65

```
> Value(std::nullptr_t ptr) = delete;
```

poppler-devel

变动很大，略

cryptsetup-devel

改变 libcryptsetup.h

20a21

```
> void crypt_logf(struct crypt_device *cd, int level, const char *format, ...);
```

212a214

```
> int crypt_dump_json(struct crypt_device *cd, const char **json, uint32_t flags);
```

221a224

```

> int crypt_header_is_detached(struct crypt_device *cd);
286a290
> int crypt_token_max(const char *type);
309a314,315
> const char *crypt_token_external_path(void);
> void crypt_token_external_disable(void);
314a321,328
> int crypt_activate_by_token_pin(struct crypt_device *cd,
>     const char *name,
>     const char *type,
>     int token,
>     const char *pin,
>     size_t pin_size,
>     void *usrptr,
>     uint32_t flags);
333c347,351
<         int (*progress)(uint64_t size, uint64_t offset, void *usrptr));
____
>         int (*progress)(uint64_t size, uint64_t offset, void *usrptr))
> __attribute__((deprecated));
> int crypt_reencrypt_run(struct crypt_device *cd,
>         int (*progress)(uint64_t size, uint64_t offset, void *usrptr),
>         void *usrptr);

```

papi-devel

改变 f77papi.h

```

13c13
<     PARAMETER (PAPI_VER_CURRENT=84279296)
____
>     PARAMETER (PAPI_VER_CURRENT=84344832)
15c15
<     PARAMETER (PAPI_VERSION=84279296)
____
>     PARAMETER (PAPI_VERSION=84344832)
73c73
<     PARAMETER (PAPI_NUM_ERRORS=25)
____
>     PARAMETER (PAPI_NUM_ERRORS=26)
243a244,245
>     INTEGER PAPI_ECMP_DISABLED
>     PARAMETER (PAPI_ECMP_DISABLED=--25)
改变 f90papi.h

```

11,12c11,12

< INTEGER, PARAMETER :: PAPI_VER_CURRENT = 84279296

< INTEGER, PARAMETER :: PAPI_VERSION = 84279296

> INTEGER, PARAMETER :: PAPI_VER_CURRENT = 84344832

> INTEGER, PARAMETER :: PAPI_VERSION = 84344832

41c41

< INTEGER, PARAMETER :: PAPI_NUM_ERRORS = 25

> INTEGER, PARAMETER :: PAPI_NUM_ERRORS = 26

126a127

> INTEGER, PARAMETER :: PAPI_ECMP_DISABLED = -25

NetworkManager-libnm-devel

改变 nm-setting-ethtool.h

7a8,9

> NM_AVAILABLE_IN_1_26

> gboolean nm_ethtool_optname_is_pause(const char *optname);

改变 nm-setting-wired.h

9a10,11

> NM_AVAILABLE_IN_1_26

> NM Ternary nm_setting_wired_get_accept_all_mac_addresses(NMSettingWired *setting);

avahi-devel

减少头文件: qt-watch.h

python3-devel

增加头文件: kysec-api.h

增加头文件: pycore_long.h

qt5-qtbase-devel

改变 qbytearray.h

29a30

```

> Q_CORE_EXPORT int qstrnicmp(const char *, qsize_t, const char *, qsize_t = -1);
100a102,103
>     inline int compare(const char *c, Qt::CaseSensitivity cs = Qt::CaseSensitive) const;
>     inline int compare(const QByteArray &a, Qt::CaseSensitivity cs = Qt::CaseSensitive) const;
377a381,390
> inline int QByteArray::compare(const char *c, Qt::CaseSensitivity cs) const
> {
>     return cs == Qt::CaseSensitive ? qstrcmp(*this, c) :
>                                     qstrnicmp(data(), size(), c, -1);
> }
> inline int QByteArray::compare(const QByteArray &a, Qt::CaseSensitivity cs) const
> {
>     return cs == Qt::CaseSensitive ? qstrcmp(*this, a) :
>                                     qstrnicmp(data(), size(), a.data(), a.size());
> }
改变 qfixed_p.h
41a42,48
> inline bool qAddOverflow(QFixed v1, QFixed v2, QFixed *r)
> {
>     int val;
>     bool result = add_overflow(v1.value(), v2.value(), &val);
>     r->setValue(val);
>     return result;
> }
改变 qxmlstream_p.h
592a593,596
>     XmlContext currentContext = XmlContext::Prolog;
>     bool foundDTD = false;
>     bool isValidToken(QXmlStreamReader::TokenType type);
>     void checkToken();
723c727
<     int fastScanName(int *prefix = 0);
——
>     FastScanNameResult fastScanName(Value *val = 0);
728a733
>     void raiseNamePrefixTooLongError();
1533c1538,1542
<         sym(1).len += fastScanName(&sym(1).prefix);
——
>         Value &val = sym(1);
>         if (auto res = fastScanName(&val))
>             val.len += *res;
>         else
>             return false;

```

1540c1549,1552

```
<          sym(1).len += fastScanName();
——
>          if (auto res = fastScanName())
>              sym(1).len += *res;
>          else
>              return false;
```

libvncserver-devel

改变 rfbclient.h

3a4,5

```
> typedef rfbBool (*LockWriteToTLSProc)(struct _rfbClient* client);
> typedef rfbBool (*UnlockWriteToTLSProc)(struct _rfbClient* client);
```

37c39

```
< extern rfbBool WriteToRFBServer(rfbClient* client, char *buf, int n);
——
```

```
> extern rfbBool WriteToRFBServer(rfbClient* client, const char *buf, unsigned int n);
```

39,46c41,52

```
< extern int ListenAtTcpPort(int port);
< extern int ListenAtTcpPortAndAddress(int port, const char *address);
< extern int ConnectClientToTcpAddr(unsigned int host, int port);
< extern int ConnectClientToTcpAddr6(const char *hostname, int port);
< extern int ConnectClientToUnixSock(const char *sockFile);
< extern int AcceptTcpConnection(int listenSock);
< extern rfbBool SetNonBlocking(int sock);
< extern rfbBool SetDSCP(int sock, int dscp);
——
```

```
> extern rfbSocket ListenAtTcpPort(int port);
> extern rfbSocket ListenAtTcpPortAndAddress(int port, const char *address);
> extern rfbSocket ConnectClientToTcpAddr(unsigned int host, int port);
> extern rfbSocket ConnectClientToTcpAddr6(const char *hostname, int port);
> extern rfbSocket ConnectClientToUnixSock(const char *sockFile);
> extern rfbSocket ConnectClientToTcpAddrWithTimeout(unsigned int host, int port, unsigned int timeout);
> extern rfbSocket ConnectClientToTcpAddr6WithTimeout(const char *hostname, int port, unsigned int timeout);
> extern rfbSocket ConnectClientToUnixSockWithTimeout(const char *sockFile, unsigned int timeout);
> extern rfbSocket AcceptTcpConnection(rfbSocket listenSock);
> extern rfbBool SetNonBlocking(rfbSocket sock);
> extern rfbBool SetBlocking(rfbSocket sock);
> extern rfbBool SetDSCP(rfbSocket sock, int dscp);
```

48c54

```
< extern rfbBool SameMachine(int sock);
——
```

```
> extern rfbBool SameMachine(rfbSocket sock);
```

改变 rfb.h

22,26c22,26

```
< extern int rfbConnect(rfbScreenInfoPtr rfbScreen, char* host, int port);
```

```
< extern int rfbConnectToTcpAddr(char* host, int port);
```

```
< extern int rfbListenOnTCPPort(int port, in_addr_t iface);
```

```
< extern int rfbListenOnTCP6Port(int port, const char* iface);
```

```
< extern int rfbListenOnUDPPort(int port, in_addr_t iface);
```

```
> extern rfbSocket rfbConnect(rfbScreenInfoPtr rfbScreen, char* host, int port);
```

```
> extern rfbSocket rfbConnectToTcpAddr(char* host, int port);
```

```
> extern rfbSocket rfbListenOnTCPPort(int port, in_addr_t iface);
```

```
> extern rfbSocket rfbListenOnTCP6Port(int port, const char* iface);
```

```
> extern rfbSocket rfbListenOnUDPPort(int port, in_addr_t iface);
```

28c28

```
< extern rfbBool rfbSetNonBlocking(int sock);
```

```
> extern rfbBool rfbSetNonBlocking(rfbSocket sock);
```

40,41c40,41

```
< extern void rfbNewClientConnection(rfbScreenInfoPtr rfbScreen,int sock);
```

```
< extern rfbClientPtr rfbNewClient(rfbScreenInfoPtr rfbScreen,int sock);
```

```
> extern void rfbNewClientConnection(rfbScreenInfoPtr rfbScreen,rfbSocket sock);
```

```
> extern rfbClientPtr rfbNewClient(rfbScreenInfoPtr rfbScreen,rfbSocket sock);
```

47c47

```
< extern void rfbNewUDPConnection(rfbScreenInfoPtr rfbScreen,int sock);
```

```
> extern void rfbNewUDPConnection(rfbScreenInfoPtr rfbScreen,rfbSocket sock);
```

55a56

```
> extern rfbBool rfbSendExtDesktopSize(rfbClientPtr cl, int w, int h);
```

94a96,97

```
> extern rfbBool rfbSendTightHeader(rfbClientPtr cl, int x, int y, int w, int h);
```

```
> extern rfbBool rfbSendCompressedDataTight(rfbClientPtr cl, char *buf, int compressedLen);
```

增加头文件: threading.h

liborcus-devel

改变 css_document_tree.hpp

14c14

```
< void load(const std::string& strm);
```

```
> void load(const char* p, size_t n);
```

改变 css_parser_base.hpp

14c14

```
< void identifier(const char*& p, size_t& len, const char* extra = nullptr);
```

```
> void identifier(const char*& p, size_t& len, const char* extra = nullptr, size_t n_extra = 0);
```

20c20

```
< void skip_to_or_blank(const char*& p, size_t& len, const char* chars);
```

```
> void skip_to_or_blank(const char*& p, size_t& len, const char* chars, size_t n_chars);
```

改变 css_parser.hpp

234c234

```
< if (!is_alpha(c) && !is_numeric(c) && !is_in(c, "-+.")) css::parse_error::throw_with("value: illegal  
first character of a value '", c, "");
```

```
> if (!is_alpha(c) && !is_numeric(c) && !is_in(c, ORCUS_ASCII("-+.")))  
css::parse_error::throw_with("value: illegal first character of a value '", c, "");
```

237c237

```
< identifier(p, len, ".%");
```

```
> identifier(p, len, ORCUS_ASCII(".%"));
```

378c378

```
< skip_to_or_blank(p, len, "");
```

```
> skip_to_or_blank(p, len, ORCUS_ASCII(""));
```

改变 dom_tree.hpp

3d2

```
< struct dom_tree_impl;
```

7c6,8

```
< class ORCUS_DLLPUBLIC dom_tree
```

```
> namespace dom {
```

```
> class document_tree;
```

```
> class ORCUS_DLLPUBLIC const_node
```

9,10c10,13

```
< dom_tree(const dom_tree&) = delete;
```

```
< dom_tree& operator= (const dom_tree&) = delete;
```

```
> friend class document_tree;
```

```
> struct impl;
```

```
> std::unique_ptr<impl> mp_impl;
```

```
> const_node(std::unique_ptr<impl>&& _impl);
```

12,14c15,41

```
< }
```

```
< dom_tree(xmlns_context& cxt);
```

```
< ~dom_tree();
```



```



---


>     const_node();
>     const_node(const const_node& other);
>     const_node(const_node&& other);
>     ~const_node();
>     node_t type() const;
>     size_t child_count() const;
>     const_node child(size_t index) const;
>     entity_name name() const;
>     pstring attribute(const entity_name& name) const;
>     pstring attribute(const pstring& name) const;
>     size_t attribute_count() const;
>     const_node parent() const;
>     void swap(const_node& other);
>     const_node& operator= (const const_node& other);
>     bool operator== (const const_node& other) const;
>     bool operator!= (const const_node& other) const;
> };
> class ORCUS_DLLPUBLIC document_tree
> {
>     struct impl;
>     std::unique_ptr<impl> mp_impl;
> public:
>     document_tree(const document_tree&) = delete;
>     document_tree& operator= (const document_tree&) = delete;
>     document_tree(xmlns_context& cxt);
>     document_tree(document_tree&& other);
>     ~document_tree();
16,23c43,46
<     void swap(dom_tree& other);
<     void start_declaration(const pstring& name);
<     void end_declaration(const pstring& name);
<     void start_element(xmlns_id_t ns, const pstring& name);
<     void end_element(xmlns_id_t ns, const pstring& name);
<     void set_characters(const pstring& val);
<     void set_attribute(xmlns_id_t ns, const pstring& name, const pstring& val);
<     void set_doctype(const sax::doctype_declaration& dtd);


---


>     void load(const char* p_strm, size_t n_strm);
>     dom::const_node root() const;
>     dom::const_node declaration(const pstring& name) const;
>     void swap(document_tree& other);
25d47
<     const attrs_type* get_declaration_attributes(const pstring& name) const;

```

27a50

> }

改变 exception.hpp

8a9,10

> protected:

> void append_msg(const std::string& s);

20a23,34

> };

> class ORCUS_PSR_DLLPUBLIC json_structure_error : public general_error

> {

> public:

> explicit json_structure_error(const ::std::string& msg);

> virtual ~json_structure_error() throw();

> };

> class ORCUS_PSR_DLLPUBLIC invalid_map_error : public general_error

> {

> public:

> explicit invalid_map_error(const ::std::string& msg);

> virtual ~invalid_map_error() throw();

改变 global.hpp

3,5c3,5

< void print_element(xmlns_id_t ns, xml_token_t name);

< void print_attrs(const tokens& tokens, const xml_attrs_t& attrs);

< date_time_t to_date_time(const pstring& str);

> ORCUS_DLLPUBLIC void print_element(xmlns_id_t ns, xml_token_t name);

> ORCUS_DLLPUBLIC void print_attrs(const tokens& tokens, const xml_attrs_t& attrs);

> ORCUS_DLLPUBLIC date_time_t to_date_time(const pstring& str);

改变 import_interface.hpp

218a219

> virtual void fill_down_cells(row_t src_row, col_t src_col, row_t range_size) = 0;

改变 interface.hpp

7c7

< impl* mp_impl;

> std::unique_ptr<impl> mp_impl;

21,24c21

< virtual void dump_flat(const std::string& outdir) const = 0;

< virtual void dump_html(const std::string& outdir) const = 0;

< virtual void dump_json(const std::string& outdir) const = 0;

< virtual void dump_csv(const std::string& outdir) const = 0;

> virtual void dump(dump_format_t format, const std::string& output) const = 0;

改变 json_document_tree.hpp

3d2

```
< class string_pool;
```

7c6

```
< struct json_value_store;
```

```
> struct document_resource;
```

21a21,42

```
> class const_node;
```

```
> class document_tree;
```

```
> class ORCUS_DLLPUBLIC const_node_iterator
```

```
> {
```

```
>     friend class const_node;
```

```
>     struct impl;
```

```
>     std::unique_ptr<impl> mp_impl;
```

```
>     const_node_iterator(const document_tree* doc, const const_node& v, bool begin);
```

```
> public:
```

```
>     const_node_iterator();
```

```
>     const_node_iterator(const const_node_iterator& other);
```

```
>     ~const_node_iterator();
```

```
>     const const_node& operator*() const;
```

```
>     const const_node* operator->() const;
```

```
>     const_node_iterator& operator++();
```

```
>     const_node_iterator operator++(int);
```

```
>     const_node_iterator& operator--();
```

```
>     const_node_iterator operator--(int);
```

```
>     bool operator==(const const_node_iterator& other) const;
```

```
>     bool operator!=(const const_node_iterator& other) const;
```

```
>     const_node_iterator& operator= (const const_node_iterator& other);
```

```
> };
```

24a46

```
>     friend class const_node_iterator;
```

38a61

```
>     bool has_key(const pstring& key) const;
```

41a65

```
>     const_node back() const;
```

45a70,71

```
>     const_node_iterator begin() const;
```

```
>     const_node_iterator end() const;
```

62a89

```
>     node back();
```

69c96

```
<     std::initializer_list<detail::init::node> m_vs;
```

```
>     std::vector<detail::init::node> m_vs;
```

97a125

```
>     node(const std::string& s);
```

101c129

```
<     node(const node&) = delete;
```

```
>     node(const node& other) = delete;
```

104c132

```
<     node& operator= (node) = delete;
```

```
>     node& operator= (node other) = delete;
```

113c141

```
<     const string_pool& get_string_pool() const;
```

```
>     const document_resource& get_resource() const;
```

118c146

```
<     document_tree(string_pool& pool);
```

```
>     document_tree(document_resource& res);
```

改变 json_parser_base.hpp

20a21

```
>     void skip_ws();
```

24d24

```
<     long parse_long_or_throw();
```

27d26

```
<     void skip_blanks();
```

改变 json_parser.hpp

19c19

```
<     skip_blanks();
```

```
>     skip_ws();
```

21a22,23

```
>     else
```

```
>         throw json::parse_error("parse: no json content could be found in file", offset());
```

88a91

```
>     skip_ws();
```

91,93c94

```
<         m_handler.end_array();
```

```
<         next();
```

```
<         skip_blanks();
```

```
>         end_array();
```

96d96

```
<         skip_blanks();
```

98c98

```

<         skip_blanks();
-----
>         skip_ws();
104,106c104
<             m_handler.end_array();
<             next();
<             skip_blanks();
-----
>             end_array();
108a107,111
>             if (next_char() == ']')
>             {
>                 json::parse_error::throw_with(
>                     "array: ']' expected but '"', cur_char(), '" found."', offset() );
>             }
122a126,132
> void json_parser<_Handler>::end_array()
> {
>     m_handler.end_array();
>     next();
>     skip_ws();
> }
> template<typename _Handler>
125a136
>     bool require_new_key = false;
129c140
<         skip_blanks();
-----
>         skip_ws();
134a146,150
>         if (require_new_key)
>         {
>             json::parse_error::throw_with(
>                 "object: new key expected, but '"', cur_char(), '" found."', offset());
>         }
137c153
<         skip_blanks();
-----
>         skip_ws();
144a161
>         require_new_key = false;
157c174
<         skip_blanks();
-----

```

```

>         skip_ws();
162c179
<         skip_blanks();
____

>         skip_ws();
166c183
<         skip_blanks();
____

>         skip_ws();
168c185
<         throw json::parse_error("object: stream ended prematurely before reaching either ']' or '!', offset());
____

>         throw json::parse_error("object: stream ended prematurely before reaching either '}' or '!', offset());
174c191
<         skip_blanks();
____

>         skip_ws();
176a194
>         require_new_key = true;
180c198
<         "object: either ']' or '!' expected, but '"', cur_char(), '" found.", offset());
____

>         "object: either '}' or '!' expected, but '"', cur_char(), '" found.", offset());
191c209
<         skip_blanks();
____

>         skip_ws();
增加头文件: json_structure_tree.hpp
增加头文件: orcus_json.hpp
改变 orcus_xls_xml.hpp
6c6,7
<         orcus_xls_xml(const orcus_xls_xml&);         orcus_xls_xml& operator= (const orcus_xls_xml&);
____

>         struct impl;
>         std::unique_ptr<impl> mp_impl;
9a11,12
>         orcus_xls_xml(const orcus_xls_xml&) = delete;
>         orcus_xls_xml& operator= (const orcus_xls_xml&) = delete;
改变 orcus_xml.hpp
11c11,12
<         orcus_xml(const orcus_xml&);         orcus_xml& operator= (const orcus_xml&);
____

>         struct impl;
>         std::unique_ptr<impl> mp_impl;

```

13a15,16

```
> orcus_xml(const orcus_xml&) = delete;
> orcus_xml& operator= (const orcus_xml&) = delete;
```

19a23

```
> void set_range_row_group(const pstring& xpath);
```

22a27

```
> void read_map_definition(const char* p, size_t n);
```

改变 parser_base.hpp

17a18

```
> const bool m_transient_stream;
```

19c20,21

```
< parser_base(const char* p, size_t n);
```

```
> parser_base(const char* p, size_t n, bool transient_stream);
```

```
> bool transient_stream() const { return m_transient_stream; }
```

25,26c27,32

```
< bool has_next() const { return (mp_char+1) != mp_end; }
```

```
< void next(size_t inc=1);
```

```
> bool has_next() const
```

```
> {
```

```
>     assert((mp_char+1) <= mp_end);
```

```
>     return (mp_char+1) != mp_end;
```

```
> }
```

```
> void next(size_t inc=1) { mp_char += inc; }
```

28c34

```
< char cur_char() const;
```

```
> char cur_char() const { return *mp_char; }
```

30,31c36,38

```
< void skip(const char* chars_to_skip);
```

```
< bool parse_expected(const char* expected);
```

```
> void skip(const char* chars_to_skip, size_t n_chars_to_skip);
```

```
> void skip_space_and_control();
```

```
> bool parse_expected(const char* expected, size_t n_expected);
```

33a41,44

```
> size_t available_size() const
```

```
> {
```

```
>     return std::distance(mp_char, mp_end);
```

```
> }
```

改变 parser_global.hpp

7c7

```
< ORCUS_PSR_DLLPUBLIC bool is_in(char c, const char* allowed);
```

```

——
> ORCUS_PSR_DLLPUBLIC bool is_in(char c, const char* allowed, size_t n_allowed);
改变 sax_ns_parser.hpp
18a19,20
>     sax_ns_parser(const char* content, const size_t size, bool transient_stream,
>
>         xmlns_context& ns_cxt, handler_type& handler);
28a31,36
> {
> }
> template<typename _Handler>
> sax_ns_parser<_Handler>::sax_ns_parser(
>     const char* content, const size_t size, bool transient_stream, xmlns_context& ns_cxt, handler_type&
handler) :
>     m_wrapper(ns_cxt, handler), m_parser(content, size, transient_stream, m_wrapper)
改变 sax_parser_base.hpp
25c25
<     parser_base(const char* content, size_t size);
——
>     parser_base(const char* content, size_t size, bool transient_stream);
74d73
<     void blank();
改变 sax_parser.hpp
6a7
>     sax_parser(const char* content, const size_t size, bool transient_stream, handler_type& handler);
15c16,23
<     sax::parser_base(content, size),
——
>     sax::parser_base(content, size, false),
>     m_handler(handler)
> {
> }
> template<typename _Handler, typename _Config>
> sax_parser<_Handler, _Config>::sax_parser(
>     const char* content, const size_t size, bool transient_stream, handler_type& handler) :
>     sax::parser_base(content, size, transient_stream),
29c37
<     blank();
——
>     skip_space_and_control();
37c45
<     blank();
——
>     skip_space_and_control();
95c103

```



```

<      blank();
----
>      skip_space_and_control();
172c180
<      blank();
----
>      skip_space_and_control();
196c204
<      blank();
----
>      skip_space_and_control();
200c208
<      blank();
----
>      skip_space_and_control();
228c236
<      m_handler.characters(pstring(p0, cdata_len), false);
----
>      m_handler.characters(pstring(p0, cdata_len), transient_stream());
242c250
<      blank();
----
>      skip_space_and_control();
260c268
<      blank();
----
>      skip_space_and_control();
264c272
<      blank();
----
>      skip_space_and_control();
275c283
<      blank();
----
>      skip_space_and_control();
298c306
<      m_handler.characters(pstring(), false);
----
>      m_handler.characters(pstring(), transient_stream());
307c315
<      m_handler.characters(val, false);
----
>      m_handler.characters(val, transient_stream());
316,317c324,325

```

```

<      std::ostringstream os;
<      os << "sax_parser::attribute: ns=\"" << attr.ns << "\", name=\"" << attr.name << "\"";
——
>      cout << "sax_parser::attribute: ns=\"" << attr.ns << "\", name=\"" << attr.name << "\" << endl;
>      skip_space_and_control();
325c333
<      next_check();
——
>      next_check();      skip_space_and_control();
329,330c337
<      os << " value=\"" << attr.value << "\" << endl;
<      cout << os.str();
——
>      cout << "sax_parser::attribute: value=\"" << attr.value << "\" << endl;

```

改变 sax_token_parser.hpp

```

39a40,42
>      sax_token_parser(
>          const char* content, const size_t size, bool transient_stream,
>          const tokens& _tokens, xmlns_context& ns_cxt, handler_type& handler);
50a54,61
> {
> }
> template<typename _Handler>
> sax_token_parser<_Handler>::sax_token_parser(
>     const char* content, const size_t size, bool transient_stream,
>     const tokens& _tokens, xmlns_context& ns_cxt, handler_type& handler) :
>     m_wrapper(_tokens, handler),
>     m_parser(content, size, transient_stream, ns_cxt, m_wrapper)

```

改变 stream.hpp

```

2c2,38
< ORCUS_PSR_DLLPUBLIC std::string load_file_content(const char* filepath);
——
> class ORCUS_PSR_DLLPUBLIC file_content
> {
>     struct impl;
>     std::unique_ptr<impl> mp_impl;
> public:
>     file_content(const file_content&) = delete;
>     file_content& operator= (const file_content&) = delete;
>     file_content();
>     file_content(file_content&& other);
>     file_content(const char* filepath);
>     ~file_content();
>     const char* data() const;

```

```

>     size_t size() const;
>     bool empty() const;
>     void swap(file_content& other);
>     void load(const char* filepath);
>     void convert_to_utf8();
>     pstring str() const;
> };
> class ORCUS_PSR_DLLPUBLIC memory_content
> {
>     struct impl;
>     std::unique_ptr<impl> mp_impl;
> public:
>     memory_content(const file_content&) = delete;
>     memory_content& operator= (const file_content&) = delete;
>     memory_content();
>     memory_content(const char* p, size_t n);
>     memory_content(memory_content&& other);
>     ~memory_content();
>     const char* data() const;
>     size_t size() const;
>     bool empty() const;
>     void swap(memory_content& other);
>     void convert_to_utf8();
>     pstring str() const;
> };

```

改变 types.hpp

1a2

```

> ORCUS_PSR_DLLPUBLIC extern const size_t index_not_found;
4,6c5,7
< ORCUS_PSR_DLLPUBLIC extern const size_t index_not_found;
< ORCUS_PSR_DLLPUBLIC extern const size_t unspecified;
< ORCUS_DLLPUBLIC std::ostream& operator<< (std::ostream& os, const date_time_t& v);
——
> ORCUS_PSR_DLLPUBLIC dump_format_t to_dump_format_enum(const char* p, size_t n);
> ORCUS_PSR_DLLPUBLIC std::vector<std::pair<pstring, dump_format_t>> get_dump_format_entries();
> ORCUS_PSR_DLLPUBLIC std::ostream& operator<< (std::ostream& os, const date_time_t& v);

```

改变 xml_structure_tree.hpp

3d2

```

< struct xml_structure_tree_impl;
6c5,6
<         xml_structure_tree(const xml_structure_tree&);           xml_structure_tree& operator= (const
xml_structure_tree&);
——
>     struct impl;

```

```

>      std::unique_ptr<impl> mp_impl;
7a8,9
>      xml_structure_tree(const xml_structure_tree&) = delete;
>      xml_structure_tree& operator= (const xml_structure_tree&) = delete;
12,13c14,15
<          walker_impl* mp_impl;
<          walker();          walker(const xml_structure_tree_impl& parent_impl);
----
>      std::unique_ptr<walker_impl> mp_impl;
>      walker(const xml_structure_tree::impl& parent_impl);
14a17
>      walker() = delete;
改变 yaml_document_tree.hpp
44a45
>      void load(const char* p, size_t n);

```

libgcrypt-devel

```

改变 gcrypt.h
86a87
> gpg_error_t gcry_mpi_get_ui (unsigned int *w, gcry_mpi_t u);
227a229,233
>

```

```
> unsigned int gcry_ecc_get_algo_keylen (int curveid);
> gpg_error_t gcry_ecc_mul_point (int curveid, unsigned char *result,
>
>                               const unsigned char *scalar,
>
>                               const unsigned char *point);
```

sleuthkit-devel

改变 tsk_hfs.h

6c6

```
<     const hfs_uni_str *);
____
>     int, const hfs_uni_str *);
```

hdf5-mpich-devel

改变 H5Apublic.h

2,17c2,18

```
< H5_DLL hid_t   H5Acreate2(hid_t loc_id, const char *attr_name, hid_t type_id,
<     hid_t space_id, hid_t acpl_id, hid_t aapl_id);
< H5_DLL hid_t   H5Acreate_by_name(hid_t loc_id, const char *obj_name, const char *attr_name,
<     hid_t type_id, hid_t space_id, hid_t acpl_id, hid_t aapl_id, hid_t lapl_id);
< H5_DLL hid_t   H5Aopen(hid_t obj_id, const char *attr_name, hid_t aapl_id);
< H5_DLL hid_t   H5Aopen_by_name(hid_t loc_id, const char *obj_name,
<     const char *attr_name, hid_t aapl_id, hid_t lapl_id);
< H5_DLL hid_t   H5Aopen_by_idx(hid_t loc_id, const char *obj_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t aapl_id,
<     hid_t lapl_id);
< H5_DLL herr_t   H5Awrite(hid_t attr_id, hid_t type_id, const void *buf);
< H5_DLL herr_t   H5Aread(hid_t attr_id, hid_t type_id, void *buf);
< H5_DLL herr_t   H5Aclose(hid_t attr_id);
< H5_DLL hid_t   H5Aget_space(hid_t attr_id);
< H5_DLL hid_t   H5Aget_type(hid_t attr_id);
< H5_DLL hid_t   H5Aget_create_plist(hid_t attr_id);
____
> H5_DLL herr_t H5Aclose(hid_t attr_id);
> H5_DLL hid_t H5Acreate2(hid_t loc_id, const char *attr_name, hid_t type_id, hid_t space_id, hid_t acpl_id,
>
>                               hid_t aapl_id);
> H5_DLL hid_t H5Acreate_by_name(hid_t loc_id, const char *obj_name, const char *attr_name, hid_t type_id,
>
>                               hid_t space_id, hid_t acpl_id, hid_t aapl_id, hid_t lapl_id);
> H5_DLL herr_t H5Adelete(hid_t loc_id, const char *attr_name);
> H5_DLL herr_t H5Adelete_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type, H5_iter_order_t
```

```

order,
>
> hsize_t n, hid_t lapl_id);
> H5_DLL herr_t H5Adelete_by_name(hid_t loc_id, const char *obj_name, const char *attr_name, hid_t lapl_id);
> H5_DLL htri_t H5Aexists(hid_t obj_id, const char *attr_name);
> H5_DLL htri_t H5Aexists_by_name(hid_t obj_id, const char *obj_name, const char *attr_name, hid_t lapl_id);
> H5_DLL hid_t H5Aget_create_plist(hid_t attr_id);
> H5_DLL herr_t H5Aget_info(hid_t attr_id, H5A_info_t *ainfo );
> H5_DLL herr_t H5Aget_info_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
> H5_iter_order_t order, hsize_t n, H5A_info_t *ainfo , hid_t lapl_id);
> H5_DLL herr_t H5Aget_info_by_name(hid_t loc_id, const char *obj_name, const char *attr_name,
>
> H5A_info_t *ainfo , hid_t lapl_id);
19,21c20,23
< H5_DLL ssize_t H5Aget_name_by_idx(hid_t loc_id, const char *obj_name,
<
< H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<
< char *name , size_t size, hid_t lapl_id);
____
> H5_DLL ssize_t H5Aget_name_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
> H5_iter_order_t order, hsize_t n, char *name , size_t size,
>
> hid_t lapl_id);
> H5_DLL hid_t H5Aget_space(hid_t attr_id);
23,51c25,45
< H5_DLL herr_t H5Aget_info(hid_t attr_id, H5A_info_t *ainfo );
< H5_DLL herr_t H5Aget_info_by_name(hid_t loc_id, const char *obj_name,
<
< const char *attr_name, H5A_info_t *ainfo , hid_t lapl_id);
< H5_DLL herr_t H5Aget_info_by_idx(hid_t loc_id, const char *obj_name,
<
< H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<
< H5A_info_t *ainfo , hid_t lapl_id);
< H5_DLL herr_t H5Arename(hid_t loc_id, const char *old_name, const char *new_name);
< H5_DLL herr_t H5Arename_by_name(hid_t loc_id, const char *obj_name,
<
< const char *old_attr_name, const char *new_attr_name, hid_t lapl_id);
< H5_DLL herr_t H5Aiterate2(hid_t loc_id, H5_index_t idx_type,
<
< H5_iter_order_t order, hsize_t *idx, H5A_operator2_t op, void *op_data);
< H5_DLL herr_t H5Aiterate_by_name(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
<
< H5_iter_order_t order, hsize_t *idx, H5A_operator2_t op, void *op_data,
<
< hid_t lapd_id);
< H5_DLL herr_t H5Adelete(hid_t loc_id, const char *name);
< H5_DLL herr_t H5Adelete_by_name(hid_t loc_id, const char *obj_name,
<
< const char *attr_name, hid_t lapl_id);
< H5_DLL herr_t H5Adelete_by_idx(hid_t loc_id, const char *obj_name,
<
< H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t lapl_id);
< H5_DLL htri_t H5Aexists(hid_t obj_id, const char *attr_name);
< H5_DLL htri_t H5Aexists_by_name(hid_t obj_id, const char *obj_name,
<
< const char *attr_name, hid_t lapl_id);
< H5_DLL hid_t H5Acreate1(hid_t loc_id, const char *name, hid_t type_id,

```

```

<      hid_t space_id, hid_t acpl_id);
< H5_DLL hid_t   H5Aopen_name(hid_t loc_id, const char *name);
< H5_DLL hid_t   H5Aopen_idx(hid_t loc_id, unsigned idx);
< H5_DLL int      H5Aget_num_attrs(hid_t loc_id);
< H5_DLL herr_t   H5Aiterate1(hid_t loc_id, unsigned *attr_num, H5A_operator1_t op,
<      void *op_data);
——
> H5_DLL hid_t H5Aget_type(hid_t attr_id);
> H5_DLL herr_t H5Aiterate2(hid_t loc_id, H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
>      H5A_operator2_t op, void *op_data);
> H5_DLL herr_t H5Aiterate_by_name(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>      H5_iter_order_t order, hsize_t *idx, H5A_operator2_t op, void
*op_data,
>      hid_t lapl_id);
> H5_DLL hid_t H5Aopen(hid_t obj_id, const char *attr_name, hid_t aapl_id);
> H5_DLL hid_t H5Aopen_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type, H5_iter_order_t order,
>      hsize_t n, hid_t aapl_id, hid_t lapl_id);
> H5_DLL hid_t H5Aopen_by_name(hid_t loc_id, const char *obj_name, const char *attr_name, hid_t aapl_id,
>      hid_t lapl_id);
> H5_DLL herr_t H5Aread(hid_t attr_id, hid_t type_id, void *buf);
> H5_DLL herr_t H5Arename(hid_t loc_id, const char *old_name, const char *new_name);
> H5_DLL herr_t H5Awrite(hid_t attr_id, hid_t type_id, const void *buf);
> H5_DLL herr_t H5Arename_by_name(hid_t loc_id, const char *obj_name, const char *old_attr_name,
>      const char *new_attr_name, hid_t lapl_id);
> H5_DLL hid_t H5Acreate1(hid_t loc_id, const char *name, hid_t type_id, hid_t space_id, hid_t acpl_id);
> H5_DLL int H5Aget_num_attrs(hid_t loc_id);
> H5_DLL herr_t H5Aiterate1(hid_t loc_id, unsigned *idx, H5A_operator1_t op, void *op_data);
> H5_DLL hid_t H5Aopen_idx(hid_t loc_id, unsigned idx);
> H5_DLL hid_t H5Aopen_name(hid_t loc_id, const char *name);

```

52a47

>

改变 H5DOPublic.h

2,12c2,7

```

< H5_HLDLL herr_t H5DOWrite_chunk(hid_t dset_id,
<      hid_t dxpl_id,
<      uint32_t filters,
<      const hsize_t *offset,
<      size_t data_size,
<      const void *buf);
< H5_HLDLL herr_t H5DORead_chunk(hid_t dset_id,
<      hid_t dxpl_id,
<      const hsize_t *offset,
<      uint32_t *filters,
<      void *buf);

```



```

> H5_DLL herr_t H5Dread_chunk(hid_t dset_id, hid_t dxpl_id, const hsize_t *offset, uint32_t *filters,
>
> void *buf);
> H5_DLL herr_t H5Diterate(void *buf, hid_t type_id, hid_t space_id, H5D_operator_t op, void *operator_data);
> H5_DLL herr_t H5Dvlen_get_buf_size(hid_t dset_id, hid_t type_id, hid_t space_id, hsize_t *size);
> H5_DLL herr_t H5Dfill(const void *fill, hid_t fill_type_id, void *buf, hid_t buf_type_id, hid_t space_id);
27,30c31,37
< H5_DLL herr_t H5Dscatter(H5D_scatter_func_t op, void *op_data, hid_t type_id,
<
< hid_t dst_space_id, void *dst_buf);
< H5_DLL herr_t H5Dgather(hid_t src_space_id, const void *src_buf, hid_t type_id,
<
< size_t dst_buf_size, void *dst_buf, H5D_gather_func_t op, void *op_data);
-----
> H5_DLL herr_t H5Dflush(hid_t dset_id);
> H5_DLL herr_t H5Drefresh(hid_t dset_id);
> H5_DLL herr_t H5Dscatter(H5D_scatter_func_t op, void *op_data, hid_t type_id, hid_t dst_space_id,
>
> void *dst_buf);
> H5_DLL herr_t H5Dgather(hid_t src_space_id, const void *src_buf, hid_t type_id, size_t dst_buf_size,
>
> void *dst_buf, H5D_gather_func_t op, void *op_data);
> H5_DLL herr_t H5Dclose(hid_t dset_id);
32,34c39,42
< H5_DLL hid_t H5Dcreate1(hid_t file_id, const char *name, hid_t type_id,
<
< hid_t space_id, hid_t dcpl_id);
< H5_DLL hid_t H5Dopen1(hid_t file_id, const char *name);
-----
> H5_DLL herr_t H5Dformat_convert(hid_t dset_id);
> H5_DLL herr_t H5Dget_chunk_index_type(hid_t did, H5D_chunk_index_t *idx_type);
> H5_DLL hid_t H5Dcreate1(hid_t loc_id, const char *name, hid_t type_id, hid_t space_id, hid_t dcpl_id);
> H5_DLL hid_t H5Dopen1(hid_t loc_id, const char *name);
35a44
> H5_DLL herr_t H5Dvlen_reclaim(hid_t type_id, hid_t space_id, hid_t dxpl_id, void *buf);
改变 H5DSpublic.h
2,30c2,12
< H5_HLDLL herr_t H5DSattach_scale( hid_t did,
<
< hid_t dsid,
<
< unsigned int idx);
< H5_HLDLL herr_t H5DSdetach_scale( hid_t did,
<
< hid_t dsid,
<
< unsigned int idx);
< H5_HLDLL herr_t H5DSset_scale( hid_t dsid,
<
< const char *dimname);
< H5_HLDLL int H5DSget_num_scales( hid_t did,
<
< unsigned int dim);
< H5_HLDLL herr_t H5DSset_label( hid_t did,
<
< unsigned int idx,
<
< const char *label);

```

```

< H5_HLDLL ssize_t H5DSget_label( hid_t did,
<
<         unsigned int idx,
<         char *label,
<         size_t size);
< H5_HLDLL ssize_t H5DSget_scale_name( hid_t did,
<
<         char *name,
<         size_t size);
< H5_HLDLL htri_t H5DSis_scale( hid_t did);
< H5_HLDLL herr_t  H5DSiterate_scales( hid_t did,
<
<         unsigned int dim,
<         int *idx,
<         H5DS_iterate_t visitor,
<         void *visitor_data);
< H5_HLDLL htri_t H5DSis_attached( hid_t did,
<
<         hid_t dsid,
<         unsigned int idx);
-----
> H5_HLDLL herr_t H5DSattach_scale(hid_t did, hid_t dsid, unsigned int idx);
> H5_HLDLL herr_t H5DSdetach_scale(hid_t did, hid_t dsid, unsigned int idx);
> H5_HLDLL herr_t H5DSset_scale(hid_t dsid, const char *dimname);
> H5_HLDLL int H5DSget_num_scales(hid_t did, unsigned int dim);
> H5_HLDLL herr_t H5DSset_label(hid_t did, unsigned int idx, const char *label);
> H5_HLDLL ssize_t H5DSget_label(hid_t did, unsigned int idx, char *label, size_t size);
> H5_HLDLL ssize_t H5DSget_scale_name(hid_t did, char *name, size_t size);
> H5_HLDLL htri_t H5DSis_scale(hid_t did);
> H5_HLDLL herr_t H5DSiterate_scales(hid_t did, unsigned int dim, int *idx, H5DS_iterate_t visitor,
>
>         void *visitor_data);
> H5_HLDLL htri_t H5DSis_attached(hid_t did, hid_t dsid, unsigned int idx);

```

改变 H5Epubgen.h

2d1

```
< H5_DLLVAR hid_t H5E_DATASET_g;
```

4d2

```
< H5_DLLVAR hid_t H5E_STORAGE_g;
```

5a4

```
> H5_DLLVAR hid_t H5E_VOL_g;
```

15,17d13

```
< H5_DLLVAR hid_t H5E_PLIST_g;
```

```
< H5_DLLVAR hid_t H5E_LINK_g;
```

```
< H5_DLLVAR hid_t H5E_DATATYPE_g;
```

18a15

```
> H5_DLLVAR hid_t H5E_FARRAY_g;
```

20,21c17

```
< H5_DLLVAR hid_t H5E_OHDR_g;
```

```
< H5_DLLVAR hid_t H5E_ATOM_g;
```

> H5_DLLVAR hid_t H5E_MAP_g;

23d18

< H5_DLLVAR hid_t H5E_NONE_MAJOR_g;

25d19

< H5_DLLVAR hid_t H5E_SLIST_g;

27a22,33

> H5_DLLVAR hid_t H5E_LIB_g;

> H5_DLLVAR hid_t H5E_PAGEBUF_g;

> H5_DLLVAR hid_t H5E_FSPACE_g;

> H5_DLLVAR hid_t H5E_DATASET_g;

> H5_DLLVAR hid_t H5E_STORAGE_g;

> H5_DLLVAR hid_t H5E_LINK_g;

> H5_DLLVAR hid_t H5E_PLIST_g;

> H5_DLLVAR hid_t H5E_DATATYPE_g;

> H5_DLLVAR hid_t H5E_OHDR_g;

> H5_DLLVAR hid_t H5E_ATOM_g;

> H5_DLLVAR hid_t H5E_NONE_MAJOR_g;

> H5_DLLVAR hid_t H5E_SLIST_g;

29c35,36

< H5_DLLVAR hid_t H5E_ERROR_g;

> H5_DLLVAR hid_t H5E_CONTEXT_g;

> H5_DLLVAR hid_t H5E_EARRAY_g;

31c38

< H5_DLLVAR hid_t H5E_FSPACE_g;

> H5_DLLVAR hid_t H5E_ERROR_g;

95a103,105

> H5_DLLVAR hid_t H5E_CANTDELETEFILE_g;

> H5_DLLVAR hid_t H5E_CANTLOCKFILE_g;

> H5_DLLVAR hid_t H5E_CANTUNLOCKFILE_g;

102a113

> H5_DLLVAR hid_t H5E_CANTUNSERIALIZE_g;

103a115

> H5_DLLVAR hid_t H5E_CANTTAG_g;

113a126,128

> H5_DLLVAR hid_t H5E_CANTMARKCLEAN_g;

> H5_DLLVAR hid_t H5E_CANTMARKUNSERIALIZED_g;

> H5_DLLVAR hid_t H5E_CANTMARKSERIALIZED_g;

114a130

> H5_DLLVAR hid_t H5E_CANTCLEAN_g;

116a133,138

> H5_DLLVAR hid_t H5E_CANTDEPEND_g;

```

> H5_DLLVAR hid_t H5E_CANTUNDEPEND_g;
> H5_DLLVAR hid_t H5E_CANTNOTIFY_g;
> H5_DLLVAR hid_t H5E_LOGGING_g;
> H5_DLLVAR hid_t H5E_CANTCORK_g;
> H5_DLLVAR hid_t H5E_CANTUNCORK_g;
124a147,148
> H5_DLLVAR hid_t H5E_CANTGATHER_g;
> H5_DLLVAR hid_t H5E_NO_INDEPENDENT_g;
130a155,156
> H5_DLLVAR hid_t H5E_INCONSISTENTSTATE_g;
> H5_DLLVAR hid_t H5E_CANTAPPEND_g;
改变 H5Epublic.h
3,4c3
< H5_DLL hid_t  H5Eregister_class(const char *cls_name, const char *lib_name,
<      const char *version);
____
> H5_DLL hid_t H5Eregister_class(const char *cls_name, const char *lib_name, const char *version);
7,9c6,8
< H5_DLL hid_t  H5Ecreate_msg(hid_t cls, H5E_type_t msg_type, const char *msg);
< H5_DLL hid_t  H5Ecreate_stack(void);
< H5_DLL hid_t  H5Eget_current_stack(void);
____
> H5_DLL hid_t H5Ecreate_msg(hid_t cls, H5E_type_t msg_type, const char *msg);
> H5_DLL hid_t H5Ecreate_stack(void);
> H5_DLL hid_t H5Eget_current_stack(void);
13,14c12,13
< H5_DLL herr_t H5Epush2(hid_t err_stack, const char *file, const char *func, unsigned line,
<      hid_t cls_id, hid_t maj_id, hid_t min_id, const char *msg, ...);
____
> H5_DLL herr_t H5Epush2(hid_t err_stack, const char *file, const char *func, unsigned line, hid_t cls_id,
>      hid_t maj_id, hid_t min_id, const char *msg, ...);
17,18c16
< H5_DLL herr_t H5Ewalk2(hid_t err_stack, H5E_direction_t direction, H5E_walk2_t func,
<      void *client_data);
____
> H5_DLL herr_t H5Ewalk2(hid_t err_stack, H5E_direction_t direction, H5E_walk2_t func, void *client_data);
23,24c21
< H5_DLL ssize_t H5Eget_msg(hid_t msg_id, H5E_type_t *type, char *msg,
<      size_t size);
____
> H5_DLL ssize_t H5Eget_msg(hid_t msg_id, H5E_type_t *type, char *msg, size_t size);
28,29c25,26
< H5_DLL herr_t H5Epush1(const char *file, const char *func, unsigned line,
<      H5E_major_t maj, H5E_minor_t min, const char *str);

```

```
> H5_DLL herr_t H5Epush1(const char *file, const char *func, unsigned line, H5E_major_t maj, H5E_minor_t min,
>                        const char *str);
```

```

> H5_DLL herr_t H5Pget_fapl_family(hid_t fapl_id, hsize_t *memb_size, hid_t *memb_fapl_id);
增加头文件: H5FDhdfs.h
改变 H5FDlog.h
3d2
< H5_DLL void H5FD_log_term(void);
增加头文件: H5FDmirror.h
改变 H5FDmpi.h
0a1
>
改变 H5FDmpio.h
4d3
< H5_DLL void H5FD_mpio_term(void);
6,7c5
< H5_DLL herr_t H5Pget_fapl_mpio(hid_t fapl_id, MPI_Comm *comm,
<
MPI_Info *info);
____
> H5_DLL herr_t H5Pget_fapl_mpio(hid_t fapl_id, MPI_Comm *comm, MPI_Info *info);
9c7
< H5_DLL herr_t H5Pget_dxpl_mpio(hid_t dxpl_id, H5FD_mpio_xfer_t *xfer_mode);
____
> H5_DLL herr_t H5Pget_dxpl_mpio(hid_t dxpl_id, H5FD_mpio_xfer_t *xfer_mode);
改变 H5FDmulti.h
3,12c3,8
< H5_DLL void H5FD_multi_term(void);
< H5_DLL herr_t H5Pset_fapl_multi(hid_t fapl_id, const H5FD_mem_t *memb_map,
<
const hid_t *memb_fapl, const char * const *memb_name,
<
const haddr_t *memb_addr, hbool_t relax);
< H5_DLL herr_t H5Pget_fapl_multi(hid_t fapl_id, H5FD_mem_t *memb_map,
<
hid_t *memb_fapl, char **memb_name,
<
haddr_t *memb_addr, hbool_t *relax);
< H5_DLL herr_t H5Pset_fapl_split(hid_t fapl, const char *meta_ext,
<
hid_t meta_plist_id, const char *raw_ext,
<
hid_t raw_plist_id);
____
> H5_DLL herr_t H5Pset_fapl_multi(hid_t fapl_id, const H5FD_mem_t *memb_map, const hid_t *memb_fapl,
>
const char *const *memb_name, const haddr_t *memb_addr, hbool_t
relax);
> H5_DLL herr_t H5Pget_fapl_multi(hid_t fapl_id, H5FD_mem_t *memb_map, hid_t *memb_fapl,
>
char **memb_name, haddr_t *memb_addr, hbool_t *relax);
> H5_DLL herr_t H5Pset_fapl_split(hid_t fapl, const char *meta_ext, hid_t meta_plist_id, const char *raw_ext,
>
hid_t raw_plist_id);
改变 H5FDpublic.h
2c2
< H5_DLL hid_t H5FDregister(const H5FD_class_t *cls);

```

```

-----
> H5_DLL hid_t H5FDregister(const H5FD_class_t *cls);
4,8c4,7
< H5_DLL H5FD_t *H5FDopen(const char *name, unsigned flags, hid_t fapl_id,
<
haddr_t maxaddr);
< H5_DLL herr_t H5FDclose(H5FD_t *file);
< H5_DLL int H5FDcmp(const H5FD_t *f1, const H5FD_t *f2);
< H5_DLL int H5FDquery(const H5FD_t *f, unsigned long *flags);
-----

> H5_DLL H5FD_t *H5FDopen(const char *name, unsigned flags, hid_t fapl_id, haddr_t maxaddr);
> H5_DLL herr_t H5FDclose(H5FD_t *file);
> H5_DLL int H5FDcmp(const H5FD_t *f1, const H5FD_t *f2);
> H5_DLL int H5FDquery(const H5FD_t *f, unsigned long *flags);
10,11c9
< H5_DLL herr_t H5FDfree(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id,
<
haddr_t addr, hsize_t size);
-----

> H5_DLL herr_t H5FDfree(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id, haddr_t addr, hsize_t size);
13,21c11,22
< H5_DLL herr_t H5FDset_eoa(H5FD_t *file, H5FD_mem_t type, haddr_t eoa);
< H5_DLL haddr_t H5FDget_eof(H5FD_t *file);
< H5_DLL herr_t H5FDget_vfd_handle(H5FD_t *file, hid_t fapl, void **file_handle);
< H5_DLL herr_t H5FDread(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id,
<
haddr_t addr, size_t size, void *buf);
< H5_DLL herr_t H5FDwrite(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id,
<
haddr_t addr, size_t size, const void *buf);
< H5_DLL herr_t H5FDflush(H5FD_t *file, hid_t dxpl_id, unsigned closing);
< H5_DLL herr_t H5FDtruncate(H5FD_t *file, hid_t dxpl_id, hbool_t closing);
-----

> H5_DLL herr_t H5FDset_eoa(H5FD_t *file, H5FD_mem_t type, haddr_t eoa);
> H5_DLL haddr_t H5FDget_eof(H5FD_t *file, H5FD_mem_t type);
> H5_DLL herr_t H5FDget_vfd_handle(H5FD_t *file, hid_t fapl, void **file_handle);
> H5_DLL herr_t H5FDread(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id, haddr_t addr, size_t size,
>
void *buf);
> H5_DLL herr_t H5FDwrite(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id, haddr_t addr, size_t size,
>
const void *buf);
> H5_DLL herr_t H5FDflush(H5FD_t *file, hid_t dxpl_id, hbool_t closing);
> H5_DLL herr_t H5FDtruncate(H5FD_t *file, hid_t dxpl_id, hbool_t closing);
> H5_DLL herr_t H5FDlock(H5FD_t *file, hbool_t rw);
> H5_DLL herr_t H5FDunlock(H5FD_t *file);
> H5_DLL herr_t H5FDdriver_query(hid_t driver_id, unsigned long *flags);
增加头文件: H5FDros3.h
改变 H5FDsec2.h
2,3c2

```

```

< H5_DLL hid_t H5FD_sec2_init(void);
< H5_DLL void H5FD_sec2_term(void);
——
> H5_DLL hid_t H5FD_sec2_init(void);
增加头文件: H5FDsplitter.h
改变 H5FDstdio.h
3d2
< H5_DLL void H5FD_stdio_term(void);
增加头文件: H5FDwindows.h
改变 H5Fpublic.h
2,7c2,5
< H5_DLL htri_t H5Fis_hdf5(const char *filename);
< H5_DLL hid_t H5Fcreate(const char *filename, unsigned flags,
<
        hid_t create_plist, hid_t access_plist);
< H5_DLL hid_t H5Fopen(const char *filename, unsigned flags,
<
        hid_t access_plist);
< H5_DLL hid_t H5Freopen(hid_t file_id);
——
> H5_DLL htri_t H5Fis_accessible(const char *container_name, hid_t fapl_id);
> H5_DLL hid_t H5Fcreate(const char *filename, unsigned flags, hid_t fcpl_id, hid_t fapl_id);
> H5_DLL hid_t H5Fopen(const char *filename, unsigned flags, hid_t fapl_id);
> H5_DLL hid_t H5Freopen(hid_t file_id);
10,12c8,12
< H5_DLL hid_t H5Fget_create_plist(hid_t file_id);
< H5_DLL hid_t H5Fget_access_plist(hid_t file_id);
< H5_DLL herr_t H5Fget_intent(hid_t file_id, unsigned * intent);
——
> H5_DLL herr_t H5Fdelete(const char *filename, hid_t fapl_id);
> H5_DLL hid_t H5Fget_create_plist(hid_t file_id);
> H5_DLL hid_t H5Fget_access_plist(hid_t file_id);
> H5_DLL herr_t H5Fget_intent(hid_t file_id, unsigned *intent);
> H5_DLL herr_t H5Fget_fileno(hid_t file_id, unsigned long *fileno);
20,30c20,27
< H5_DLL ssize_t H5Fget_file_image(hid_t file_id, void * buf_ptr, size_t buf_len);
< H5_DLL herr_t H5Fget_mdc_config(hid_t file_id,
<
        H5AC_cache_config_t * config_ptr);
< H5_DLL herr_t H5Fset_mdc_config(hid_t file_id,
<
        H5AC_cache_config_t * config_ptr);
< H5_DLL herr_t H5Fget_mdc_hit_rate(hid_t file_id, double * hit_rate_ptr);
< H5_DLL herr_t H5Fget_mdc_size(hid_t file_id,
<
        size_t * max_size_ptr,
<
        size_t * min_clean_size_ptr,
<
        size_t * cur_size_ptr,
<
        int * cur_num_entries_ptr);

```



```

-----
> H5_DLL herr_t H5Fget_eoa(hid_t file_id, haddr_t *eoa);
> H5_DLL herr_t H5Fincrement_filesize(hid_t file_id, hsize_t increment);
> H5_DLL ssize_t H5Fget_file_image(hid_t file_id, void *buf_ptr, size_t buf_len);
> H5_DLL herr_t H5Fget_mdc_config(hid_t file_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Fset_mdc_config(hid_t file_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Fget_mdc_hit_rate(hid_t file_id, double *hit_rate_ptr);
> H5_DLL herr_t H5Fget_mdc_size(hid_t file_id, size_t *max_size_ptr, size_t *min_clean_size_ptr,
>                               size_t *cur_size_ptr, int *cur_num_entries_ptr);
33c30,34
< H5_DLL herr_t H5Fget_info(hid_t obj_id, H5F_info_t *bh_info);
-----
> H5_DLL herr_t H5Fget_info2(hid_t obj_id, H5F_info2_t *file_info);
> H5_DLL herr_t H5Fget_metadata_read_retry_info(hid_t file_id, H5F_retry_info_t *info);
> H5_DLL herr_t H5Fstart_swmr_write(hid_t file_id);
> H5_DLL ssize_t H5Fget_free_sections(hid_t file_id, H5F_mem_t type, size_t nsects,
>                                     H5F_sect_info_t *sect_info );
34a36,46
> H5_DLL herr_t H5Fset_libver_bounds(hid_t file_id, H5F_libver_t low, H5F_libver_t high);
> H5_DLL herr_t H5Fstart_mdc_logging(hid_t file_id);
> H5_DLL herr_t H5Fstop_mdc_logging(hid_t file_id);
> H5_DLL herr_t H5Fget_mdc_logging_status(hid_t file_id, hbool_t *is_enabled, hbool_t *is_currently_logging);
> H5_DLL herr_t H5Fformat_convert(hid_t fid);
> H5_DLL herr_t H5Freset_page_buffering_stats(hid_t file_id);
> H5_DLL herr_t H5Fget_page_buffering_stats(hid_t file_id, unsigned accesses[2], unsigned hits[2],
>                                           unsigned misses[2], unsigned evictions[2], unsigned
bypasses[2]);
> H5_DLL herr_t H5Fget_mdc_image_info(hid_t file_id, haddr_t *image_addr, hsize_t *image_size);
> H5_DLL herr_t H5Fget_dset_no_attrs_hint(hid_t file_id, hbool_t *minimize);
> H5_DLL herr_t H5Fset_dset_no_attrs_hint(hid_t file_id, hbool_t minimize);
36a49,51
> H5_DLL herr_t H5Fget_info1(hid_t obj_id, H5F_info1_t *file_info);
> H5_DLL herr_t H5Fset_latest_format(hid_t file_id, hbool_t latest_format);
> H5_DLL htri_t H5Fis_hdf5(const char *file_name);
37a53
>
改变 H5Gpublic.h
2,3c2
< H5_DLL hid_t H5Gcreate2(hid_t loc_id, const char *name, hid_t lcpl_id,
<     hid_t gcpl_id, hid_t gapl_id);
-----
> H5_DLL hid_t H5Gcreate2(hid_t loc_id, const char *name, hid_t lcpl_id, hid_t gcpl_id, hid_t gapl_id);
8,12c7,11
< H5_DLL herr_t H5Gget_info_by_name(hid_t loc_id, const char *name, H5G_info_t *ginfo,

```

```

<     hid_t lapl_id);
< H5_DLL herr_t H5Gget_info_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, H5G_info_t *ginfo,
<     hid_t lapl_id);
____

> H5_DLL herr_t H5Gget_info_by_name(hid_t loc_id, const char *name, H5G_info_t *ginfo, hid_t lapl_id);
> H5_DLL herr_t H5Gget_info_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>
>         H5_iter_order_t order, hsize_t n, H5G_info_t *ginfo, hid_t lapl_id);
> H5_DLL herr_t H5Gflush(hid_t group_id);
> H5_DLL herr_t H5Grefresh(hid_t group_id);
16,23c15,19
< H5_DLL herr_t H5Glink(hid_t cur_loc_id, H5G_link_t type, const char *cur_name,
<     const char *new_name);
< H5_DLL herr_t H5Glink2(hid_t cur_loc_id, const char *cur_name, H5G_link_t type,
<     hid_t new_loc_id, const char *new_name);
< H5_DLL herr_t H5Gmove(hid_t src_loc_id, const char *src_name,
<     const char *dst_name);
< H5_DLL herr_t H5Gmove2(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id,
<     const char *dst_name);
____

> H5_DLL herr_t H5Glink(hid_t cur_loc_id, H5G_link_t type, const char *cur_name, const char *new_name);
> H5_DLL herr_t H5Glink2(hid_t cur_loc_id, const char *cur_name, H5G_link_t type, hid_t new_loc_id,
>
>         const char *new_name);
> H5_DLL herr_t H5Gmove(hid_t src_loc_id, const char *src_name, const char *dst_name);
> H5_DLL herr_t H5Gmove2(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id, const char *dst_name);
25,26c21
< H5_DLL herr_t H5Gget_linkval(hid_t loc_id, const char *name, size_t size,
<     char *buf);
____

> H5_DLL herr_t H5Gget_linkval(hid_t loc_id, const char *name, size_t size, char *buf );
28,31c23,24
< H5_DLL int H5Gget_comment(hid_t loc_id, const char *name, size_t bufsize,
<     char *buf);
< H5_DLL herr_t H5Giterate(hid_t loc_id, const char *name, int *idx,
<     H5G_iterate_t op, void *op_data);
____

> H5_DLL int H5Gget_comment(hid_t loc_id, const char *name, size_t bufsize, char *buf);
> H5_DLL herr_t H5Giterate(hid_t loc_id, const char *name, int *idx, H5G_iterate_t op, void *op_data);
33,36c26,28
< H5_DLL herr_t H5Gget_objinfo(hid_t loc_id, const char *name,
<     hbool_t follow_link, H5G_stat_t *statbuf);
< H5_DLL ssize_t H5Gget_objname_by_idx(hid_t loc_id, hsize_t idx, char* name,
<     size_t size);
____

```

```
> H5_DLL herr_t H5Gget_objinfo(hid_t loc_id, const char *name, hbool_t follow_link,
>
>                                H5G_stat_t *statbuf );
> H5_DLL ssize_t H5Gget_objname_by_idx(hid_t loc_id, hsize_t idx, char *name, size_t size);
38a31
```

```
>
```

改变 H5IMpublic.h

2,47c2,18

```
< H5_HLDLL herr_t H5IMmake_image_8bit( hid_t loc_id,
<
<                                const char *dset_name,
<                                hsize_t width,
<                                hsize_t height,
<                                const unsigned char *buffer );
< H5_HLDLL herr_t H5IMmake_image_24bit( hid_t loc_id,
<
<                                const char *dset_name,
<                                hsize_t width,
<                                hsize_t height,
<                                const char *interlace,
<                                const unsigned char *buffer );
< H5_HLDLL herr_t H5IMget_image_info( hid_t loc_id,
<
<                                const char *dset_name,
<                                hsize_t *width,
<                                hsize_t *height,
<                                hsize_t *planes,
<                                char *interlace,
<                                hssize_t *npals );
< H5_HLDLL herr_t H5IMread_image( hid_t loc_id,
<
<                                const char *dset_name,
<                                unsigned char *buffer );
< H5_HLDLL herr_t H5IMmake_palette( hid_t loc_id,
<
<                                const char *pal_name,
<                                const hsize_t *pal_dims,
<                                const unsigned char *pal_data );
< H5_HLDLL herr_t H5IMlink_palette( hid_t loc_id,
<
<                                const char *image_name,
<                                const char *pal_name );
< H5_HLDLL herr_t H5IMunlink_palette( hid_t loc_id,
<
<                                const char *image_name,
<                                const char *pal_name );
< H5_HLDLL herr_t H5IMget_npalettes( hid_t loc_id,
<
<                                const char *image_name,
<                                hssize_t *npals );
< H5_HLDLL herr_t H5IMget_palette_info( hid_t loc_id,
<
<                                const char *image_name,
<                                int pal_number,
```

```

<                                     hsize_t *pal_dims );
< H5_HLDLL herr_t  H5IMget_palette( hid_t loc_id,
<                                     const char *image_name,
<                                     int pal_number,
<                                     unsigned char *pal_data );
< H5_HLDLL herr_t  H5IMis_image( hid_t loc_id,
<                                     const char *dset_name );
< H5_HLDLL herr_t  H5IMis_palette( hid_t loc_id,
<                                     const char *dset_name );
____
> H5_HLDLL herr_t H5IMmake_image_8bit(hid_t loc_id, const char *dset_name, hsize_t width, hsize_t height,
>                                     const unsigned char *buffer);
> H5_HLDLL herr_t H5IMmake_image_24bit(hid_t loc_id, const char *dset_name, hsize_t width, hsize_t height,
>                                     const char *interlace, const unsigned char *buffer);
> H5_HLDLL herr_t H5IMget_image_info(hid_t loc_id, const char *dset_name, hsize_t *width, hsize_t *height,
>                                     hsize_t *planes, char *interlace, hsize_t *npals);
> H5_HLDLL herr_t H5IMread_image(hid_t loc_id, const char *dset_name, unsigned char *buffer);
> H5_HLDLL herr_t H5IMmake_palette(hid_t loc_id, const char *pal_name, const hsize_t *pal_dims,
>                                     const unsigned char *pal_data);
> H5_HLDLL herr_t H5IMlink_palette(hid_t loc_id, const char *image_name, const char *pal_name);
> H5_HLDLL herr_t H5IMunlink_palette(hid_t loc_id, const char *image_name, const char *pal_name);
> H5_HLDLL herr_t H5IMget_npalletes(hid_t loc_id, const char *image_name, hsize_t *npals);
> H5_HLDLL herr_t H5IMget_palette_info(hid_t loc_id, const char *image_name, int pal_number, hsize_t
*pal_dims);
> H5_HLDLL herr_t H5IMget_palette(hid_t loc_id, const char *image_name, int pal_number,
>                                     unsigned char *pal_data);
> H5_HLDLL herr_t H5IMis_image(hid_t loc_id, const char *dset_name);
> H5_HLDLL herr_t H5IMis_palette(hid_t loc_id, const char *dset_name);

```

改变 H5Ipublic.h

3,4c3,4

```

< H5_DLL void *H5Iobject_verify(hid_t id, H5I_type_t id_type);
< H5_DLL void *H5Iremove_verify(hid_t id, H5I_type_t id_type);
____

```

```

> H5_DLL void *H5Iobject_verify(hid_t id, H5I_type_t type);
> H5_DLL void *H5Iremove_verify(hid_t id, H5I_type_t type);

```

7c7

```

< H5_DLL ssize_t H5Iget_name(hid_t id, char *name, size_t size);
____

```

```

> H5_DLL ssize_t H5Iget_name(hid_t id, char *name , size_t size);

```

17a18

```

> H5_DLL herr_t H5Iiterate(H5I_type_t type, H5I_iterate_func_t op, void *op_data);

```

21a23

```

>

```

增加头文件: H5LDpublic.h

改变 H5Lpublic.h

2,9c2,9

```
< H5_DLL herr_t H5Lmove(hid_t src_loc, const char *src_name, hid_t dst_loc,  
<     const char *dst_name, hid_t lcpl_id, hid_t lapl_id);  
< H5_DLL herr_t H5Lcopy(hid_t src_loc, const char *src_name, hid_t dst_loc,  
<     const char *dst_name, hid_t lcpl_id, hid_t lapl_id);  
< H5_DLL herr_t H5Lcreate_hard(hid_t cur_loc, const char *cur_name,  
<     hid_t dst_loc, const char *dst_name, hid_t lcpl_id, hid_t lapl_id);  
< H5_DLL herr_t H5Lcreate_soft(const char *link_target, hid_t link_loc_id,  
<     const char *link_name, hid_t lcpl_id, hid_t lapl_id);  
____
```

```
> H5_DLL herr_t H5Lmove(hid_t src_loc, const char *src_name, hid_t dst_loc, const char *dst_name, hid_t lcpl_id,  
>     hid_t lapl_id);  
> H5_DLL herr_t H5Lcopy(hid_t src_loc, const char *src_name, hid_t dst_loc, const char *dst_name, hid_t lcpl_id,  
>     hid_t lapl_id);  
> H5_DLL herr_t H5Lcreate_hard(hid_t cur_loc, const char *cur_name, hid_t dst_loc, const char *dst_name,  
>     hid_t lcpl_id, hid_t lapl_id);  
> H5_DLL herr_t H5Lcreate_soft(const char *link_target, hid_t link_loc_id, const char *link_name, hid_t lcpl_id,  
>     hid_t lapl_id);
```

11,17c11,16

```
< H5_DLL herr_t H5Ldelete_by_idx(hid_t loc_id, const char *group_name,  
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t lapl_id);  
< H5_DLL herr_t H5Lget_val(hid_t loc_id, const char *name, void *buf,  
<     size_t size, hid_t lapl_id);  
< H5_DLL herr_t H5Lget_val_by_idx(hid_t loc_id, const char *group_name,  
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,  
<     void *buf, size_t size, hid_t lapl_id);  
____  
> H5_DLL herr_t H5Ldelete_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,  
>     H5_iter_order_t order, hsize_t n, hid_t lapl_id);  
> H5_DLL herr_t H5Lget_val(hid_t loc_id, const char *name, void *buf , size_t size, hid_t lapl_id);  
> H5_DLL herr_t H5Lget_val_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,  
>     H5_iter_order_t order, hsize_t n, void *buf , size_t size,  
>     hid_t lapl_id);
```

19,39c18,34

```
< H5_DLL herr_t H5Lget_info(hid_t loc_id, const char *name,  
<     H5L_info_t *linfo , hid_t lapl_id);  
< H5_DLL herr_t H5Lget_info_by_idx(hid_t loc_id, const char *group_name,  
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,  
<     H5L_info_t *linfo , hid_t lapl_id);  
< H5_DLL ssize_t H5Lget_name_by_idx(hid_t loc_id, const char *group_name,  
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,  
<     char *name , size_t size, hid_t lapl_id);  
< H5_DLL herr_t H5Literate(hid_t grp_id, H5_index_t idx_type,
```

```

<      H5_iter_order_t order, hsize_t *idx, H5L_iterate_t op, void *op_data);
< H5_DLL herr_t H5Literate_by_name(hid_t loc_id, const char *group_name,
<      H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
<      H5L_iterate_t op, void *op_data, hid_t lcpl_id);
< H5_DLL herr_t H5Lvisit(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order,
<      H5L_iterate_t op, void *op_data);
< H5_DLL herr_t H5Lvisit_by_name(hid_t loc_id, const char *group_name,
<      H5_index_t idx_type, H5_iter_order_t order, H5L_iterate_t op,
<      void *op_data, hid_t lcpl_id);
< H5_DLL herr_t H5Lcreate_ud(hid_t link_loc_id, const char *link_name,
<      H5L_type_t link_type, const void *udata, size_t udata_size, hid_t lcpl_id,
<      hid_t lapl_id);
____
> H5_DLL herr_t H5Lget_info2(hid_t loc_id, const char *name, H5L_info2_t *linfo, hid_t lapl_id);
> H5_DLL herr_t H5Lget_info_by_idx2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>      H5_iter_order_t order, hsize_t n, H5L_info2_t *linfo, hid_t lapl_id);
> H5_DLL ssize_t H5Lget_name_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>      H5_iter_order_t order, hsize_t n, char *name , size_t size,
>      hid_t lapl_id);
> H5_DLL herr_t H5Literate2(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
>      H5L_iterate2_t op, void *op_data);
> H5_DLL herr_t H5Literate_by_name2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>      H5_iter_order_t order, hsize_t *idx, H5L_iterate2_t op, void *op_data,
>      hid_t lapl_id);
> H5_DLL herr_t H5Lvisit2(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order, H5L_iterate2_t op,
>      void *op_data);
> H5_DLL herr_t H5Lvisit_by_name2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>      H5_iter_order_t order, H5L_iterate2_t op, void *op_data, hid_t lapl_id);
> H5_DLL herr_t H5Lcreate_ud(hid_t link_loc_id, const char *link_name, H5L_type_t link_type, const void *udata,
>      size_t udata_size, hid_t lcpl_id, hid_t lapl_id);
43,46c38,54
< H5_DLL herr_t H5Lunpack_elink_val(const void *ext_linkval, size_t link_size,
<      unsigned *flags, const char **filename, const char **obj_path );
< H5_DLL herr_t H5Lcreate_external(const char *file_name, const char *obj_name,
<      hid_t link_loc_id, const char *link_name, hid_t lcpl_id, hid_t lapl_id);
____
> H5_DLL herr_t H5Lunpack_elink_val(const void *ext_linkval , size_t link_size, unsigned *flags,
>      const char **filename , const char **obj_path );
> H5_DLL herr_t H5Lcreate_external(const char *file_name, const char *obj_name, hid_t link_loc_id,
>      const char *link_name, hid_t lcpl_id, hid_t lapl_id);
> H5_DLL herr_t H5Lget_info1(hid_t loc_id, const char *name, H5L_info1_t *linfo , hid_t lapl_id);
> H5_DLL herr_t H5Lget_info_by_idx1(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>      H5_iter_order_t order, hsize_t n, H5L_info1_t *linfo ,
>      hid_t lapl_id);

```



```

const char *attr_name,
<
const char *attr_data );
<
< H5_HLDLL herr_t H5LTset_attribute_char( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const char *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_uchar( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const unsigned char *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_short( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const short *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_ushort( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const unsigned short *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_int( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const int *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_uint( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const unsigned int *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_long( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const long *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_long_long( hid_t loc_id,
<
const char *obj_name,
<
const char *attr_name,
<
const long long *buffer,
<
size_t size );
< H5_HLDLL herr_t H5LTset_attribute_ulong( hid_t loc_id,
<
const char *obj_name,

```



```

<                                const char *attr_name,
<                                unsigned int *data );
< H5_HLDLL herr_t  H5LTget_attribute_long( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                long *data );
< H5_HLDLL herr_t  H5LTget_attribute_long_long( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                long long *data );
< H5_HLDLL herr_t  H5LTget_attribute_ulong( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                unsigned long *data );
< H5_HLDLL herr_t  H5LTget_attribute_float( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                float *data );
< H5_HLDLL herr_t  H5LTget_attribute_double( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                double *data );
< H5_HLDLL herr_t  H5LTget_attribute_ndims( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                int *rank );
< H5_HLDLL herr_t  H5LTget_attribute_info( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                hsize_t *dims,
<                                H5T_class_t *type_class,
<                                size_t *type_size );
< H5_HLDLL hid_t H5LTtext_to_dtype(const char *text, H5LT_lang_t lang_type);
_____
> H5_HLDLL herr_t H5LTmake_dataset(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>                                hid_t type_id, const void *buffer);
> H5_HLDLL herr_t H5LTmake_dataset_char(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>                                const char *buffer);
> H5_HLDLL herr_t H5LTmake_dataset_short(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>                                const short *buffer);
> H5_HLDLL herr_t H5LTmake_dataset_int(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>                                const int *buffer);
> H5_HLDLL herr_t H5LTmake_dataset_long(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>                                const long *buffer);

```

```

> H5_HLDLL herr_t H5Lmake_dataset_float(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>
>                                     const float *buffer);
> H5_HLDLL herr_t H5Lmake_dataset_double(hid_t loc_id, const char *dset_name, int rank, const hsize_t *dims,
>
>                                     const double *buffer);
> H5_HLDLL herr_t H5Lmake_dataset_string(hid_t loc_id, const char *dset_name, const char *buf);
> H5_HLDLL herr_t H5LTread_dataset(hid_t loc_id, const char *dset_name, hid_t type_id, void *buffer);
> H5_HLDLL herr_t H5LTread_dataset_char(hid_t loc_id, const char *dset_name, char *buffer);
> H5_HLDLL herr_t H5LTread_dataset_short(hid_t loc_id, const char *dset_name, short *buffer);
> H5_HLDLL herr_t H5LTread_dataset_int(hid_t loc_id, const char *dset_name, int *buffer);
> H5_HLDLL herr_t H5LTread_dataset_long(hid_t loc_id, const char *dset_name, long *buffer);
> H5_HLDLL herr_t H5LTread_dataset_float(hid_t loc_id, const char *dset_name, float *buffer);
> H5_HLDLL herr_t H5LTread_dataset_double(hid_t loc_id, const char *dset_name, double *buffer);
> H5_HLDLL herr_t H5LTread_dataset_string(hid_t loc_id, const char *dset_name, char *buf);
> H5_HLDLL herr_t H5LTget_dataset_ndims(hid_t loc_id, const char *dset_name, int *rank);
> H5_HLDLL herr_t H5LTget_dataset_info(hid_t loc_id, const char *dset_name, hsize_t *dims,
>
>                                     H5T_class_t *type_class, size_t *type_size);
> H5_HLDLL herr_t H5LTfind_dataset(hid_t loc_id, const char *name);
> H5_HLDLL herr_t H5LTset_attribute_string(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const char *attr_data);
> H5_HLDLL herr_t H5LTset_attribute_char(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const char *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_uchar(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned char *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_short(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const short *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_ushort(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned short *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_int(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const int *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_uint(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned int *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_long(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const long *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_long_long(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const long long *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_ulong(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned long *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_float(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const float *buffer, size_t size);
> H5_HLDLL herr_t H5LTset_attribute_double(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const double *buffer, size_t size);
> H5_HLDLL herr_t H5LTget_attribute(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     hid_t mem_type_id, void *data);
> H5_HLDLL herr_t H5LTget_attribute_string(hid_t loc_id, const char *obj_name, const char *attr_name,

```

```

>                                     char *data);
> H5_HLDLL herr_t H5LTget_attribute_char(hid_t loc_id, const char *obj_name, const char *attr_name, char
    *data);
> H5_HLDLL herr_t H5LTget_attribute_uchar(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     unsigned char *data);
> H5_HLDLL herr_t H5LTget_attribute_short(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     short *data);
> H5_HLDLL herr_t H5LTget_attribute_ushort(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     unsigned short *data);
> H5_HLDLL herr_t H5LTget_attribute_int(hid_t loc_id, const char *obj_name, const char *attr_name, int *data);
> H5_HLDLL herr_t H5LTget_attribute_uint(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     unsigned int *data);
> H5_HLDLL herr_t H5LTget_attribute_long(hid_t loc_id, const char *obj_name, const char *attr_name, long
    *data);
> H5_HLDLL herr_t H5LTget_attribute_long_long(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     long long *data);
> H5_HLDLL herr_t H5LTget_attribute_ulong(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     unsigned long *data);
> H5_HLDLL herr_t H5LTget_attribute_float(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     float *data);
> H5_HLDLL herr_t H5LTget_attribute_double(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     double *data);
> H5_HLDLL herr_t H5LTget_attribute_ndims(hid_t loc_id, const char *obj_name, const char *attr_name, int
    *rank);
> H5_HLDLL herr_t H5LTget_attribute_info(hid_t loc_id, const char *obj_name, const char *attr_name,
>                                     hsize_t *dims, H5T_class_t *type_class, size_t *type_size);
> H5_HLDLL hid_t H5LTtext_to_dtype(const char *text, H5LT_lang_t lang_type);
199c81
< H5_HLDLL herr_t H5LTfind_attribute( hid_t loc_id, const char *name );
-----
> H5_HLDLL herr_t H5LTfind_attribute(hid_t loc_id, const char *name);
改变 H5MMpublic.h
2a3
>
增加头文件: H5Mpublic.h
改变 H5Opublic.h
3,5c3,5
< H5_DLL hid_t H5Oopen_by_addr(hid_t loc_id, haddr_t addr);
< H5_DLL hid_t H5Oopen_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t lapl_id);
-----
> H5_DLL hid_t H5Oopen_by_token(hid_t loc_id, H5O_token_t token);
> H5_DLL hid_t H5Oopen_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type, H5_iter_order_t
    order,

```

```

>                                     hsize_t n, hid_t lapl_id);
7,14c7,19
< H5_DLL herr_t H5Oget_info(hid_t loc_id, H5O_info_t *oinfo);
< H5_DLL herr_t H5Oget_info_by_name(hid_t loc_id, const char *name, H5O_info_t *oinfo,
<     hid_t lapl_id);
< H5_DLL herr_t H5Oget_info_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, H5O_info_t *oinfo,
<     hid_t lapl_id);
< H5_DLL herr_t H5Olink(hid_t obj_id, hid_t new_loc_id, const char *new_name,
<     hid_t lcpl_id, hid_t lapl_id);
____

> H5_DLL herr_t H5Oget_info3(hid_t loc_id, H5O_info2_t *oinfo, unsigned fields);
> H5_DLL herr_t H5Oget_info_by_name3(hid_t loc_id, const char *name, H5O_info2_t *oinfo, unsigned fields,
>     hid_t lapl_id);
> H5_DLL herr_t H5Oget_info_by_idx3(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, H5O_info2_t *oinfo, unsigned fields,
>     hid_t lapl_id);
> H5_DLL herr_t H5Oget_native_info(hid_t loc_id, H5O_native_info_t *oinfo, unsigned fields);
> H5_DLL herr_t H5Oget_native_info_by_name(hid_t loc_id, const char *name, H5O_native_info_t *oinfo,
>     unsigned fields, hid_t lapl_id);
> H5_DLL herr_t H5Oget_native_info_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, H5O_native_info_t *oinfo,
>     unsigned fields, hid_t lapl_id);
> H5_DLL herr_t H5Olink(hid_t obj_id, hid_t new_loc_id, const char *new_name, hid_t lcpl_id, hid_t lapl_id);
17,18c22,23
< H5_DLL herr_t H5Ocopy(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id,
<     const char *dst_name, hid_t ocpypl_id, hid_t lcpl_id);
____

> H5_DLL herr_t H5Ocopy(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id, const char *dst_name,
>     hid_t ocpypl_id, hid_t lcpl_id);
20,21c25
< H5_DLL herr_t H5Oset_comment_by_name(hid_t loc_id, const char *name,
<     const char *comment, hid_t lapl_id);
____

> H5_DLL herr_t H5Oset_comment_by_name(hid_t loc_id, const char *name, const char *comment, hid_t lapl_id);
23,29c27,33
< H5_DLL ssize_t H5Oget_comment_by_name(hid_t loc_id, const char *name,
<     char *comment, size_t bufsize, hid_t lapl_id);
< H5_DLL herr_t H5Ovisit(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order,
<     H5O_iterate_t op, void *op_data);
< H5_DLL herr_t H5Ovisit_by_name(hid_t loc_id, const char *obj_name,
<     H5_index_t idx_type, H5_iter_order_t order, H5O_iterate_t op,
<     void *op_data, hid_t lapl_id);
____

```

```

> H5_DLL ssize_t H5Oget_comment_by_name(hid_t loc_id, const char *name, char *comment, size_t bufsize,
>
>                                     hid_t lapl_id);
> H5_DLL herr_t H5Ovisit3(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order, H5O_iterate2_t op,
>
>                                     void *op_data, unsigned fields);
> H5_DLL herr_t H5Ovisit_by_name3(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, H5O_iterate2_t op, void *op_data, unsigned
fields,
>
>                                     hid_t lapl_id);
30a35,64
> H5_DLL herr_t H5Oflush(hid_t obj_id);
> H5_DLL herr_t H5Orefresh(hid_t oid);
> H5_DLL herr_t H5Odisable_mdc_flushes(hid_t object_id);
> H5_DLL herr_t H5Oenable_mdc_flushes(hid_t object_id);
> H5_DLL herr_t H5Oare_mdc_flushes_disabled(hid_t object_id, hbool_t *are_disabled);
> H5_DLL herr_t H5Otoken_cmp(hid_t loc_id, const H5O_token_t *token1, const H5O_token_t *token2,
>
>                                     int *cmp_value);
> H5_DLL herr_t H5Otoken_to_str(hid_t loc_id, const H5O_token_t *token, char **token_str);
> H5_DLL herr_t H5Otoken_from_str(hid_t loc_id, const char *token_str, H5O_token_t *token);
> H5_DLLVAR const H5O_token_t H5O_TOKEN_UNDEF_g;
> H5_DLL hid_t H5Oopen_by_addr(hid_t loc_id, haddr_t addr);
> H5_DLL herr_t H5Oget_info1(hid_t loc_id, H5O_info1_t *oinfo);
> H5_DLL herr_t H5Oget_info_by_name1(hid_t loc_id, const char *name, H5O_info1_t *oinfo, hid_t lapl_id);
> H5_DLL herr_t H5Oget_info_by_idx1(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, H5O_info1_t *oinfo, hid_t lapl_id);
> H5_DLL herr_t H5Oget_info2(hid_t loc_id, H5O_info1_t *oinfo, unsigned fields);
> H5_DLL herr_t H5Oget_info_by_name2(hid_t loc_id, const char *name, H5O_info1_t *oinfo, unsigned fields,
>
>                                     hid_t lapl_id);
> H5_DLL herr_t H5Oget_info_by_idx2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, H5O_info1_t *oinfo, unsigned fields,
>
>                                     hid_t lapl_id);
> H5_DLL herr_t H5Ovisit1(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order, H5O_iterate1_t op,
>
>                                     void *op_data);
> H5_DLL herr_t H5Ovisit_by_name1(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, H5O_iterate1_t op, void *op_data, hid_t lapl_id);
> H5_DLL herr_t H5Ovisit2(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order, H5O_iterate1_t op,
>
>                                     void *op_data, unsigned fields);
> H5_DLL herr_t H5Ovisit_by_name2(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, H5O_iterate1_t op, void *op_data, unsigned
fields,
>
>                                     hid_t lapl_id);
31a66
>
> 改变 H5PLextern.h
4a5

```

>

改变 H5PLpublic.h

2,7c2,7

```
< H5_DLL herr_t H5PLset_loading_state(unsigned int plugin_type);
< H5_DLL herr_t H5PLget_loading_state(unsigned int *plugin_type);
< H5_DLL herr_t H5PLappend(const char *plugin_path);
< H5_DLL herr_t H5PLprepend(const char *plugin_path);
< H5_DLL herr_t H5PLreplace(const char *plugin_path, unsigned int index);
< H5_DLL herr_t H5PLinsert(const char *plugin_path, unsigned int index);
```

```
> H5_DLL herr_t H5PLset_loading_state(unsigned int plugin_control_mask);
> H5_DLL herr_t H5PLget_loading_state(unsigned int *plugin_control_mask );
> H5_DLL herr_t H5PLappend(const char *search_path);
> H5_DLL herr_t H5PLprepend(const char *search_path);
> H5_DLL herr_t H5PLreplace(const char *search_path, unsigned int index);
> H5_DLL herr_t H5PLinsert(const char *search_path, unsigned int index);
```

9,10c9,10

```
< H5_DLL ssize_t H5PLget(unsigned int index, char *pathname, size_t size);
< H5_DLL herr_t H5PLsize(unsigned int *listsize);
```

```
> H5_DLL ssize_t H5PLget(unsigned int index, char *path_buf , size_t buf_size);
> H5_DLL herr_t H5PLsize(unsigned int *num_paths );
```

11a12

>

改变 H5Ppublic.h

13a14,15

```
> H5_DLLVAR hid_t H5P_CLS_MAP_CREATE_ID_g;
> H5_DLLVAR hid_t H5P_CLS_MAP_ACCESS_ID_g;
```

15a18

```
> H5_DLLVAR hid_t H5P_CLS_ATTRIBUTE_ACCESS_ID_g;
```

18a22,23

```
> H5_DLLVAR hid_t H5P_CLS_VOL_INITIALIZE_ID_g;
> H5_DLLVAR hid_t H5P_CLS_REFERENCE_ACCESS_ID_g;
```

28a34,35

```
> H5_DLLVAR hid_t H5P_LST_MAP_CREATE_ID_g;
> H5_DLLVAR hid_t H5P_LST_MAP_ACCESS_ID_g;
```

29a37

```
> H5_DLLVAR hid_t H5P_LST_ATTRIBUTE_ACCESS_ID_g;
```

33,37c41,46

```
< H5_DLL hid_t H5Pcreate_class(hid_t parent, const char *name,
<     H5P_cls_create_func_t cls_create, void *create_data,
<     H5P_cls_copy_func_t cls_copy, void *copy_data,
<     H5P_cls_close_func_t cls_close, void *close_data);
< H5_DLL char *H5Pget_class_name(hid_t pclass_id);
```



```

-----
> H5_DLLVAR hid_t H5P_LST_VOL_INITIALIZE_ID_g;
> H5_DLLVAR hid_t H5P_LST_REFERENCE_ACCESS_ID_g;
> H5_DLL herr_t H5Pclose(hid_t plist_id);
> H5_DLL herr_t H5Pclose_class(hid_t plist_id);
> H5_DLL hid_t H5Pcopy(hid_t plist_id);
> H5_DLL herr_t H5Pcopy_prop(hid_t dst_id, hid_t src_id, const char *name);
39,48c48,53
< H5_DLL herr_t H5Pregister2(hid_t cls_id, const char *name, size_t size,
<     void *def_value, H5P_prp_create_func_t prp_create,
<     H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_compare_func_t prp_cmp, H5P_prp_close_func_t prp_close);
< H5_DLL herr_t H5Pinsert2(hid_t plist_id, const char *name, size_t size,
<     void *value, H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_delete, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_compare_func_t prp_cmp, H5P_prp_close_func_t prp_close);
< H5_DLL herr_t H5Pset(hid_t plist_id, const char *name, void *value);
-----
> H5_DLL hid_t H5Pcreate_class(hid_t parent, const char *name, H5P_cls_create_func_t create, void *create_data,
>
>     H5P_cls_copy_func_t copy, void *copy_data, H5P_cls_close_func_t close,
>
>     void *close_data);
> H5_DLL hid_t H5Pdecode(const void *buf);
> H5_DLL herr_t H5Pencode2(hid_t plist_id, void *buf, size_t *nalloc, hid_t fapl_id);
> H5_DLL htri_t H5Pequal(hid_t id1, hid_t id2);
50,51c55
< H5_DLL herr_t H5Pget_size(hid_t id, const char *name, size_t *size);
< H5_DLL herr_t H5Pget_nprops(hid_t id, size_t *nprops);
-----
> H5_DLL herr_t H5Pget(hid_t plist_id, const char *name, void *value);
52a57
> H5_DLL char *H5Pget_class_name(hid_t pclass_id);
54,55c59,63
< H5_DLL herr_t H5Pget(hid_t plist_id, const char *name, void *value);
< H5_DLL htri_t H5Pequal(hid_t id1, hid_t id2);
-----
> H5_DLL herr_t H5Pget_nprops(hid_t id, size_t *nprops);
> H5_DLL herr_t H5Pget_size(hid_t id, const char *name, size_t *size);
> H5_DLL herr_t H5Pinsert2(hid_t plist_id, const char *name, size_t size, void *value, H5P_prp_set_func_t set,
>
>     H5P_prp_get_func_t get, H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t
copy,
>
>     H5P_prp_compare_func_t compare, H5P_prp_close_func_t close);
57,59c65,69
< H5_DLL int H5Piterate(hid_t id, int *idx, H5P_iterate_t iter_func,

```

```

< void *iter_data);
< H5_DLL herr_t H5Pcopy_prop(hid_t dst_id, hid_t src_id, const char *name);
-----
> H5_DLL int H5Piterate(hid_t id, int *idx, H5P_iterate_t iter_func, void *iter_data);
> H5_DLL herr_t H5Pregister2(hid_t cls_id, const char *name, size_t size, void *def_value,
> H5P_prp_create_func_t create, H5P_prp_set_func_t set, H5P_prp_get_func_t
get,
> H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t copy,
> H5P_prp_compare_func_t compare, H5P_prp_close_func_t close);
60a71
> H5_DLL herr_t H5Pset(hid_t plist_id, const char *name, const void *value);
62,67c73
< H5_DLL herr_t H5Pclose_class(hid_t plist_id);
< H5_DLL herr_t H5Pclose(hid_t plist_id);
< H5_DLL hid_t H5Pcopy(hid_t plist_id);
< H5_DLL herr_t H5Pset_attr_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
< H5_DLL herr_t H5Pget_attr_phase_change(hid_t plist_id, unsigned *max_compact, unsigned *min_dense);
< H5_DLL herr_t H5Pset_attr_creation_order(hid_t plist_id, unsigned crt_order_flags);
-----
> H5_DLL htri_t H5Pall_filters_avail(hid_t plist_id);
69,76c75,81
< H5_DLL herr_t H5Pset_obj_track_times(hid_t plist_id, hbool_t track_times);
< H5_DLL herr_t H5Pget_obj_track_times(hid_t plist_id, hbool_t *track_times);
< H5_DLL herr_t H5Pmodify_filter(hid_t plist_id, H5Z_filter_t filter,
< unsigned int flags, size_t cd_nelmts,
< const unsigned int cd_values[]);
< H5_DLL herr_t H5Pset_filter(hid_t plist_id, H5Z_filter_t filter,
< unsigned int flags, size_t cd_nelmts,
< const unsigned int c_values[]);
-----
> H5_DLL herr_t H5Pget_attr_phase_change(hid_t plist_id, unsigned *max_compact, unsigned *min_dense);
> H5_DLL H5Z_filter_t H5Pget_filter2(hid_t plist_id, unsigned idx, unsigned int *flags ,
> size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
> char name[], unsigned *filter_config );
> H5_DLL herr_t H5Pget_filter_by_id2(hid_t plist_id, H5Z_filter_t filter_id, unsigned int *flags ,
> size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
> char name[] , unsigned *filter_config );
78,88c83,85
< H5_DLL H5Z_filter_t H5Pget_filter2(hid_t plist_id, unsigned filter,
< unsigned int *flags,
< size_t *cd_nelmts,
< unsigned cd_values[],
< size_t namelen, char name[],
< unsigned *filter_config );

```

```

< H5_DLL herr_t H5Pget_filter_by_id2(hid_t plist_id, H5Z_filter_t id,
<     unsigned int *flags, size_t *cd_nelmts,
<     unsigned cd_values[], size_t namelen, char name[],
<     unsigned *filter_config);
< H5_DLL htri_t H5Pall_filters_avail(hid_t plist_id);
____

> H5_DLL herr_t H5Pget_obj_track_times(hid_t plist_id, hbool_t *track_times);
> H5_DLL herr_t H5Pmodify_filter(hid_t plist_id, H5Z_filter_t filter, unsigned int flags, size_t cd_nelmts,
>     const unsigned int cd_values[]);
90c87,91
< H5_DLL herr_t H5Pset_deflate(hid_t plist_id, unsigned aggression);
____

> H5_DLL herr_t H5Pset_attr_creation_order(hid_t plist_id, unsigned crt_order_flags);
> H5_DLL herr_t H5Pset_attr_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
> H5_DLL herr_t H5Pset_deflate(hid_t plist_id, unsigned level);
> H5_DLL herr_t H5Pset_filter(hid_t plist_id, H5Z_filter_t filter, unsigned int flags, size_t cd_nelmts,
>     const unsigned int c_values[]);
92,95c93,103
< H5_DLL herr_t H5Pget_version(hid_t plist_id, unsigned *boot,
<     unsigned *freelist, unsigned *stab,
<     unsigned *shhdr);
< H5_DLL herr_t H5Pset_userblock(hid_t plist_id, hsize_t size);
____

> H5_DLL herr_t H5Pset_obj_track_times(hid_t plist_id, hbool_t track_times);
> H5_DLL herr_t H5Pget_file_space_page_size(hid_t plist_id, hsize_t *fsp_size);
> H5_DLL herr_t H5Pget_file_space_strategy(hid_t plist_id, H5F_fspace_strategy_t *strategy, hbool_t *persist,
>     hsize_t *threshold);
> H5_DLL herr_t H5Pget_istore_k(hid_t plist_id, unsigned *ik );
> H5_DLL herr_t H5Pget_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned *mesg_type_flags,
>     unsigned *min_mesg_size);
> H5_DLL herr_t H5Pget_shared_mesg_nindexes(hid_t plist_id, unsigned *nindexes);
> H5_DLL herr_t H5Pget_shared_mesg_phase_change(hid_t plist_id, unsigned *max_list, unsigned *min_btree);
> H5_DLL herr_t H5Pget_sizes(hid_t plist_id, size_t *sizeof_addr , size_t *sizeof_size );
> H5_DLL herr_t H5Pget_sym_k(hid_t plist_id, unsigned *ik , unsigned *lk );
97,102c105,107
< H5_DLL herr_t H5Pset_sizes(hid_t plist_id, size_t sizeof_addr,
<     size_t sizeof_size);
< H5_DLL herr_t H5Pget_sizes(hid_t plist_id, size_t *sizeof_addr,
<     size_t *sizeof_size);
< H5_DLL herr_t H5Pset_sym_k(hid_t plist_id, unsigned ik, unsigned lk);
< H5_DLL herr_t H5Pget_sym_k(hid_t plist_id, unsigned *ik, unsigned *lk);
____

> H5_DLL herr_t H5Pset_file_space_page_size(hid_t plist_id, hsize_t fsp_size);
> H5_DLL herr_t H5Pset_file_space_strategy(hid_t plist_id, H5F_fspace_strategy_t strategy, hbool_t persist,

```

```

>                                     hsize_t threshold);
104c109,110
< H5_DLL herr_t H5Pget_istore_k(hid_t plist_id, unsigned *ik);
____
> H5_DLL herr_t H5Pset_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned mesg_type_flags,
>                                     unsigned min_mesg_size);
106,108d111
< H5_DLL herr_t H5Pget_shared_mesg_nindexes(hid_t plist_id, unsigned *nindexes);
< H5_DLL herr_t H5Pset_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned mesg_type_flags,
unsigned min_mesg_size);
< H5_DLL herr_t H5Pget_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned *mesg_type_flags,
unsigned *min_mesg_size);
110,116c113,119
< H5_DLL herr_t H5Pget_shared_mesg_phase_change(hid_t plist_id, unsigned *max_list, unsigned *min_btree);
< H5_DLL herr_t H5Pset_alignment(hid_t fapl_id, hsize_t threshold,
<     hsize_t alignment);
< H5_DLL herr_t H5Pget_alignment(hid_t fapl_id, hsize_t *threshold,
<     hsize_t *alignment);
< H5_DLL herr_t H5Pset_driver(hid_t plist_id, hid_t driver_id,
<     const void *driver_info);
____
> H5_DLL herr_t H5Pset_sizes(hid_t plist_id, size_t sizeof_addr, size_t sizeof_size);
> H5_DLL herr_t H5Pset_sym_k(hid_t plist_id, unsigned ik, unsigned lk);
> H5_DLL herr_t H5Pset_userblock(hid_t plist_id, hsize_t size);
> H5_DLL herr_t H5Pget_alignment(hid_t fapl_id, hsize_t *threshold , hsize_t *alignment );
> H5_DLL herr_t H5Pget_cache(hid_t plist_id, int *mdc_nelmts,
>                                     size_t *rdcc_nslots , size_t *rdcc_nbytes , double *rdcc_w0);
> H5_DLL herr_t H5Pget_core_write_tracking(hid_t fapl_id, hbool_t *is_enabled, size_t *page_size);
118,119c121,123
< H5_DLL void *H5Pget_driver_info(hid_t plist_id);
< H5_DLL herr_t H5Pset_family_offset(hid_t fapl_id, hsize_t offset);
____
> H5_DLL const void *H5Pget_driver_info(hid_t plist_id);
> H5_DLL herr_t H5Pget_elink_file_cache_size(hid_t plist_id, unsigned *efc_size);
> H5_DLL herr_t H5Pget_evict_on_close(hid_t fapl_id, hbool_t *evict_on_close);
121c125,136
< H5_DLL herr_t H5Pset_multi_type(hid_t fapl_id, H5FD_mem_t type);
____
> H5_DLL herr_t H5Pget_fclose_degree(hid_t fapl_id, H5F_close_degree_t *degree);
> H5_DLL herr_t H5Pget_file_image(hid_t fapl_id, void **buf_ptr_ptr, size_t *buf_len_ptr);
> H5_DLL herr_t H5Pget_file_image_callbacks(hid_t fapl_id, H5FD_file_image_callbacks_t *callbacks_ptr);
> H5_DLL herr_t H5Pget_file_locking(hid_t fapl_id, hbool_t *use_file_locking, hbool_t *ignore_when_disabled);
> H5_DLL herr_t H5Pget_gc_references(hid_t fapl_id, unsigned *gc_ref );
> H5_DLL herr_t H5Pget_libver_bounds(hid_t plist_id, H5F_libver_t *low, H5F_libver_t *high);

```

```

> H5_DLL herr_t H5Pget_mdc_config(hid_t plist_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Pget_mdc_image_config(hid_t plist_id, H5AC_cache_image_config_t *config_ptr);
> H5_DLL herr_t H5Pget_mdc_log_options(hid_t plist_id, hbool_t *is_enabled, char *location,
>                                     size_t *location_size, hbool_t *start_on_access);
> H5_DLL herr_t H5Pget_meta_block_size(hid_t fapl_id, hsize_t *size);
> H5_DLL herr_t H5Pget_metadata_read_attempts(hid_t plist_id, unsigned *attempts);
123,135c138,152
< H5_DLL herr_t H5Pset_cache(hid_t plist_id, int mdc_nelmts,
<     size_t rdcc_nslots, size_t rdcc_nbytes,
<     double rdcc_w0);
< H5_DLL herr_t H5Pget_cache(hid_t plist_id,
<     int *mdc_nelmts,
<     size_t *rdcc_nslots,
<     size_t *rdcc_nbytes, double *rdcc_w0);
< H5_DLL herr_t H5Pset_mdc_config(hid_t      plist_id,
<     H5AC_cache_config_t * config_ptr);
< H5_DLL herr_t H5Pget_mdc_config(hid_t      plist_id,
<     H5AC_cache_config_t * config_ptr);
< H5_DLL herr_t H5Pset_gc_references(hid_t fapl_id, unsigned gc_ref);
< H5_DLL herr_t H5Pget_gc_references(hid_t fapl_id, unsigned *gc_ref);
---
> H5_DLL herr_t H5Pget_object_flush_cb(hid_t plist_id, H5F_flush_cb_t *func, void **udata);
> H5_DLL herr_t H5Pget_page_buffer_size(hid_t plist_id, size_t *buf_size, unsigned *min_meta_perc,
>                                     unsigned *min_raw_perc);
> H5_DLL herr_t H5Pget_sieve_buf_size(hid_t fapl_id, size_t *size);
> H5_DLL herr_t H5Pget_small_data_block_size(hid_t fapl_id, hsize_t *size);
> H5_DLL herr_t H5Pget_vol_id(hid_t plist_id, hid_t *vol_id);
> H5_DLL herr_t H5Pget_vol_info(hid_t plist_id, void **vol_info);
> H5_DLL herr_t H5Pset_alignment(hid_t fapl_id, hsize_t threshold, hsize_t alignment);
> H5_DLL herr_t H5Pset_cache(hid_t plist_id, int mdc_nelmts, size_t rdcc_nslots, size_t rdcc_nbytes,
>                             double rdcc_w0);
> H5_DLL herr_t H5Pset_core_write_tracking(hid_t fapl_id, hbool_t is_enabled, size_t page_size);
> H5_DLL herr_t H5Pset_driver(hid_t plist_id, hid_t driver_id, const void *driver_info);
> H5_DLL herr_t H5Pset_elink_file_cache_size(hid_t plist_id, unsigned efc_size);
> H5_DLL herr_t H5Pset_evict_on_close(hid_t fapl_id, hbool_t evict_on_close);
> H5_DLL herr_t H5Pset_family_offset(hid_t fapl_id, hsize_t offset);
137c154,161
< H5_DLL herr_t H5Pget_fc_close_degree(hid_t fapl_id, H5F_close_degree_t *degree);
---
> H5_DLL herr_t H5Pset_file_image(hid_t fapl_id, void *buf_ptr, size_t buf_len);
> H5_DLL herr_t H5Pset_file_image_callbacks(hid_t fapl_id, H5FD_file_image_callbacks_t *callbacks_ptr);
> H5_DLL herr_t H5Pset_file_locking(hid_t fapl_id, hbool_t use_file_locking, hbool_t ignore_when_disabled);
> H5_DLL herr_t H5Pset_gc_references(hid_t fapl_id, unsigned gc_ref);
> H5_DLL herr_t H5Pset_libver_bounds(hid_t plist_id, H5F_libver_t low, H5F_libver_t high);

```

```

> H5_DLL herr_t H5Pset_mdc_config(hid_t plist_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Pset_mdc_log_options(hid_t plist_id, hbool_t is_enabled, const char *location,
>                                     hbool_t start_on_access);
139c163,165
< H5_DLL herr_t H5Pget_meta_block_size(hid_t fapl_id, hsize_t *size);
____

> H5_DLL herr_t H5Pset_metadata_read_attempts(hid_t plist_id, unsigned attempts);
> H5_DLL herr_t H5Pset_multi_type(hid_t fapl_id, H5FD_mem_t type);
> H5_DLL herr_t H5Pset_object_flush_cb(hid_t plist_id, H5F_flush_cb_t func, void *udata);
141d166
< H5_DLL herr_t H5Pget_sieve_buf_size(hid_t fapl_id, size_t *size);
143,158c168,187
< H5_DLL herr_t H5Pget_small_data_block_size(hid_t fapl_id, hsize_t *size);
< H5_DLL herr_t H5Pset_libver_bounds(hid_t plist_id, H5F_libver_t low,
<     H5F_libver_t high);
< H5_DLL herr_t H5Pget_libver_bounds(hid_t plist_id, H5F_libver_t *low,
<     H5F_libver_t *high);
< H5_DLL herr_t H5Pset_elink_file_cache_size(hid_t plist_id, unsigned efc_size);
< H5_DLL herr_t H5Pget_elink_file_cache_size(hid_t plist_id, unsigned *efc_size);
< H5_DLL herr_t H5Pset_file_image(hid_t fapl_id, void *buf_ptr, size_t buf_len);
< H5_DLL herr_t H5Pget_file_image(hid_t fapl_id, void **buf_ptr_ptr, size_t *buf_len_ptr);
< H5_DLL herr_t H5Pset_file_image_callbacks(hid_t fapl_id,
<     H5FD_file_image_callbacks_t *callbacks_ptr);
< H5_DLL herr_t H5Pget_file_image_callbacks(hid_t fapl_id,
<     H5FD_file_image_callbacks_t *callbacks_ptr);
< H5_DLL herr_t H5Pset_core_write_tracking(hid_t fapl_id, hbool_t is_enabled, size_t page_size);
< H5_DLL herr_t H5Pget_core_write_tracking(hid_t fapl_id, hbool_t *is_enabled, size_t *page_size);
< H5_DLL herr_t H5Pset_layout(hid_t plist_id, H5D_layout_t layout);
____

> H5_DLL herr_t H5Pset_vol(hid_t plist_id, hid_t new_vol_id, const void *new_vol_info);
> H5_DLL herr_t H5Pset_all_coll_metadata_ops(hid_t plist_id, hbool_t is_collective);
> H5_DLL herr_t H5Pget_all_coll_metadata_ops(hid_t plist_id, hbool_t *is_collective);
> H5_DLL herr_t H5Pset_coll_metadata_write(hid_t plist_id, hbool_t is_collective);
> H5_DLL herr_t H5Pget_coll_metadata_write(hid_t plist_id, hbool_t *is_collective);
> H5_DLL herr_t H5Pget_mpi_params(hid_t fapl_id, MPI_Comm *comm, MPI_Info *info);
> H5_DLL herr_t H5Pset_mpi_params(hid_t fapl_id, MPI_Comm comm, MPI_Info info);
> H5_DLL herr_t H5Pset_mdc_image_config(hid_t plist_id, H5AC_cache_image_config_t *config_ptr);
> H5_DLL herr_t H5Pset_page_buffer_size(hid_t plist_id, size_t buf_size, unsigned min_meta_per,
>                                     unsigned min_raw_per);
> H5_DLL herr_t H5Pfill_value_defined(hid_t plist, H5D_fill_value_t *status);
> H5_DLL herr_t H5Pget_alloc_time(hid_t plist_id, H5D_alloc_time_t *alloc_time);
> H5_DLL int H5Pget_chunk(hid_t plist_id, int max_ndims, hsize_t dim[]);
> H5_DLL herr_t H5Pget_chunk_opts(hid_t plist_id, unsigned *opts);
> H5_DLL herr_t H5Pget_dset_no_attrs_hint(hid_t dcpl_id, hbool_t *minimize);

```

```
> H5_DLL herr_t H5Pget_external(hid_t plist_id, unsigned idx, size_t name_size, char *name ,
>                               off_t *offset , hsize_t *size );
> H5_DLL int H5Pget_external_count(hid_t plist_id);
> H5_DLL herr_t H5Pget_fill_time(hid_t plist_id, H5D_fill_time_t *fill_time );
> H5_DLL herr_t H5Pget_fill_value(hid_t plist_id, hid_t type_id, void *value );
159a189,194
> H5_DLL herr_t H5Pget_virtual_count(hid_t dcpl_id, size_t *count );
> H5_DLL ssize_t H5Pget_virtual_dsetname(hid_t dcpl_id, size_t index, char *name , size_t size);
> H5_DLL ssize_t H5Pget_virtual_filename(hid_t dcpl_id, size_t index, char *name , size_t size);
> H5_DLL hid_t H5Pget_virtual_srcspace(hid_t dcpl_id, size_t index);
> H5_DLL hid_t H5Pget_virtual_vspace(hid_t dcpl_id, size_t index);
> H5_DLL herr_t H5Pset_alloc_time(hid_t plist_id, H5D_alloc_time_t alloc_time);
161,168c196,200
< H5_DLL int H5Pget_chunk(hid_t plist_id, int max_ndims, hsize_t dim[]);
< H5_DLL herr_t H5Pset_external(hid_t plist_id, const char *name, off_t offset,
<                               hsize_t size);
< H5_DLL int H5Pget_external_count(hid_t plist_id);
< H5_DLL herr_t H5Pget_external(hid_t plist_id, unsigned idx, size_t name_size,
<                               char *name, off_t *offset,
<                               hsize_t *size);
< H5_DLL herr_t H5Pset_szip(hid_t plist_id, unsigned options_mask, unsigned pixels_per_block);
----
> H5_DLL herr_t H5Pset_chunk_opts(hid_t plist_id, unsigned opts);
> H5_DLL herr_t H5Pset_dset_no_attr_hint(hid_t dcpl_id, hbool_t minimize);
> H5_DLL herr_t H5Pset_external(hid_t plist_id, const char *name, off_t offset, hsize_t size);
> H5_DLL herr_t H5Pset_fill_time(hid_t plist_id, H5D_fill_time_t fill_time);
> H5_DLL herr_t H5Pset_fill_value(hid_t plist_id, hid_t type_id, const void *value);
169a202
> H5_DLL herr_t H5Pset_layout(hid_t plist_id, H5D_layout_t layout);
172,198c205,228
< H5_DLL herr_t H5Pset_fill_value(hid_t plist_id, hid_t type_id,
<                               const void *value);
< H5_DLL herr_t H5Pget_fill_value(hid_t plist_id, hid_t type_id,
<                               void *value);
< H5_DLL herr_t H5Pfill_value_defined(hid_t plist, H5D_fill_value_t *status);
< H5_DLL herr_t H5Pset_alloc_time(hid_t plist_id, H5D_alloc_time_t
<   alloc_time);
< H5_DLL herr_t H5Pget_alloc_time(hid_t plist_id, H5D_alloc_time_t
<   *alloc_time);
< H5_DLL herr_t H5Pset_fill_time(hid_t plist_id, H5D_fill_time_t fill_time);
< H5_DLL herr_t H5Pget_fill_time(hid_t plist_id, H5D_fill_time_t
<   *fill_time);
< H5_DLL herr_t H5Pset_chunk_cache(hid_t dapl_id, size_t rdcc_nslots,
<   size_t rdcc_nbytes, double rdcc_w0);
```

```

< H5_DLL herr_t H5Pget_chunk_cache(hid_t dapl_id,
<     size_t *rdcc_nslots,
<     size_t *rdcc_nbytes,
<     double *rdcc_w0);
< H5_DLL herr_t H5Pset_efile_prefix(hid_t dapl_id, const char* prefix);
< H5_DLL ssize_t H5Pget_efile_prefix(hid_t dapl_id, char* prefix , size_t size);
< H5_DLL herr_t H5Pset_data_transform(hid_t plist_id, const char* expression);
< H5_DLL ssize_t H5Pget_data_transform(hid_t plist_id, char* expression , size_t size);
< H5_DLL herr_t H5Pset_buffer(hid_t plist_id, size_t size, void *tconv,
<     void *bkg);
< H5_DLL size_t H5Pget_buffer(hid_t plist_id, void **tconv,
<     void **bkg);
< H5_DLL herr_t H5Pset_preserve(hid_t plist_id, hbool_t status);
____
> H5_DLL herr_t H5Pset_szip(hid_t plist_id, unsigned options_mask, unsigned pixels_per_block);
> H5_DLL herr_t H5Pset_virtual(hid_t dcpl_id, hid_t vspace_id, const char *src_file_name,
>     const char *src_dset_name, hid_t src_space_id);
> H5_DLL herr_t H5Pget_append_flush(hid_t dapl_id, unsigned dims, hsize_t boundary[], H5D_append_cb_t *func,
>     void **udata);
> H5_DLL herr_t H5Pget_chunk_cache(hid_t dapl_id, size_t *rdcc_nslots , size_t *rdcc_nbytes ,
>     double *rdcc_w0 );
> H5_DLL ssize_t H5Pget_efile_prefix(hid_t dapl_id, char *prefix , size_t size);
> H5_DLL ssize_t H5Pget_virtual_prefix(hid_t dapl_id, char *prefix , size_t size);
> H5_DLL herr_t H5Pget_virtual_printf_gap(hid_t dapl_id, hsize_t *gap_size);
> H5_DLL herr_t H5Pget_virtual_view(hid_t dapl_id, H5D_vds_view_t *view);
> H5_DLL herr_t H5Pset_append_flush(hid_t dapl_id, unsigned ndims, const hsize_t boundary[],
>     H5D_append_cb_t func, void *udata);
> H5_DLL herr_t H5Pset_chunk_cache(hid_t dapl_id, size_t rdcc_nslots, size_t rdcc_nbytes, double rdcc_w0);
> H5_DLL herr_t H5Pset_efile_prefix(hid_t dapl_id, const char *prefix);
> H5_DLL herr_t H5Pset_virtual_prefix(hid_t dapl_id, const char *prefix);
> H5_DLL herr_t H5Pset_virtual_printf_gap(hid_t dapl_id, hsize_t gap_size);
> H5_DLL herr_t H5Pset_virtual_view(hid_t dapl_id, H5D_vds_view_t view);
> H5_DLL herr_t H5Pget_btree_ratios(hid_t plist_id, double *left , double *middle ,
>     double *right );
> H5_DLL size_t H5Pget_buffer(hid_t plist_id, void **tconv , void **bkg );
> H5_DLL ssize_t H5Pget_data_transform(hid_t plist_id, char *expression , size_t size);
> H5_DLL H5Z_EDC_t H5Pget_edc_check(hid_t plist_id);
> H5_DLL herr_t H5Pget_hyper_vector_size(hid_t fapl_id, size_t *size );
199a230,235
> H5_DLL herr_t H5Pget_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t *op, void **operate_data);
> H5_DLL herr_t H5Pget_vlen_mem_manager(hid_t plist_id, H5MM_allocate_t *alloc_func, void **alloc_info,
>     H5MM_free_t *free_func, void **free_info);
> H5_DLL herr_t H5Pset_btree_ratios(hid_t plist_id, double left, double middle, double right);
> H5_DLL herr_t H5Pset_buffer(hid_t plist_id, size_t size, void *tconv, void *bkg);

```



```

> H5_DLL herr_t H5Pset_data_transform(hid_t plist_id, const char *expression);
201,222c237,244
< H5_DLL H5Z_EDC_t H5Pget_edc_check(hid_t plist_id);
< H5_DLL herr_t H5Pset_filter_callback(hid_t plist_id, H5Z_filter_func_t func,
<
< void* op_data);
< H5_DLL herr_t H5Pset_btree_ratios(hid_t plist_id, double left, double middle,
<
< double right);
< H5_DLL herr_t H5Pget_btree_ratios(hid_t plist_id, double *left,
<
< double *middle,
<
< double *right);
< H5_DLL herr_t H5Pset_vlen_mem_manager(hid_t plist_id,
<
< H5MM_allocate_t alloc_func,
<
< void *alloc_info, H5MM_free_t free_func,
<
< void *free_info);
< H5_DLL herr_t H5Pget_vlen_mem_manager(hid_t plist_id,
<
< H5MM_allocate_t *alloc_func,
<
< void **alloc_info,
<
< H5MM_free_t *free_func,
<
< void **free_info);
< H5_DLL herr_t H5Pset_hyper_vector_size(hid_t fapl_id, size_t size);
< H5_DLL herr_t H5Pget_hyper_vector_size(hid_t fapl_id, size_t *size);
< H5_DLL herr_t H5Pset_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t op, void* operate_data);
< H5_DLL herr_t H5Pget_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t *op, void** operate_data);
< H5_DLL herr_t H5Pget_mpio_actual_chunk_opt_mode(hid_t plist_id, H5D_mpio_actual_chunk_opt_mode_t
*actual_chunk_opt_mode);
---
> H5_DLL herr_t H5Pset_filter_callback(hid_t plist_id, H5Z_filter_func_t func, void *op_data);
> H5_DLL herr_t H5Pset_hyper_vector_size(hid_t plist_id, size_t size);
> H5_DLL herr_t H5Pset_preserve(hid_t plist_id, hbool_t status);
> H5_DLL herr_t H5Pset_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t op, void *operate_data);
> H5_DLL herr_t H5Pset_vlen_mem_manager(hid_t plist_id, H5MM_allocate_t alloc_func, void *alloc_info,
>
> H5MM_free_t free_func, void *free_info);
> H5_DLL herr_t H5Pget_mpio_actual_chunk_opt_mode(hid_t
plist_id,
>
H5D_mpio_actual_chunk_opt_mode_t
*actual_chunk_opt_mode);
224,225c246,247
< H5_DLL herr_t H5Pget_mpio_no_collective_cause(hid_t plist_id, uint32_t *local_no_collective_cause, uint32_t
*global_no_collective_cause);
< H5_DLL herr_t H5Pset_create_intermediate_group(hid_t plist_id, unsigned crt_intmd);
---
> H5_DLL herr_t H5Pget_mpio_no_collective_cause(hid_t plist_id, uint32_t *local_no_collective_cause,
>
uint32_t *global_no_collective_cause);
227c249,254
< H5_DLL herr_t H5Pset_local_heap_size_hint(hid_t plist_id, size_t size_hint);

```

```

-----
> H5_DLL herr_t H5Pset_create_intermediate_group(hid_t plist_id, unsigned crt_intmd);
> H5_DLL herr_t H5Pget_est_link_info(hid_t plist_id, unsigned *est_num_entries ,
>
>                                     unsigned *est_name_len );
> H5_DLL herr_t H5Pget_link_creation_order(hid_t plist_id, unsigned *crt_order_flags );
> H5_DLL herr_t H5Pget_link_phase_change(hid_t plist_id, unsigned *max_compact ,
>
>                                     unsigned *min_dense );
229,230d255
< H5_DLL herr_t H5Pset_link_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
< H5_DLL herr_t H5Pget_link_phase_change(hid_t plist_id, unsigned *max_compact , unsigned *min_dense );
232d256
< H5_DLL herr_t H5Pget_est_link_info(hid_t plist_id, unsigned *est_num_entries , unsigned *est_name_len );
234,235c258,262
< H5_DLL herr_t H5Pget_link_creation_order(hid_t plist_id, unsigned *crt_order_flags );
< H5_DLL herr_t H5Pset_char_encoding(hid_t plist_id, H5T_cset_t encoding);
-----
> H5_DLL herr_t H5Pset_link_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
> H5_DLL herr_t H5Pset_local_heap_size_hint(hid_t plist_id, size_t size_hint);
> H5_DLL herr_t H5Pset_map_iterate_hints(hid_t mapl_id, size_t key_prefetch_size, size_t key_alloc_size);
> H5_DLL herr_t H5Pget_map_iterate_hints(hid_t mapl_id, size_t *key_prefetch_size ,
>
>                                     size_t *key_alloc_size );
237,240c264,266
< H5_DLL herr_t H5Pset_nlinks(hid_t plist_id, size_t nlinks);
< H5_DLL herr_t H5Pget_nlinks(hid_t plist_id, size_t *nlinks);
< H5_DLL herr_t H5Pset_elink_prefix(hid_t plist_id, const char *prefix);
< H5_DLL ssize_t H5Pget_elink_prefix(hid_t plist_id, char *prefix, size_t size);
-----
> H5_DLL herr_t H5Pset_char_encoding(hid_t plist_id, H5T_cset_t encoding);
> H5_DLL herr_t H5Pget_elink_acc_flags(hid_t lapl_id, unsigned *flags);
> H5_DLL herr_t H5Pget_elink_cb(hid_t lapl_id, H5L_elink_traverse_t *func, void **op_data);
242c268,269
< H5_DLL herr_t H5Pset_elink_fapl(hid_t lapl_id, hid_t fapl_id);
-----
> H5_DLL ssize_t H5Pget_elink_prefix(hid_t plist_id, char *prefix, size_t size);
> H5_DLL herr_t H5Pget_nlinks(hid_t plist_id, size_t *nlinks);
244d270
< H5_DLL herr_t H5Pget_elink_acc_flags(hid_t lapl_id, unsigned *flags);
246,248c272,274
< H5_DLL herr_t H5Pget_elink_cb(hid_t lapl_id, H5L_elink_traverse_t *func, void **op_data);
< H5_DLL herr_t H5Pset_copy_object(hid_t plist_id, unsigned crt_intmd);
< H5_DLL herr_t H5Pget_copy_object(hid_t plist_id, unsigned *crt_intmd );
-----
> H5_DLL herr_t H5Pset_elink_fapl(hid_t lapl_id, hid_t fapl_id);
> H5_DLL herr_t H5Pset_elink_prefix(hid_t plist_id, const char *prefix);

```

```

> H5_DLL herr_t H5Pset_nlinks(hid_t plist_id, size_t nlinks);
251c277
< H5_DLL herr_t H5Pset_mcdt_search_cb(hid_t plist_id, H5O_mcdt_search_cb_t func, void *op_data);
____
> H5_DLL herr_t H5Pget_copy_object(hid_t plist_id, unsigned *copy_options );
253,267c279,299
< H5_DLL herr_t H5Pregister1(hid_t cls_id, const char *name, size_t size,
<     void *def_value, H5P_prp_create_func_t prp_create,
<     H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_close_func_t prp_close);
< H5_DLL herr_t H5Pinsert1(hid_t plist_id, const char *name, size_t size,
<     void *value, H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_delete, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_close_func_t prp_close);
< H5_DLL H5Z_filter_t H5Pget_filter1(hid_t plist_id, unsigned filter,
<     unsigned int *flags, size_t *cd_nelmts,
<     unsigned cd_values[], size_t namelen, char name[]);
< H5_DLL herr_t H5Pget_filter_by_id1(hid_t plist_id, H5Z_filter_t id,
<     unsigned int *flags, size_t *cd_nelmts,
<     unsigned cd_values[], size_t namelen, char name[]);
____
> H5_DLL herr_t H5Pset_copy_object(hid_t plist_id, unsigned copy_options);
> H5_DLL herr_t H5Pset_mcdt_search_cb(hid_t plist_id, H5O_mcdt_search_cb_t func, void *op_data);
> H5_DLL herr_t H5Pregister1(hid_t cls_id, const char *name, size_t size, void *def_value,
>     H5P_prp_create_func_t prp_create, H5P_prp_set_func_t prp_set,
>     H5P_prp_get_func_t prp_get, H5P_prp_delete_func_t prp_del,
>     H5P_prp_copy_func_t prp_copy, H5P_prp_close_func_t prp_close);
> H5_DLL herr_t H5Pinsert1(hid_t plist_id, const char *name, size_t size, void *value,
>     H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
>     H5P_prp_delete_func_t prp_delete, H5P_prp_copy_func_t prp_copy,
>     H5P_prp_close_func_t prp_close);
> H5_DLL herr_t H5Pencode1(hid_t plist_id, void *buf, size_t *nalloc);
> H5_DLL H5Z_filter_t H5Pget_filter1(hid_t plist_id, unsigned filter, unsigned int *flags ,
>     size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
>     char name[]);
> H5_DLL herr_t H5Pget_filter_by_id1(hid_t plist_id, H5Z_filter_t id, unsigned int *flags ,
>     size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
>     char name[] );
> H5_DLL herr_t H5Pget_version(hid_t plist_id, unsigned *boot , unsigned *freelist ,
>     unsigned *stab , unsigned *shhdr );
> H5_DLL herr_t H5Pset_file_space(hid_t plist_id, H5F_file_space_type_t strategy, hsize_t threshold);
> H5_DLL herr_t H5Pget_file_space(hid_t plist_id, H5F_file_space_type_t *strategy, hsize_t *threshold);
268a301

```

```

>
改变 H5PTpublic.h
2,3c2,3
< H5_HLDLL hid_t H5PTcreate(hid_t loc_id, const char *dset_name,
<         hid_t dtype_id, hsize_t chunk_size, hid_t plist_id);
____
> H5_HLDLL hid_t H5PTcreate(hid_t loc_id, const char *dset_name, hid_t dtype_id, hsize_t chunk_size,
>         hid_t plist_id);
6,7c6,7
< H5_HLDLL hid_t H5PTcreate_fl(hid_t loc_id, const char *dset_name,
<         hid_t dtype_id, hsize_t chunk_size, int compression);
____
> H5_HLDLL hid_t H5PTcreate_fl(hid_t loc_id, const char *dset_name, hid_t dtype_id, hsize_t chunk_size,
>         int compression);
9,11c9,10
< H5_HLDLL herr_t H5PTget_next(hid_t table_id, size_t nrecords, void * data);
< H5_HLDLL herr_t H5PTread_packets(hid_t table_id, hsize_t start,
<         size_t nrecords, void *data);
____
> H5_HLDLL herr_t H5PTget_next(hid_t table_id, size_t nrecords, void *data);
> H5_HLDLL herr_t H5PTread_packets(hid_t table_id, hsize_t start, size_t nrecords, void *data);
17,24c16,19
< H5_HLDLL herr_t  H5PTcreate_index( hid_t table_id );
< H5_HLDLL herr_t  H5PTset_index( hid_t table_id,
<         hsize_t pt_index );
< H5_HLDLL herr_t  H5PTget_index( hid_t table_id,
<         hsize_t *pt_index );
< H5_HLDLL herr_t  H5PTfree_vlen_buff( hid_t table_id,
<         size_t buflen,
<         void * buff );
____
> H5_HLDLL herr_t H5PTcreate_index(hid_t table_id);
> H5_HLDLL herr_t H5PTset_index(hid_t table_id, hsize_t pt_index);
> H5_HLDLL herr_t H5PTget_index(hid_t table_id, hsize_t *pt_index);
> H5_HLDLL herr_t H5PTfree_vlen_buff(hid_t table_id, size_t buflen, void *buff);
改变 H5public.h
2,4c2,4
<         H5_GCC_DIAG_OFF(long-long)
< H5_GCC_DIAG_ON(long-long)
<
H5_DLL herr_t H5open(void);
____
>  H5_GCC_DIAG_OFF("long-long")
> H5_GCC_DIAG_ON("long-long")

```

```

> H5_DLL herr_t H5open(void);
8,14c8,13
< H5_DLL herr_t H5set_free_list_limits (int reg_global_lim, int reg_list_lim,
<
int arr_global_lim, int arr_list_lim, int blk_global_lim,
<
int blk_list_lim);
< H5_DLL herr_t H5get_libversion(unsigned *majnum, unsigned *minnum,
<
unsigned *relnum);
< H5_DLL herr_t H5check_version(unsigned majnum, unsigned minnum,
<
unsigned relnum);
——
> H5_DLL herr_t H5set_free_list_limits(int reg_global_lim, int reg_list_lim, int arr_global_lim,
>
int arr_list_lim, int blk_global_lim, int blk_list_lim);
> H5_DLL herr_t H5get_free_list_sizes(size_t *reg_size, size_t *arr_size, size_t *blk_size, size_t *fac_size);
> H5_DLL herr_t H5get_alloc_stats(H5_alloc_stats_t *stats);
> H5_DLL herr_t H5get_libversion(unsigned *majnum, unsigned *minnum, unsigned *relnum);
> H5_DLL herr_t H5check_version(unsigned majnum, unsigned minnum, unsigned relnum);
19a19
>
改变 H5Rpublic.h
1d0
< typedef haddr_t hobj_ref_t;
3,5c2,22
< H5_DLL herr_t H5Rcreate(void *ref, hid_t loc_id, const char *name,
<
H5R_type_t ref_type, hid_t space_id);
< H5_DLL hid_t H5Rdreference(hid_t dataset, H5R_type_t ref_type, const void *ref);
——
> H5_DLL herr_t H5Rcreate_object(hid_t loc_id, const char *name, hid_t oapl_id, H5R_ref_t *ref_ptr);
> H5_DLL herr_t H5Rcreate_region(hid_t loc_id, const char *name, hid_t space_id, hid_t oapl_id,
>
H5R_ref_t *ref_ptr);
> H5_DLL herr_t H5Rcreate_attr(hid_t loc_id, const char *name, const char *attr_name, hid_t oapl_id,
>
H5R_ref_t *ref_ptr);
> H5_DLL herr_t H5Rdestroy(H5R_ref_t *ref_ptr);
> H5_DLL H5R_type_t H5Rget_type(const H5R_ref_t *ref_ptr);
> H5_DLL htri_t H5Requal(const H5R_ref_t *ref1_ptr, const H5R_ref_t *ref2_ptr);
> H5_DLL herr_t H5Rcopy(const H5R_ref_t *src_ref_ptr, H5R_ref_t *dst_ref_ptr);
> H5_DLL hid_t H5Ropen_object(H5R_ref_t *ref_ptr, hid_t rapl_id, hid_t oapl_id);
> H5_DLL hid_t H5Ropen_region(H5R_ref_t *ref_ptr, hid_t rapl_id, hid_t oapl_id);
> H5_DLL hid_t H5Ropen_attr(H5R_ref_t *ref_ptr, hid_t rapl_id, hid_t aapl_id);
> H5_DLL herr_t H5Rget_obj_type3(H5R_ref_t *ref_ptr, hid_t rapl_id, H5O_type_t *obj_type);
> H5_DLL ssize_t H5Rget_file_name(const H5R_ref_t *ref_ptr, char *name, size_t size);
> H5_DLL ssize_t H5Rget_obj_name(H5R_ref_t *ref_ptr, hid_t rapl_id, char *name, size_t size);
> H5_DLL ssize_t H5Rget_attr_name(const H5R_ref_t *ref_ptr, char *name, size_t size);
> H5_DLL H5G_obj_t H5Rget_obj_type1(hid_t id, H5R_type_t ref_type, const void *ref);
> H5_DLL hid_t H5Rdreference1(hid_t obj_id, H5R_type_t ref_type, const void *ref);

```

```

> H5_DLL herr_t H5Rcreate(void *ref, hid_t loc_id, const char *name, H5R_type_t ref_type, hid_t space_id);
> H5_DLL herr_t H5Rget_obj_type2(hid_t id, H5R_type_t ref_type, const void *ref, H5O_type_t *obj_type);
> H5_DLL hid_t H5Rdereference2(hid_t obj_id, hid_t oapl_id, H5R_type_t ref_type, const void *ref);
7,11c24
< H5_DLL herr_t H5Rget_obj_type2(hid_t id, H5R_type_t ref_type, const void *_ref,
<     H5O_type_t *obj_type);
< H5_DLL ssize_t H5Rget_name(hid_t loc_id, H5R_type_t ref_type, const void *ref,
<     char *name, size_t size);
< H5_DLL H5G_obj_t H5Rget_obj_type1(hid_t id, H5R_type_t ref_type, const void *_ref);
----
> H5_DLL ssize_t H5Rget_name(hid_t loc_id, H5R_type_t ref_type, const void *ref, char *name, size_t size);
12a26
>
改变 H5Spublic.h
2,8d1
< H5_DLL hid_t H5Screate(H5S_class_t type);
< H5_DLL hid_t H5Screate_simple(int rank, const hsize_t dims[],
<     const hsize_t maxdims[]);
< H5_DLL herr_t H5Sset_extent_simple(hid_t space_id, int rank,
<     const hsize_t dims[],
<     const hsize_t max[]);
< H5_DLL hid_t H5Scopy(hid_t space_id);
10c3,8
< H5_DLL herr_t H5Sencode(hid_t obj_id, void *buf, size_t *nalloc);
----
> H5_DLL hid_t H5Scombine_hyperslab(hid_t space_id, H5S_seloper_t op, const hsize_t start[],
>     const hsize_t stride[], const hsize_t count[], const hsize_t block[]);
> H5_DLL hid_t H5Scombine_select(hid_t space1_id, H5S_seloper_t op, hid_t space2_id);
> H5_DLL hid_t H5Scopy(hid_t space_id);
> H5_DLL hid_t H5Screate(H5S_class_t type);
> H5_DLL hid_t H5Screate_simple(int rank, const hsize_t dims[], const hsize_t maxdims[]);
12,16c10,21
< H5_DLL hssize_t H5Sget_simple_extent_npoints(hid_t space_id);
< H5_DLL int H5Sget_simple_extent_ndims(hid_t space_id);
< H5_DLL int H5Sget_simple_extent_dims(hid_t space_id, hsize_t dims[],
<     hsize_t maxdims[]);
< H5_DLL htri_t H5Sis_simple(hid_t space_id);
----
> H5_DLL herr_t H5Sencode2(hid_t obj_id, void *buf, size_t *nalloc, hid_t fapl);
> H5_DLL herr_t H5Sextent_copy(hid_t dst_id, hid_t src_id);
> H5_DLL htri_t H5Sextent_equal(hid_t space1_id, hid_t space2_id);
> H5_DLL htri_t H5Sget_regular_hyperslab(hid_t spaceid, hsize_t start[], hsize_t stride[], hsize_t count[],
>     hsize_t block[]);
> H5_DLL herr_t H5Sget_select_bounds(hid_t spaceid, hsize_t start[], hsize_t end[]);

```

```

> H5_DLL hssize_t H5Sget_select_elem_npoints(hid_t spaceid);
> H5_DLL herr_t H5Sget_select_elem_pointlist(hid_t spaceid, hsize_t startpoint, hsize_t numpoints,
>
>                                     hsize_t buf[]);
> H5_DLL herr_t H5Sget_select_hyper_blocklist(hid_t spaceid, hsize_t startblock, hsize_t numblocks,
>
>                                     hsize_t buf[]);
> H5_DLL hssize_t H5Sget_select_hyper_nblocks(hid_t spaceid);
18,33c23,26
< H5_DLL herr_t H5Sselect_hyperslab(hid_t space_id, H5S_seloper_t op,
<
<         const hsize_t start[],
<         const hsize_t _stride[],
<         const hsize_t count[],
<         const hsize_t _block[]);
< H5_DLL hid_t H5Scombine_hyperslab(hid_t space_id, H5S_seloper_t op,
<
<         const hsize_t start[],
<         const hsize_t _stride[],
<         const hsize_t count[],
<         const hsize_t _block[]);
< H5_DLL herr_t H5Sselect_select(hid_t space1_id, H5S_seloper_t op,
<
<                                     hid_t space2_id);
< H5_DLL hid_t H5Scombine_select(hid_t space1_id, H5S_seloper_t op,
<
<                                     hid_t space2_id);
< H5_DLL herr_t H5Sselect_elements(hid_t space_id, H5S_seloper_t op,
<     size_t num_elem, const hsize_t *coord);
____
> H5_DLL H5S_sel_type H5Sget_select_type(hid_t spaceid);
> H5_DLL int H5Sget_simple_extent_dims(hid_t space_id, hsize_t dims[], hsize_t maxdims[]);
> H5_DLL int H5Sget_simple_extent_ndims(hid_t space_id);
> H5_DLL hssize_t H5Sget_simple_extent_npoints(hid_t space_id);
35,37c28,37
< H5_DLL herr_t H5Sset_extent_none(hid_t space_id);
< H5_DLL herr_t H5Sextent_copy(hid_t dst_id, hid_t src_id);
< H5_DLL htri_t H5Sextent_equal(hid_t sid1, hid_t sid2);
____
> H5_DLL htri_t H5Sis_regular_hyperslab(hid_t spaceid);
> H5_DLL htri_t H5Sis_simple(hid_t space_id);
> H5_DLL herr_t H5Smodify_select(hid_t space1_id, H5S_seloper_t op, hid_t space2_id);
> H5_DLL herr_t H5Soffset_simple(hid_t space_id, const hssize_t *offset);
> H5_DLL herr_t H5Ssel_iter_close(hid_t sel_iter_id);
> H5_DLL hid_t H5Ssel_iter_create(hid_t spaceid, size_t elmt_size, unsigned flags);
> H5_DLL herr_t H5Ssel_iter_get_seq_list(hid_t sel_iter_id, size_t maxseq, size_t maxbytes, size_t *nseq,
>
>                                     size_t *nbytes, hsize_t *off, size_t *len);
> H5_DLL herr_t H5Ssel_iter_reset(hid_t sel_iter_id, hid_t space_id);
> H5_DLL herr_t H5Sselect_adjust(hid_t spaceid, const hssize_t *offset);
38a39,43

```

```

> H5_DLL herr_t H5Sselect_copy(hid_t dst_id, hid_t src_id);
> H5_DLL herr_t H5Sselect_elements(hid_t space_id, H5S_seloper_t op, size_t num_elem, const hsize_t *coord);
> H5_DLL herr_t H5Sselect_hyperslab(hid_t space_id, H5S_seloper_t op, const hsize_t start[],
>
> const hsize_t stride[], const hsize_t count[], const hsize_t block[]);
> H5_DLL htri_t H5Sselect_intersect_block(hid_t space_id, const hsize_t *start, const hsize_t *end);
40c45,47
< H5_DLL herr_t H5Soffset_simple(hid_t space_id, const hssize_t *offset);
-----
> H5_DLL hid_t H5Sselect_project_intersection(hid_t src_space_id, hid_t dst_space_id,
>
> hid_t src_intersect_space_id);
> H5_DLL htri_t H5Sselect_shape_same(hid_t space1_id, hid_t space2_id);
42,50c49,51
< H5_DLL hssize_t H5Sget_select_hyper_nblocks(hid_t spaceid);
< H5_DLL hssize_t H5Sget_select_elem_npoints(hid_t spaceid);
< H5_DLL herr_t H5Sget_select_hyper_blocklist(hid_t spaceid, hsize_t startblock,
<
< hsize_t numblocks, hsize_t buff[]);
< H5_DLL herr_t H5Sget_select_elem_pointlist(hid_t spaceid, hsize_t startpoint,
<
< hsize_t numpoints, hsize_t buff[]);
< H5_DLL herr_t H5Sget_select_bounds(hid_t spaceid, hsize_t start[],
<
< hsize_t end[]);
< H5_DLL H5S_sel_type H5Sget_select_type(hid_t spaceid);
-----
> H5_DLL herr_t H5Sset_extent_none(hid_t space_id);
> H5_DLL herr_t H5Sset_extent_simple(hid_t space_id, int rank, const hsize_t dims[], const hsize_t max[]);
> H5_DLL herr_t H5Sencode1(hid_t obj_id, void *buf, size_t *nalloc);
51a53
>
改变 H5TBpublic.h
2,130c2,45
< H5_HLDLL herr_t H5TBmake_table( const char *table_title,
<
< hid_t loc_id,
<
< const char *dset_name,
<
< hsize_t nfields,
<
< hsize_t nrecords,
<
< size_t type_size,
<
< const char *field_names[],
<
< const size_t *field_offset,
<
< const hid_t *field_types,
<
< hsize_t chunk_size,
<
< void *fill_data,
<
< int compress,
<
< const void *buf );
< H5_HLDLL herr_t H5TBappend_records( hid_t loc_id,
<
< const char *dset_name,

```



```

<                hsize_t nrecords,
<                size_t type_size,
<                const size_t *field_offset,
<                const size_t *dst_sizes,
<                const void *buf );
< H5_HLDLL herr_t  H5TBwrite_records( hid_t loc_id,
<                const char *dset_name,
<                hsize_t start,
<                hsize_t nrecords,
<                size_t type_size,
<                const size_t *field_offset,
<                const size_t *dst_sizes,
<                const void *buf );
< H5_HLDLL herr_t  H5TBwrite_fields_name( hid_t loc_id,
<                const char *dset_name,
<                const char *field_names,
<                hsize_t start,
<                hsize_t nrecords,
<                size_t type_size,
<                const size_t *field_offset,
<                const size_t *dst_sizes,
<                const void *buf );
< H5_HLDLL herr_t  H5TBwrite_fields_index( hid_t loc_id,
<                const char *dset_name,
<                hsize_t nfields,
<                const int *field_index,
<                hsize_t start,
<                hsize_t nrecords,
<                size_t type_size,
<                const size_t *field_offset,
<                const size_t *dst_sizes,
<                const void *buf );
< H5_HLDLL herr_t  H5TBread_table( hid_t loc_id,
<                const char *dset_name,
<                size_t dst_size,
<                const size_t *dst_offset,
<                const size_t *dst_sizes,
<                void *dst_buf );
< H5_HLDLL herr_t  H5TBread_fields_name( hid_t loc_id,
<                const char *dset_name,
<                const char *field_names,
<                hsize_t start,
<                hsize_t nrecords,
<                size_t type_size,

```

```

<                                const size_t *field_offset,
<                                const size_t *dst_sizes,
<                                void *buf );
< H5_HLDLL herr_t  H5TBread_fields_index( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t nfields,
<                                const int *field_index,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t type_size,
<                                const size_t *field_offset,
<                                const size_t *dst_sizes,
<                                void *buf );
< H5_HLDLL herr_t  H5TBread_records( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t type_size,
<                                const size_t *dst_offset,
<                                const size_t *dst_sizes,
<                                void *buf );
< H5_HLDLL herr_t  H5TBget_table_info ( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t *nfields,
<                                hsize_t *nrecords );
< H5_HLDLL herr_t  H5TBget_field_info( hid_t loc_id,
<                                const char *dset_name,
<                                char *field_names[],
<                                size_t *field_sizes,
<                                size_t *field_offsets,
<                                size_t *type_size );
< H5_HLDLL herr_t  H5TBdelete_record( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t start,
<                                hsize_t nrecords );
< H5_HLDLL herr_t  H5TBinsert_record( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t dst_size,
<                                const size_t *dst_offset,
<                                const size_t *dst_sizes,
<                                void *buf );
< H5_HLDLL herr_t  H5TBadd_records_from( hid_t loc_id,

```

```

<                                const char *dset_name1,
<                                hsize_t start1,
<                                hsize_t nrecords,
<                                const char *dset_name2,
<                                hsize_t start2 );
< H5_HLDLL herr_t  H5TBcombine_tables( hid_t loc_id1,
<                                const char *dset_name1,
<                                hid_t loc_id2,
<                                const char *dset_name2,
<                                const char *dset_name3 );
< H5_HLDLL herr_t  H5TBinsert_field( hid_t loc_id,
<                                const char *dset_name,
<                                const char *field_name,
<                                hid_t field_type,
<                                hsize_t position,
<                                const void *fill_data,
<                                const void *buf );
< H5_HLDLL herr_t  H5TBdelete_field( hid_t loc_id,
<                                const char *dset_name,
<                                const char *field_name );
< H5_HLDLL herr_t  H5TBAget_title( hid_t loc_id,
<                                char *table_title );
< H5_HLDLL htri_t  H5TBAget_fill(hid_t loc_id,
<                                const char *dset_name,
<                                hid_t dset_id,
<                                unsigned char *dst_buf);
____
> H5_HLDLL herr_t H5TBmake_table(const char *table_title, hid_t loc_id, const char *dset_name, hsize_t nfields,
>                                hsize_t nrecords, size_t type_size, const char *field_names[],
>                                const size_t *field_offset, const hid_t *field_types, hsize_t chunk_size,
>                                void *fill_data, int compress, const void *buf);
> H5_HLDLL herr_t H5TBappend_records(hid_t loc_id, const char *dset_name, hsize_t nrecords, size_t type_size,
>                                const size_t *field_offset, const size_t *dst_sizes, const void *buf);
> H5_HLDLL herr_t H5TBwrite_records(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords,
>                                size_t type_size, const size_t *field_offset, const size_t *dst_sizes,
>                                const void *buf);
> H5_HLDLL herr_t H5TBwrite_fields_name(hid_t loc_id, const char *dset_name, const char *field_names,
>                                hsize_t start, hsize_t nrecords, size_t type_size,
>                                const size_t *field_offset, const size_t *dst_sizes, const void *buf);
> H5_HLDLL herr_t H5TBwrite_fields_index(hid_t loc_id, const char *dset_name, hsize_t nfields,
>                                const int *field_index, hsize_t start, hsize_t nrecords,
>                                size_t type_size, const size_t *field_offset, const size_t
*dst_sizes,
>                                const void *buf);

```

```

> H5_HLDLL herr_t H5TBread_table(hid_t loc_id, const char *dset_name, size_t dst_size, const size_t *dst_offset,
>
>                                const size_t *dst_sizes, void *dst_buf);
> H5_HLDLL herr_t H5TBread_fields_name(hid_t loc_id, const char *dset_name, const char *field_names,
>
>                                hsize_t start, hsize_t nrecords, size_t type_size,
>
>                                const size_t *field_offset, const size_t *dst_sizes, void *buf);
> H5_HLDLL herr_t H5TBread_fields_index(hid_t loc_id, const char *dset_name, hsize_t nfields,
>
>                                const int *field_index, hsize_t start, hsize_t nrecords,
>
>                                size_t type_size, const size_t *field_offset, const size_t *dst_sizes,
>
>                                void *buf);
> H5_HLDLL herr_t H5TBread_records(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords,
>
>                                size_t type_size, const size_t *dst_offset, const size_t *dst_sizes,
>
>                                void *buf);
> H5_HLDLL herr_t H5TBget_table_info(hid_t loc_id, const char *dset_name, hsize_t *nfields, hsize_t *nrecords);
> H5_HLDLL herr_t H5TBget_field_info(hid_t loc_id, const char *dset_name, char *field_names[],
>
>                                size_t *field_sizes, size_t *field_offsets, size_t *type_size);
> H5_HLDLL herr_t H5TBdelete_record(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords);
> H5_HLDLL herr_t H5TBinsert_record(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords,
>
>                                size_t dst_size, const size_t *dst_offset, const size_t *dst_sizes,
>
>                                void *buf);
> H5_HLDLL herr_t H5TBadd_records_from(hid_t loc_id, const char *dset_name1, hsize_t start1, hsize_t nrecords,
>
>                                const char *dset_name2, hsize_t start2);
> H5_HLDLL herr_t H5TBcombine_tables(hid_t loc_id1, const char *dset_name1, hid_t loc_id2,
>
>                                const char *dset_name2, const char *dset_name3);
> H5_HLDLL herr_t H5TBinsert_field(hid_t loc_id, const char *dset_name, const char *field_name,
>
>                                hid_t field_type, hsize_t position, const void *fill_data, const void *buf);
> H5_HLDLL herr_t H5TBdelete_field(hid_t loc_id, const char *dset_name, const char *field_name);
> H5_HLDLL herr_t H5TBAget_title(hid_t loc_id, char *table_title);
> H5_HLDLL htri_t H5TBAget_fill(hid_t loc_id, const char *dset_name, hid_t dset_id, unsigned char *dst_buf);

```

改变 H5Tpublic.h

31a32

```
> H5_DLLVAR hid_t H5T_STD_REF_g;
```

92,93c93,94

```
< H5_DLL herr_t H5Tcommit2(hid_t loc_id, const char *name, hid_t type_id,
```

```
<     hid_t lcpl_id, hid_t tcpl_id, hid_t tapl_id);
```

```
> H5_DLL herr_t H5Tcommit2(hid_t loc_id, const char *name, hid_t type_id, hid_t lcpl_id, hid_t tcpl_id,
```

```
>                                hid_t tapl_id);
```

100,101c101,103

```
< H5_DLL herr_t H5Tinsert(hid_t parent_id, const char *name, size_t offset,
```

```
<     hid_t member_id);
```

```
> H5_DLL herr_t H5Tflush(hid_t type_id);
```

```
> H5_DLL herr_t H5Trefresh(hid_t type_id);
```

```
> H5_DLL herr_t H5Tinsert(hid_t parent_id, const char *name, size_t offset, hid_t member_id);
```

105,108c107,108

```
< H5_DLL herr_t H5Tenum_nameof(hid_t type, const void *value, char *name,  
< size_t size);  
< H5_DLL herr_t H5Tenum_valueof(hid_t type, const char *name,  
< void *value);  
____
```

```
> H5_DLL herr_t H5Tenum_nameof(hid_t type, const void *value, char *name , size_t size);  
> H5_DLL herr_t H5Tenum_valueof(hid_t type, const char *name, void *value );
```

110,111c110

```
< H5_DLL hid_t H5Tarray_create2(hid_t base_id, unsigned ndims,  
< const hsize_t dim[]);  
____
```

```
> H5_DLL hid_t H5Tarray_create2(hid_t base_id, unsigned ndims, const hsize_t dim[]);
```

123,124c122

```
< H5_DLL herr_t H5Tget_pad(hid_t type_id, H5T_pad_t *lsb,  
< H5T_pad_t *msb);  
____
```

```
> H5_DLL herr_t H5Tget_pad(hid_t type_id, H5T_pad_t *lsb , H5T_pad_t *msb );
```

126,128c124,125

```
< H5_DLL herr_t H5Tget_fields(hid_t type_id, size_t *spos,  
< size_t *epos, size_t *esize,  
< size_t *mpos, size_t *msize);  
____
```

```
> H5_DLL herr_t H5Tget_fields(hid_t type_id, size_t *spos , size_t *epos , size_t *esize ,  
> size_t *mpos , size_t *msize );
```

139c136

```
< H5_DLL herr_t H5Tget_member_value(hid_t type_id, unsigned membno, void *value);  
____
```

```
> H5_DLL herr_t H5Tget_member_value(hid_t type_id, unsigned membno, void *value );
```

149,150c146

```
< H5_DLL herr_t H5Tset_fields(hid_t type_id, size_t spos, size_t epos,  
< size_t esize, size_t mpos, size_t msize);  
____
```

```
> H5_DLL herr_t H5Tset_fields(hid_t type_id, size_t spos, size_t epos, size_t esize, size_t mpos, size_t msize);
```

156,159c152,153

```
< H5_DLL herr_t H5Tregister(H5T_pers_t pers, const char *name, hid_t src_id,  
< hid_t dst_id, H5T_conv_t func);  
< H5_DLL herr_t H5Tunregister(H5T_pers_t pers, const char *name, hid_t src_id,  
< hid_t dst_id, H5T_conv_t func);  
____
```

```
> H5_DLL herr_t H5Tregister(H5T_pers_t pers, const char *name, hid_t src_id, hid_t dst_id, H5T_conv_t func);
```

```
> H5_DLL herr_t H5Tunregister(H5T_pers_t pers, const char *name, hid_t src_id, hid_t dst_id, H5T_conv_t func);
```

162,163c156,158

```
< H5_DLL herr_t H5Tconvert(hid_t src_id, hid_t dst_id, size_t nelmts,
```

```

< void *buf, void *background, hid_t plist_id);
____
> H5_DLL herr_t H5Tconvert(hid_t src_id, hid_t dst_id, size_t nelmts, void *buf, void *background,
> hid_t plist_id);
> H5_DLL herr_t H5Treclaim(hid_t type_id, hid_t space_id, hid_t plist_id, void *buf);
166,168c161,162
< H5_DLL hid_t H5Tarray_create1(hid_t base_id, int ndims,
< const hsize_t dim[],
< const int perm[]);
____
> H5_DLL hid_t H5Tarray_create1(hid_t base_id, int ndims, const hsize_t dim[],
> const int perm[]);
170a165
>
增加头文件: H5VLconnector.h
增加头文件: H5VLconnector_passthru.h
增加头文件: H5VLnative.h
增加头文件: H5VLpassthru.h
增加头文件: H5VLpublic.h
改变 hdf5_hl.h
0a1
>

```

protobuf-c-devel

```

改变 protobuf-c.h
1a2
> static const char protobuf_c_empty_string[] = "";
增加头文件: protobuf-c.proto

```

libkysec-extend-devel

```

增加头文件: bwlist.h
改变 kmod_protect.h
3a4,5
> int add_to_kmod_protect_sm(char *ko_file);
> int add_to_kmod_protect_kdk(char *ko_file, int data_source_type, char *data_source_name);
4a7,8
> int remove_from_kmod_protect_sm(char *ko_file);
> int remove_from_kmod_protect_kdk(char *ko_file, int data_source_type, char *data_source_name);
5a10,11
> int add_kmod_to_kmod_protect_sm(char *kmod_name);

```

```

> int add_kmod_to_kmod_protect_kdk(char *kmod_name, int data_source_type, char *data_source_name);
6a13,16

> int remove_kmod_from_kmod_protect_sm(char *kmod_name);
> int remove_kmod_from_kmod_protect_kdk(char *kmod_name, int data_source_type, char *data_source_name);
> int kysec_xattr_check_kmod_del_from_kernel(char *ko_name, int data_source_type, char *data_source_name);
> int check_kmod_in_sm(char *kmod_name);
改变 kysec_communicate.h
12a13,14

> int kysec_kmod_communicate_with_kernel_sm(int mesg_type, char *kmod_name);
> int kysec_kmod_communicate_with_kernel_kdk(int mesg_type, char *kmod_name, int data_source_type, char
*data_source_name);
16a19,22

> int kysec_ppro_communicate_with_kernel_sm(int mesg_type, char *hash, int ppro_type);
> int kysec_ppro_communicate_with_kernel_kdk(int mesg_type, char *hash, int ppro_type , int data_source_type ,
char *data_source_name);
> int kysec_pblk_communicate_with_kernel(int mesg_type, char *hash);
> int kysec_pblk_communicate_with_kernel_kdk(int mesg_type, char *hash, int data_source_type, char
*data_source_name);
改变 kysec_devctl.h
7a8

> int kysec_devctl_batch_add_perm(struct kysec_devctl_info *data, int count);
12a14

> struct kysec_devctl_sysf_sm *kysec_devctl_read_sys_devinfo_sm(int *count);
14a17,27

> int kysec_devctl_xdg_insert_record(struct device_record *record, int count);
> int kysec_devctl_sm_add_perm(struct devctl_sm_info *recv);
> int kysec_devctl_sm_remove_perm(struct devctl_sm_info *recv);
> struct devctl_sm_info * kysec_devctl_sm_get_all_policies(int *count);
> struct devctl_sm_info *kysec_devctl_sm_get_no_special(int *count);
> struct devctl_sm_info *kysec_devctl_sm_read_special_xmldata(int *count);
> int kysec_devctl_get_sm();
> int kysec_devctl_rebind_usb_device();
> void kysec_devctl_sm_clean_all_policies();
> void kysec_devctl_load_ksc_policies();
> int judeg_file_existd_dir(const char *dir, const char *file_name);
增加头文件: kysec_netflow.h
改变 libkysec_dbus.h
2a3

> int kysec_protect_set_kdk(char *path, char *protect , int data_source_type , char *data_source_name);
3a5,8

> int kysec_protect_set_sm(char *path, char *protect);
> int kysec_protect_set_for_dir_kdk(char *path, char *protect, int data_source_type , char *data_source_name);
> int kysec_protect_set_for_dir(char *path, char *protect);
> int kysec_protect_set_for_dir_sm(char *path, char *protect);

```

14a20

```
> int kysec_fpro_clear_policy_kdk(const int data_source_type, char *data_source_name);
```

增加头文件：pblk.h

改变 ppro.h

3,8c3,21

```
< extern int kysec_ppro_add(char *path, int type);
```

```
< extern int kysec_ppro_remove(char *path);
```

```
< extern int kysec_ppro_update(char *path, int type);
```

```
< extern struct kysec_ppro_kernel *kysec_ppro_read_node_with_path(char *filepath);
```

```
< extern struct kysec_ppro_kernel *kysec_ppro_read_all_data(int *count);
```

```
< extern int kysec_ppro_get_action(char **action);
```

——

```
> int kysec_ppro_add(char *path, int type);
```

```
> int kysec_ppro_remove(char *path);
```

```
> int kysec_ppro_update(char *path, int type);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_node_with_path(char *filepath);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_all_data(int *count);
```

```
> int kysec_ppro_get_action(char **action);
```

```
> int kysec_ppro_add_sm(char *path, int type);
```

```
> int kysec_ppro_remove_sm(char *path);
```

```
> int kysec_ppro_update_sm(char *path, int type);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_node_with_path_sm(char *filepath);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_all_data_sm(int *count);
```

```
> int kysec_ppro_check_in_sm(char *hash);
```

```
> int kysec_ppro_add_kdk(char *path, int type, int data_source_type , char *data_source_name);
```

```
> int kysec_ppro_remove_kdk(char *path , int data_source_type , char *data_source_name);
```

```
> int kysec_ppro_update_kdk(char *path, int type , int data_source_type , char *data_source_name);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_node_with_path_kdk(char *filepath, int data_source_type , char *data_source_name);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_node_with_hash_kdk(char *hash, int data_source_type, char *data_source_name);
```

```
> struct kysec_ppro_kernel *kysec_ppro_read_all_data_kdk(int *count, int data_source_type , char *data_source_name);
```

```
> int kysec_ppro_get_same_node_by_hash(char *hash, int *count);
```

ImageMagick-c++-devel

改变 STL.h

411c411

```
< void density(const Geometry &geomery_);
```

——

```
> void density(const Geometry &geometry_);
```

1166c1166


```

< void flattenImages( Image *flattendImage_,
——
> void flattenImages( Image *flattenedImage_,
1175,1176c1175,1176
< flattendImage_->replaceImage( image );
< ThrowPPEException(flattendImage_>quiet());
——
> flattenedImage_>replaceImage( image );
> ThrowPPEException(flattenedImage_>quiet());

```

libbpf-devel

改变 bpf.h

```

2,3c2,11
< LIBBPF_API int
< bpf_create_map_xattr(const struct bpf_create_map_attr *create_attr);
——
> int libbpf_set_memlock_rlim(size_t memlock_bytes);
> LIBBPF_API int bpf_map_create(enum bpf_map_type map_type,
> const char *map_name,
> __u32 key_size,
> __u32 value_size,
> __u32 max_entries,
> const struct bpf_map_create_opts *opts);
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_map_create() instead")
> LIBBPF_API int bpf_create_map_xattr(const struct bpf_create_map_attr *create_attr);
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_map_create() instead")
6a15
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_map_create() instead")
9a19
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_map_create() instead")
11a22
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_map_create() instead")
15a27
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_map_create() instead")
20,22c32,43
< LIBBPF_API int
< bpf_load_program_xattr(const struct bpf_load_program_attr *load_attr,
< char *log_buf, size_t log_buf_sz);
——
> LIBBPF_API int bpf_prog_load(enum bpf_prog_type prog_type,
> const char *prog_name, const char *license,
> const struct bpf_insn *insns, size_t insn_cnt,

```

```

> const struct bpf_prog_load_opts *opts);
> LIBBPF_API int bpf_prog_load_v0_6_0(enum bpf_prog_type prog_type,
> const char *prog_name, const char *license,
> const struct bpf_insn *insns, size_t insn_cnt,
> const struct bpf_prog_load_opts *opts);
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_prog_load() instead")
> LIBBPF_API int bpf_load_program_xattr(const struct bpf_load_program_attr *load_attr,
> char *log_buf, size_t log_buf_sz);
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_prog_load() instead")
26a48
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_prog_load() instead")
32a55,59
> LIBBPF_API int bpf_btf_load(const void *btf_data, size_t btf_size,
> const struct bpf_btf_load_opts *opts);
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_btf_load() instead")
> LIBBPF_API int bpf_load_btf(const void *btf, __u32 btf_size, char *log_buf,
> __u32 log_buf_size, bool do_log);
39a67,68
> LIBBPF_API int bpf_map_lookup_and_delete_elem_flags(int fd, const void *key,
> void *value, __u64 flags);
40a70
> LIBBPF_API int bpf_map_delete_elem_flags(int fd, const void *key, __u64 flags);
43c73
< LIBBPF_API int bpf_map_delete_batch(int fd, void *keys,
____
> LIBBPF_API int bpf_map_delete_batch(int fd, const void *keys,
53c83
< LIBBPF_API int bpf_map_update_batch(int fd, void *keys, void *values,
____
> LIBBPF_API int bpf_map_update_batch(int fd, const void *keys, const void *values,
59a90,93
> LIBBPF_API int bpf_prog_attach_opts(int prog_fd, int attachable_fd,
> enum bpf_attach_type type,
> const struct bpf_prog_attach_opts *opts);
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_prog_attach_opts() instead")
72a107
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_prog_test_run_opts() instead")
73a109
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_prog_test_run_opts() instead")
90,91d125
< LIBBPF_API int bpf_load_btf(const void *btf, __u32 btf_size, char *log_buf,
< __u32 log_buf_size, bool do_log);
94a129
> LIBBPF_API int bpf_enable_stats(int type);

```

改变 bpf_helper_defs.h

26a27

> struct unix_sock;

35a37

> struct bpf_timer;

191,200c193,207

```
< static void *(*bpf_task_storage_get)(void *map, struct task_struct *task, void *value, __u64 flags) = (void *) 156;
< static long (*bpf_task_storage_delete)(void *map, struct task_struct *task) = (void *) 157;
< static struct task_struct *(*bpf_get_current_task_btf)(void) = (void *) 158;
< static long (*bpf_bprm_opts_set)(struct linux_binprm *bprm, __u64 flags) = (void *) 159;
< static __u64 (*bpf_ktime_get_coarse_ns)(void) = (void *) 160;
< static long (*bpf_ima_inode_hash)(struct inode *inode, void *dst, __u32 size) = (void *) 161;
< static struct socket *(*bpf_sock_from_file)(struct file *file) = (void *) 162;
< static long (*bpf_check_mtu)(void *ctx, __u32 ifindex, __u32 *mtu_len, __s32 len_diff, __u64 flags) = (void *)
163;
< static long (*bpf_for_each_map_elem)(void *map, void *callback_fn, void *callback_ctx, __u64 flags) = (void *)
164;
< static long (*bpf_snprintf)(char *str, __u32 str_size, const char *fmt, __u64 *data, __u32 data_len) = (void *) 165;
____
> static __u64 (*bpf_get_sockops_uid_gid)(void *sockops) = (void *) 156;
> static int (*bpf_sk_original_addr)(void *bpf_socket, int optname, char *optval, int optlen) = (void *) 157;
> static long (*bpf_sched_tg_tag_of)(struct task_group *tg) = (void *) 158;
> static long (*bpf_sched_task_tag_of)(struct task_struct *tsk) = (void *) 159;
> static int (*bpf_sched_set_tg_tag)(struct task_group *tg, __s64 tag) = (void *) 160;
> static int (*bpf_sched_set_task_tag)(struct task_struct *tsk, __s64 tag) = (void *) 161;
> static int (*bpf_sched_cpu_stats_of)(int cpu, struct bpf_sched_cpu_stats *ctx, int len) = (void *) 162;
> static long (*bpf_init_cpu_topology)(void *map) = (void *) 163;
> static int (*bpf_get_cpumask_info)(void *map, struct bpf_cpumask_info *cpus) = (void *) 164;
> static long (*bpf_sched_entity_is_task)(struct sched_entity *se) = (void *) 165;
> static struct task_struct *(*bpf_sched_entity_to_task)(struct sched_entity *se) = (void *) 166;
> static struct task_group *(*bpf_sched_entity_to_tg)(struct sched_entity *se) = (void *) 167;
> static int (*bpf_cpumask_op)(struct cpumask_op_args *op, int len) = (void *) 168;
> static int (*bpf_cpus_share_cache)(int src_cpu, int dst_cpu) = (void *) 169;
> static long (*bpf_is_local_ipaddr)(__u32 ipaddr) = (void *) 170;
```

改变 bpf_tracing.h

3d2

< struct pt_regs;

改变 btf.h

16a17,24

```
> LIBBPF_API struct btf *btf__load_vmlinux_btf(void);
> LIBBPF_API struct btf *btf__load_module_btf(const char *module_name, struct btf *vmlinux_btf);
> LIBBPF_API struct btf *libbpf_find_kernel_btf(void);
> LIBBPF_API struct btf *btf__load_from_kernel_by_id(__u32 id);
> LIBBPF_API struct btf *btf__load_from_kernel_by_id_split(__u32 id, struct btf *base_btf);
```

```

> LIBBPF_DEPRECATED_SINCE(0, 6, "use btf__load_from_kernel_by_id instead")
> LIBBPF_API int btf__get_from_id(__u32 id, struct btf **btf);
> LIBBPF_DEPRECATED_SINCE(0, 6, "intended for internal libbpf use only")
17a26
> LIBBPF_DEPRECATED_SINCE(0, 6, "use btf__load_into_kernel instead")
18a28
> LIBBPF_API int btf__load_into_kernel(struct btf *btf);
22a33
> LIBBPF_DEPRECATED_SINCE(0, 7, "use btf__type_cnt() instead; note that btf__get_nr_types() ==
btf__type_cnt() - 1")
23a35
> LIBBPF_API __u32 btf__type_cnt(const struct btf *btf);
36c48
< LIBBPF_API const void *btf__get_raw_data(const struct btf *btf, __u32 *size);
----
> LIBBPF_API const void *btf__raw_data(const struct btf *btf, __u32 *size);
39c51
< LIBBPF_API int btf__get_from_id(__u32 id, struct btf **btf);
----
> LIBBPF_DEPRECATED_SINCE(0, 7, "this API is not necessary when BTF-defined maps are used")
44c56
< LIBBPF_API struct btf_ext *btf_ext__new(__u8 *data, __u32 size);
----
> LIBBPF_API struct btf_ext *btf_ext__new(const __u8 *data, __u32 size);
46,47c58
< LIBBPF_API const void *btf_ext__get_raw_data(const struct btf_ext *btf_ext,
<
__u32 *size);
----
> LIBBPF_API const void *btf_ext__raw_data(const struct btf_ext *btf_ext, __u32 *size);
58,60c69,72
< LIBBPF_API __u32 btf_ext__func_info_rec_size(const struct btf_ext *btf_ext);
< LIBBPF_API __u32 btf_ext__line_info_rec_size(const struct btf_ext *btf_ext);
< LIBBPF_API struct btf *libbpf_find_kernel_btf(void);
----
> LIBBPF_API LIBBPF_DEPRECATED("btf_ext__reloc_func_info is deprecated; write custom func_info parsing to
fetch rec_size")
> __u32 btf_ext__func_info_rec_size(const struct btf_ext *btf_ext);
> LIBBPF_API LIBBPF_DEPRECATED("btf_ext__reloc_line_info is deprecated; write custom line_info parsing to
fetch rec_size")
> __u32 btf_ext__line_info_rec_size(const struct btf_ext *btf_ext);
64a77
> LIBBPF_API int btf__add_btf(struct btf *btf, const struct btf *src_btf);
79a93
> LIBBPF_API int btf__add_type_tag(struct btf *btf, const char *value, int ref_type_id);

```

88,89c102,107

```
< LIBBPF_API int btf__dedup(struct btf *btf, struct btf_ext *btf_ext,
<
    const struct btf_dedup_opts *opts);
____
> LIBBPF_API int btf__add_decl_tag(struct btf *btf, const char *value, int ref_type_id,
>
    int component_idx);
> LIBBPF_API int btf__dedup(struct btf *btf, const struct btf_dedup_opts *opts);
> LIBBPF_API int btf__dedup_v0_6_0(struct btf *btf, const struct btf_dedup_opts *opts);
> LIBBPF_DEPRECATED_SINCE(0, 7, "use btf__dedup() instead")
> LIBBPF_API int btf__dedup_deprecated(struct btf *btf, struct btf_ext *btf_ext, const void *opts);
```

92,94c110,120

```
<
    const struct btf_ext *btf_ext,
<
    const struct btf_dump_opts *opts,
<
    btf_dump_printf_fn_t printf_fn);
____
>
    btf_dump_printf_fn_t printf_fn,
>
    void *ctx,
>
    const struct btf_dump_opts *opts);
> LIBBPF_API struct btf_dump *btf_dump__new_v0_6_0(const struct btf *btf,
>
    btf_dump_printf_fn_t printf_fn,
>
    void *ctx,
>
    const struct btf_dump_opts *opts);
> LIBBPF_API struct btf_dump *btf_dump__new_deprecated(const struct btf *btf,
>
    const struct btf_ext *btf_ext,
>
    const struct btf_dump_opts *opts,
>
    btf_dump_printf_fn_t printf_fn);
```

99a126,129

```
> LIBBPF_API int
> btf_dump__dump_type_data(struct btf_dump *d, __u32 id,
>
    const void *data, size_t data_sz,
>
    const struct btf_dump_type_data_opts *opts);
```

143a174,177

```
> static inline bool btf_is_decl_tag(const struct btf_type *t)
>
>
> static inline bool btf_is_type_tag(const struct btf_type *t)
>
```

164a199,201

```
>
> struct btf_decl_tag;
> static inline struct btf_decl_tag *btf_decl_tag(const struct btf_type *t)
```

改变 libbpf.h

1a2,4

```
> LIBBPF_API __u32 libbpf_major_version(void);
> LIBBPF_API __u32 libbpf_minor_version(void);
```

```

> LIBBPF_API const char *libbpf_version_string(void);
10a14
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_object__open_mem() instead")
13a18
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__open_file() instead")
25a31
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_object__load() instead")
26a33
> LIBBPF_DEPRECATED_SINCE(0, 6, "bpf_object__unload() is deprecated, use bpf_object__close() instead")
33a41
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__find_program_by_name() instead")
40c48,50
< LIBBPF_API struct bpf_object *bpf_object__next(struct bpf_object *prev);
____
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "track bpf_objects in application code instead")
> struct bpf_object *bpf_object__next(struct bpf_object *prev);
> LIBBPF_DEPRECATED_SINCE(0, 7, "storage via set_priv/priv is deprecated")
42a53
> LIBBPF_DEPRECATED_SINCE(0, 7, "storage via set_priv/priv is deprecated")
52,55c63,72
< LIBBPF_API struct bpf_program *bpf_program__next(struct bpf_program *prog,
<
const struct bpf_object *obj);
< LIBBPF_API struct bpf_program *bpf_program__prev(struct bpf_program *prog,
<
const struct bpf_object *obj);
____
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__next_program() instead")
>
const struct bpf_object *obj);
> LIBBPF_API struct bpf_program *
> bpf_object__next_program(const struct bpf_object *obj, struct bpf_program *prog);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__prev_program() instead")
> struct bpf_program *bpf_program__prev(struct bpf_program *prog,
>
const struct bpf_object *obj);
> LIBBPF_API struct bpf_program *
> bpf_object__prev_program(const struct bpf_object *obj, struct bpf_program *prog);
> LIBBPF_DEPRECATED_SINCE(0, 7, "storage via set_priv/priv is deprecated")
57a75
> LIBBPF_DEPRECATED_SINCE(0, 7, "storage via set_priv/priv is deprecated")
66a85
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_program__insn_cnt() instead")
68,69c87,93
< LIBBPF_API int bpf_program__load(struct bpf_program *prog, char *license,
<
__u32 kern_version);
____
> struct bpf_insn;

```

```

> LIBBPF_API const struct bpf_insn *bpf_program__insns(const struct bpf_program *prog);
> LIBBPF_API int bpf_program__set_insns(struct bpf_program *prog,
>                                     struct bpf_insn *new_insns, size_t new_insn_cnt);
> LIBBPF_API size_t bpf_program__insn_cnt(const struct bpf_program *prog);
> LIBBPF_DEPRECATED_SINCE(0, 6, "use bpf_object__load() instead")
> LIBBPF_API int bpf_program__load(struct bpf_program *prog, const char *license, __u32 kern_version);
70a95
> LIBBPF_DEPRECATED_SINCE(0, 7, "multi-instance bpf_program support is deprecated")
73a99
> LIBBPF_DEPRECATED_SINCE(0, 7, "multi-instance bpf_program support is deprecated")
92c118,124
< bpf_program__attach(struct bpf_program *prog);
____
> bpf_program__attach(const struct bpf_program *prog);
> struct bpf_perf_event_opts {
>     size_t sz;
>     __u64 bpf_cookie;
> };
> LIBBPF_API struct bpf_link *
> bpf_program__attach_perf_event(const struct bpf_program *prog, int pfd);
94c126,127
< bpf_program__attach_perf_event(struct bpf_program *prog, int pfd);
____
> bpf_program__attach_perf_event_opts(const struct bpf_program *prog, int pfd,
>                                     const struct bpf_perf_event_opts *opts);
96c129
< bpf_program__attach_kprobe(struct bpf_program *prog, bool retprobe,
____
> bpf_program__attach_kprobe(const struct bpf_program *prog, bool retprobe,
99c132,140
< bpf_program__attach_uprobe(struct bpf_program *prog, bool retprobe,
____
> bpf_program__attach_kprobe_opts(const struct bpf_program *prog,
>                                 const char *func_name,
>                                 const struct bpf_kprobe_opts *opts);
> LIBBPF_API struct bpf_link *
> bpf_program__attach_kprobe_multi_opts(const struct bpf_program *prog,
>                                       const char *pattern,
>                                       const struct bpf_kprobe_multi_opts *opts);
> LIBBPF_API struct bpf_link *
> bpf_program__attach_uprobe(const struct bpf_program *prog, bool retprobe,
103c144,153
< bpf_program__attach_tracepoint(struct bpf_program *prog,
____

```

```

> bpf_program__attach_uprobe_opts(const struct bpf_program *prog, pid_t pid,
>
>         const char *binary_path, size_t func_offset,
>
>         const struct bpf_uprobe_opts *opts);
> LIBBPF_API struct bpf_link *
> bpf_program__attach_usdt(const struct bpf_program *prog,
>
>         pid_t pid, const char *binary_path,
>
>         const char *usdt_provider, const char *usdt_name,
>
>         const struct bpf_usdt_opts *opts);
> LIBBPF_API struct bpf_link *
> bpf_program__attach_tracepoint(const struct bpf_program *prog,
107c157,162
< bpf_program__attach_raw_tracepoint(struct bpf_program *prog,
____
> bpf_program__attach_tracepoint_opts(const struct bpf_program *prog,
>
>         const char *tp_category,
>
>         const char *tp_name,
>
>         const struct bpf_tracepoint_opts *opts);
> LIBBPF_API struct bpf_link *
> bpf_program__attach_raw_tracepoint(const struct bpf_program *prog,
110c165,167
< bpf_program__attach_trace(struct bpf_program *prog);
____
> bpf_program__attach_trace(const struct bpf_program *prog);
> LIBBPF_API struct bpf_link *
> bpf_program__attach_trace_opts(const struct bpf_program *prog, const struct bpf_trace_opts *opts);
112c169
< bpf_program__attach_lsm(struct bpf_program *prog);
____
> bpf_program__attach_lsm(const struct bpf_program *prog);
114c171
< bpf_program__attach_cgroup(struct bpf_program *prog, int cgroup_fd);
____
> bpf_program__attach_cgroup(const struct bpf_program *prog, int cgroup_fd);
116c173
< bpf_program__attach_netns(struct bpf_program *prog, int netns_fd);
____
> bpf_program__attach_netns(const struct bpf_program *prog, int netns_fd);
118c175
< bpf_program__attach_xdp(struct bpf_program *prog, int ifindex);
____
> bpf_program__attach_xdp(const struct bpf_program *prog, int ifindex);
120c177
< bpf_program__attach_freplace(struct bpf_program *prog,
____

```



```

> bpf_program__attach_freplace(const struct bpf_program *prog,
123c180
< LIBBPF_API struct bpf_link *bpf_map__attach_struct_ops(struct bpf_map *map);
-----
> LIBBPF_API struct bpf_link *bpf_map__attach_struct_ops(const struct bpf_map *map);
125c182
< bpf_program__attach_iter(struct bpf_program *prog,
-----
> bpf_program__attach_iter(const struct bpf_program *prog,
127c184
< struct bpf_insn;
-----
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_program__insns() for getting bpf_program instructions")
129a187
> LIBBPF_DEPRECATED_SINCE(0, 7, "multi-instance bpf_program support is deprecated")
130a189
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
131a191
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
132a193
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
133a195
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
134a197
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
135a199
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
136a201
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
137a203
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
138a205
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
139a207
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
140a209
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
141a211
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
142a213
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__set_type() instead")
144,146c215,217
< LIBBPF_API enum bpf_prog_type bpf_program__get_type(const struct bpf_program *prog);
< LIBBPF_API void bpf_program__set_type(struct bpf_program *prog,

```

```

<
enum bpf_prog_type type);
____

> LIBBPF_API enum bpf_prog_type bpf_program__type(const struct bpf_program *prog);
> LIBBPF_API int bpf_program__set_type(struct bpf_program *prog,
>
enum bpf_prog_type type);
148,149c219,220
< bpf_program__get_expected_attach_type(const struct bpf_program *prog);
< LIBBPF_API void
____

> bpf_program__expected_attach_type(const struct bpf_program *prog);
> LIBBPF_API int
151a223,228
> LIBBPF_API __u32 bpf_program__flags(const struct bpf_program *prog);
> LIBBPF_API int bpf_program__set_flags(struct bpf_program *prog, __u32 flags);
> LIBBPF_API __u32 bpf_program__log_level(const struct bpf_program *prog);
> LIBBPF_API int bpf_program__set_log_level(struct bpf_program *prog, __u32 log_level);
> LIBBPF_API const char *bpf_program__log_buf(const struct bpf_program *prog, size_t *log_size);
> LIBBPF_API int bpf_program__set_log_buf(struct bpf_program *prog, char *log_buf, size_t log_size);
154a232
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
155a234
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
156a236
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
157a238
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
158a240
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
159a242
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
160a244
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
161a246
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
162a248
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
163a250
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
164a252
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
165a254
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")
166a256
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_program__type() instead")

```

172c262

< LIBBPF_API struct bpf_map *

> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_object__find_map_by_name() instead")

173a264,265

> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__next_map() instead")

> struct bpf_map *bpf_map__next(const struct bpf_map *map, const struct bpf_object *obj);

175c267,269

< bpf_map__next(const struct bpf_map *map, const struct bpf_object *obj);

> bpf_object__next_map(const struct bpf_object *obj, const struct bpf_map *map);

> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__prev_map() instead")

> struct bpf_map *bpf_map__prev(const struct bpf_map *map, const struct bpf_object *obj);

177c271,273

< bpf_map__prev(const struct bpf_map *map, const struct bpf_object *obj);

> bpf_object__prev_map(const struct bpf_object *obj, const struct bpf_map *map);

> LIBBPF_API int bpf_map__set_autocreate(struct bpf_map *map, bool autocreate);

> LIBBPF_API bool bpf_map__autocreate(const struct bpf_map *map);

180c276,277

< LIBBPF_API const struct bpf_map_def *bpf_map__def(const struct bpf_map *map);

> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 8, "use appropriate getters or setters instead")

> const struct bpf_map_def *bpf_map__def(const struct bpf_map *map);

185a283

> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_map__set_max_entries() instead")

198a297,299

> LIBBPF_API __u64 bpf_map__map_extra(const struct bpf_map *map);

> LIBBPF_API int bpf_map__set_map_extra(struct bpf_map *map, __u64 map_extra);

> LIBBPF_DEPRECATED_SINCE(0, 7, "storage via set_priv/priv is deprecated")

200a302

> LIBBPF_DEPRECATED_SINCE(0, 7, "storage via set_priv/priv is deprecated")

203a306,307

> LIBBPF_API const void *bpf_map__initial_value(struct bpf_map *map, size_t *psize);

> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_map__type() instead")

207c311

< LIBBPF_API const char *bpf_map__get_pin_path(const struct bpf_map *map);

> LIBBPF_API const char *bpf_map__pin_path(const struct bpf_map *map);

212a317,329

> LIBBPF_API int bpf_map__lookup_elem(const struct bpf_map *map,

> const void *key, size_t key_sz,

> void *value, size_t value_sz, __u64 flags);

> LIBBPF_API int bpf_map__update_elem(const struct bpf_map *map,

```

>         const void *key, size_t key_sz,
>         const void *value, size_t value_sz, __u64 flags);
> LIBBPF_API int bpf_map__delete_elem(const struct bpf_map *map,
>         const void *key, size_t key_sz, __u64 flags);
> LIBBPF_API int bpf_map__lookup_and_delete_elem(const struct bpf_map *map,
>         const void *key, size_t key_sz,
>         void *value, size_t value_sz, __u64 flags);
> LIBBPF_API int bpf_map__get_next_key(const struct bpf_map *map,
>         const void *cur_key, void *next_key, size_t key_sz);
213a331,337
> struct bpf_prog_load_attr {
>     const char *file;
>     int ifindex;
>     int log_level;
>     int prog_flags;
> };
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_object__open() and bpf_object__load() instead")
216,217c340,343
< LIBBPF_API int bpf_prog_load(const char *file, enum bpf_prog_type type,
<         struct bpf_object **pobj, int *prog_fd);
---
> LIBBPF_DEPRECATED_SINCE(0, 7, "use bpf_object__open() and bpf_object__load() instead")
> LIBBPF_API int bpf_prog_load_deprecated(const char *file, enum bpf_prog_type type,
>         struct bpf_object **pobj, int *prog_fd);
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_xdp_attach() instead")
218a345
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_xdp_attach() instead")
220a348
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_xdp_query_id() instead")
221a350
> LIBBPF_DEPRECATED_SINCE(0, 8, "use bpf_xdp_query() instead")
223a353,358
> LIBBPF_API int bpf_xdp_attach(int ifindex, int prog_fd, __u32 flags,
>         const struct bpf_xdp_attach_opts *opts);
> LIBBPF_API int bpf_xdp_detach(int ifindex, __u32 flags,
>         const struct bpf_xdp_attach_opts *opts);
> LIBBPF_API int bpf_xdp_query(int ifindex, int flags, struct bpf_xdp_query_opts *opts);
> LIBBPF_API int bpf_xdp_query_id(int ifindex, int flags, __u32 *prog_id);
244a380
>         perf_buffer_sample_fn sample_cb, perf_buffer_lost_fn lost_cb, void *ctx,
245a382,392
> LIBBPF_API struct perf_buffer *
> perf_buffer__new_v0_6_0(int map_fd, size_t page_cnt,
>         perf_buffer_sample_fn sample_cb, perf_buffer_lost_fn lost_cb, void *ctx,

```

```

>         const struct perf_buffer_opts *opts);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "use new variant of perf_buffer__new() instead")
>         const struct perf_buffer_opts *opts);
> enum bpf_perf_event_ret {
>     LIBBPF_PERF_EVENT_DONE    = 0,
>     LIBBPF_PERF_EVENT_ERROR   = -1,
>     LIBBPF_PERF_EVENT_CONT    = -2,
> };
246a394,398
>     };
>     int cpu_cnt;
>     int *cpus;
>     int *map_keys;
> };
248c400,401
< perf_buffer__new_raw(int map_fd, size_t page_cnt,
-----
> perf_buffer__new_raw(int map_fd, size_t page_cnt, struct perf_event_attr *attr,
>         perf_buffer_event_fn event_cb, void *ctx,
249a403,408
> LIBBPF_API struct perf_buffer *
> perf_buffer__new_raw_v0_6_0(int map_fd, size_t page_cnt, struct perf_event_attr *attr,
>         perf_buffer_event_fn event_cb, void *ctx,
>         const struct perf_buffer_raw_opts *opts);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "use new variant of perf_buffer__new_raw() instead")
>         const struct perf_buffer_raw_opts *opts);
256a416,457
> LIBBPF_DEPRECATED_SINCE(0, 8, "use perf_buffer__poll() or perf_buffer__consume() instead")
> LIBBPF_API enum bpf_perf_event_ret
> bpf_perf_event_read_simple(void *mmap_mem, size_t mmap_size, size_t page_size,
>         void **copy_mem, size_t *copy_size,
>         bpf_perf_event_print_t fn, void *private_data);
> struct bpf_prog_linfo;
> struct bpf_prog_info;
> LIBBPF_API void bpf_prog_linfo__free(struct bpf_prog_linfo *prog_linfo);
> LIBBPF_API struct bpf_prog_linfo *
> bpf_prog_linfo__new(const struct bpf_prog_info *info);
> LIBBPF_API const struct bpf_line_info *
> bpf_prog_linfo__lfind_addr_func(const struct bpf_prog_linfo *prog_linfo,
>         __u64 addr, __u32 func_idx, __u32 nr_skip);
> LIBBPF_API const struct bpf_line_info *
> bpf_prog_linfo__lfind(const struct bpf_prog_linfo *prog_linfo,
>         __u32 insn_off, __u32 nr_skip);
> LIBBPF_DEPRECATED_SINCE(0, 8, "use libbpf_probe_bpf_prog_type() instead")

```

```

> LIBBPF_API bool bpf_probe_prog_type(enum bpf_prog_type prog_type, __u32 ifindex);
> LIBBPF_DEPRECATED_SINCE(0, 8, "use libbpf_probe_bpf_map_type() instead")
> LIBBPF_API bool bpf_probe_map_type(enum bpf_map_type map_type, __u32 ifindex);
> LIBBPF_DEPRECATED_SINCE(0, 8, "use libbpf_probe_bpf_helper() instead")
> LIBBPF_API bool bpf_probe_helper(enum bpf_func_id id, enum bpf_prog_type prog_type, __u32 ifindex);
> LIBBPF_DEPRECATED_SINCE(0, 8, "implement your own or use bpftool for feature detection")
> LIBBPF_API bool bpf_probe_large_insn_limit(__u32 ifindex);
> LIBBPF_API int libbpf_probe_bpf_prog_type(enum bpf_prog_type prog_type, const void *opts);
> LIBBPF_API int libbpf_probe_bpf_map_type(enum bpf_map_type map_type, const void *opts);
> LIBBPF_API int libbpf_probe_bpf_helper(enum bpf_prog_type prog_type,
>
>         enum bpf_func_id helper_id, const void *opts);
> enum bpf_prog_info_array {
>     BPF_PROG_INFO_FIRST_ARRAY = 0,
>     BPF_PROG_INFO_JITED_INSNS = 0,
>     BPF_PROG_INFO_XLATED_INSNS,
>     BPF_PROG_INFO_MAP_IDS,
>     BPF_PROG_INFO_JITED_KSYMS,
>     BPF_PROG_INFO_JITED_FUNC_LENS,
>     BPF_PROG_INFO_FUNC_INFO,
>     BPF_PROG_INFO_LINE_INFO,
>     BPF_PROG_INFO_JITED_LINE_INFO,
>     BPF_PROG_INFO_PROG_TAGS,
>     BPF_PROG_INFO_LAST_ARRAY,
> };
> LIBBPF_DEPRECATED_SINCE(0, 6, "use a custom linear prog_info wrapper")
258a460
> LIBBPF_DEPRECATED_SINCE(0, 6, "use a custom linear prog_info wrapper")
260a463
> LIBBPF_DEPRECATED_SINCE(0, 6, "use a custom linear prog_info wrapper")
270a474,479
> LIBBPF_API int
> bpf_object__open_subskeleton(struct bpf_object_subskeleton *s);
> LIBBPF_API void
> bpf_object__destroy_subskeleton(struct bpf_object_subskeleton *s);
> LIBBPF_API int bpf_object__gen_loader(struct bpf_object *obj,
>
>         struct gen_loader_opts *opts);
277a487,492
> struct bpf_prog_load_opts;
> LIBBPF_API int libbpf_register_prog_handler(const char *sec,
>
>         enum bpf_prog_type prog_type,
>
>         enum bpf_attach_type exp_attach_type,
>
>         const struct libbpf_prog_handler_opts *opts);
> LIBBPF_API int libbpf_unregister_prog_handler(int handler_id);

```

增加头文件：libbpf_legacy.h

增加头文件: libbpf_version.h

增加头文件: skel_internal.h

增加头文件: usdt.bpf.h

改变 xsk.h

42,79c42,87

```
< LIBBPF_API int xsk_umem__fd(const struct xsk_umem *umem);
< LIBBPF_API int xsk_socket__fd(const struct xsk_socket *xsk);
< LIBBPF_API int xsk_setup_xdp_prog(int ifindex,
<
<         int *xsks_map_fd);
< LIBBPF_API int xsk_socket__update_xskmap(struct xsk_socket *xsk,
<
<         int xsks_map_fd);
< LIBBPF_API int xsk_umem__create(struct xsk_umem **umem,
<
<         void *umem_area, __u64 size,
<
<         struct xsk_ring_prod *fill,
<
<         struct xsk_ring_cons *comp,
<
<         const struct xsk_umem_config *config);
< LIBBPF_API int xsk_umem__create_v0_0_2(struct xsk_umem **umem,
<
<         void *umem_area, __u64 size,
<
<         struct xsk_ring_prod *fill,
<
<         struct xsk_ring_cons *comp,
<
<         const struct xsk_umem_config *config);
< LIBBPF_API int xsk_umem__create_v0_0_4(struct xsk_umem **umem,
<
<         void *umem_area, __u64 size,
<
<         struct xsk_ring_prod *fill,
<
<         struct xsk_ring_cons *comp,
<
<         const struct xsk_umem_config *config);
< LIBBPF_API int xsk_socket__create(struct xsk_socket **xsk,
<
<         const char *ifname, __u32 queue_id,
<
<         struct xsk_umem *umem,
<
<         struct xsk_ring_cons *rx,
<
<         struct xsk_ring_prod *tx,
<
<         const struct xsk_socket_config *config);
< LIBBPF_API int
< xsk_socket__create_shared(struct xsk_socket **xsk_ptr,
<
<         const char *ifname,
<
<         __u32 queue_id, struct xsk_umem *umem,
<
<         struct xsk_ring_cons *rx,
<
<         struct xsk_ring_prod *tx,
<
<         struct xsk_ring_prod *fill,
<
<         struct xsk_ring_cons *comp,
<
<         const struct xsk_socket_config *config);
< LIBBPF_API int xsk_umem__delete(struct xsk_umem *umem);
< LIBBPF_API void xsk_socket__delete(struct xsk_socket *xsk);
```

—

```

> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_umem__fd(const struct xsk_umem *umem);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_socket__fd(const struct xsk_socket *xsk);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_setup_xdp_prog(int ifindex, int *xsks_map_fd);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_socket__update_xskmap(struct xsk_socket *xsk, int xsks_map_fd);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_umem__create(struct xsk_umem **umem,
>                     void *umem_area, __u64 size,
>                     struct xsk_ring_prod *fill,
>                     struct xsk_ring_cons *comp,
>                     const struct xsk_umem_config *config);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_umem__create_v0_0_2(struct xsk_umem **umem,
>                             void *umem_area, __u64 size,
>                             struct xsk_ring_prod *fill,
>                             struct xsk_ring_cons *comp,
>                             const struct xsk_umem_config *config);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_umem__create_v0_0_4(struct xsk_umem **umem,
>                             void *umem_area, __u64 size,
>                             struct xsk_ring_prod *fill,
>                             struct xsk_ring_cons *comp,
>                             const struct xsk_umem_config *config);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_socket__create(struct xsk_socket **xsk,
>                       const char *ifname, __u32 queue_id,
>                       struct xsk_umem *umem,
>                       struct xsk_ring_cons *rx,
>                       struct xsk_ring_prod *tx,
>                       const struct xsk_socket_config *config);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_socket__create_shared(struct xsk_socket **xsk_ptr,
>                              const char *ifname,
>                              __u32 queue_id, struct xsk_umem *umem,
>                              struct xsk_ring_cons *rx,
>                              struct xsk_ring_prod *tx,
>                              struct xsk_ring_prod *fill,
>                              struct xsk_ring_cons *comp,
>                              const struct xsk_socket_config *config);
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> int xsk_umem__delete(struct xsk_umem *umem);

```



```
> LIBBPF_API LIBBPF_DEPRECATED_SINCE(0, 7, "AF_XDP support deprecated and moved to libxdp")
> void xsk_socket__delete(struct xsk_socket *xsk);
```

protobuf-devel

变动很大，略

varnish-devel

增加头文件: bereq_flags.h

增加头文件: beresp_flags.h

减少头文件: bo_flags.h

改变 cache_backend.h

5d4

```
< void VBE_SetHappy(const struct backend *, uint64_t);
```

改变 cache_filter.h

8,9c8,11

```
< void VRT_AddVDP(VRT_CTX, const struct vdp *);
```

```
< void VRT_RemoveVDP(VRT_CTX, const struct vdp *);
```

```
> void v_deprecated_ VRT_AddVDP(VRT_CTX, const struct vdp *);
```

```
> void v_deprecated_ VRT_RemoveVDP(VRT_CTX, const struct vdp *);
```

```
> const char *VRT_AddFilter(VRT_CTX, const struct vfp *, const struct vdp *);
```

```
> void VRT_RemoveFilter(VRT_CTX, const struct vfp *, const struct vdp *);
```

改变 cache.h

1,3c1

```
< struct VSC_lck;
```

```
< struct VSC_main;
```

```
< struct VSC_main_wrk;
```

```
> extern const struct stream_close SC_NULL[1];
```

7a6

```
> struct listen_sock;
```

11a11

```
> struct req_step;
```

14,15d13

```
< struct worker;
```

```
< struct listen_sock;
```

17c15,19

```
< struct req_step;
```

```
> struct VSC_lck;
```

```
> struct VSC_main;
```

```
> struct VSC_main_wrk;
```

```

> struct worker;
> struct worker_priv;
改变 cache_varnishd.h
12d11
< void VBE_Poll(void);
20c19
< struct busyobj *VBO_GetBusyObj(struct worker *, const struct req *);
——
> struct busyobj *VBO_GetBusyObj(const struct worker *, const struct req *);
24a24
> stream_close_t VDI_Http1Pipe(struct req *, struct busyobj *);
30c30
< uint64_t VDP_Close(struct vdp_ctx *);
——
> uint64_t VDP_Close(struct vdp_ctx *, struct objcore *, struct boc *);
32c32,33
< int VDP_Push(struct vdp_ctx *, struct ws *, const struct vdp *, void *priv);
——
> int VDP_Push(VRT_CTX, struct vdp_ctx *, struct ws *, const struct vdp *,
> void *priv);
44d44
< extern int cache_shutdown;
64c64
< int VFP_Open(struct vfp_ctx *);
——
> int VFP_Open(VRT_CTX, struct vfp_ctx *);
76a77,78
> vxid_t VXID_Get(const struct worker *, uint64_t marker);
> extern pthread_key_t witness_key;
82a85,86
> void THR_SetWorker(const struct worker *);
> struct worker * THR_GetWorker(void);
109d112
< int ObjSetU32(struct worker *, struct objcore *, enum obj_attr, uint32_t);
110a114
> int ObjSetXID(struct worker *, struct objcore *, vxid_t);
116c120,122
< const char *sess_close_2str(enum sess_close sc, int want_desc);
——
> int PAN__DumpStruct(struct vsb *vsb, int block, int track, const void *ptr,
> const char *smagic, unsigned magic, const char *fmt, ...)
> v_printflike_(7,8);
126c132,133
< struct req *Req_New(const struct worker *, struct sess *);

```

```

-----
> int VRG_CheckBo(struct busyobj *);
> struct req *Req_New(struct sess *);
130c137
< void Req_Fail(struct req *req, enum sess_close reason);
-----
> void Req_Fail(struct req *req, stream_close_t reason);
131a139,140
> void Req_LogHit(struct worker *, struct req *, struct objcore *, intmax_t);
> const char *Req_LogStart(const struct worker *, struct req *);
150c159
< void HTC_RxPipeline(struct http_conn *htc, void *);
-----
> void HTC_RxPipeline(struct http_conn *htc, char *);
168c177
< void VCL_Recache(struct worker *, struct vcl **);
-----
> void VCL_Recache(const struct worker *, struct vcl **);
180c189,192
< void WS_Id(const struct ws *ws, char *id);
-----
> void WRK_Log(enum VSL_tag_e, const char *, ...);
> extern const size_t vpi_wrk_len;
> void VPI_wrk_init(struct worker *, void *, size_t);
> void VPI_Panic(struct vsb *, const struct wrk_vpi *, const struct vcl *);
184d195
< void WS_Rollback(struct ws *, uintptr_t);
186,187c197,200
< static inline unsigned
<
-----
> unsigned WS_ReservationOffset(const struct ws *ws);
> unsigned WS_ReqPipeline(struct ws *, const void *b, const void *e);
> void WS_Id(const struct ws *ws, char *id);
> void WS_Rollback(struct ws *, uintptr_t);
197a211,215
> struct stv_buffer;
> struct stv_buffer *STV_AllocBuf(struct worker *wrk, const struct stevedore *stv,
>     size_t size);
> void STV_FreeBuf(struct worker *wrk, struct stv_buffer **pstvbuf);
> void *STV_GetBufPtr(struct stv_buffer *stvbuf, size_t *psize);
改变 cli_cmds.h
43,44c43,44
<     "vcl.state <configname> [auto|cold|warm]",

```

```

<  "Force the state of the named configuration.",
---
>  "vcl.state <configname> auto|cold|warm",
>  "  Force the state of the named configuration.",
52c52
<  "Unload the named configurations and labels matching at least"
---
>  "  Unload the named configurations and labels matching at least"
88c88
<  "Only direct dependencies are listed, and VCLs with"
---
>  "  Only direct dependencies are listed, and VCLs with"
94c94
<  "vcl.show [-v] <configname>",
---
>  "vcl.show [-v] [<configname>]",
97c97
<  1, 2
---
>  0, 2
110,112c110,112
<  "A VCL label is like a UNIX symbolic link, "
<  "a name without substance, which points to another VCL.\n\n"
<  "Labels are mandatory whenever one VCL references another.",
---
>  "  A VCL label is like a UNIX symbolic link, "
>  "  a name without substance, which points to another VCL.\n\n"
>  "  Labels are mandatory whenever one VCL references another.",
137c137
<  "param.set <param> <value>",
---
>  "param.set [-j] <param> <value>",
139c139,144
<  "",
---
>  "  The JSON output is the same as ``param.show -j <param>`` and"
>  " contains the updated value as it would be represented by a"
>  " subsequent execution of ``param.show``.\n\n"
>  "  This can be useful to later verify that a parameter value didn't"
>  " change and to use the value from the JSON output to reset the"
>  " parameter to the desired value.",
165c170
<  "help [-j] [<command>]",
---

```

```

> "help [-j]<command>]",
285a291,297
> CLI_CMD(DEBUG_SHUTDOWN_DELAY,
> "debug.shutdown.delay",
> "debug.shutdown.delay",
> "Add a delay to the child process shutdown.",
> "",
> 1, 1
> )
288,289c300,301
< "debug.xid",
< "Examine or set XID.",
——
> "debug.xid [<xid> [<cacheSize>]]",
> "Examine or set XID. <cacheSize> defaults to 1.",
291c303
< 0, 1
——
> 0, 2
改变 common_param.h
4a5,13
> static inline int
>
> static inline int
>
> PARAM_BITMAP(vsl_mask_t, 256);
> PARAM_BITMAP(debug_t, DBG_Reserved);
> PARAM_BITMAP(experimental_t, EXPERIMENT_Reserved);
> PARAM_BITMAP(feature_t, FEATURE_Reserved);
> PARAM_BITMAP(vcc_feature_t, VCC_FEATURE_Reserved);
改变 debug_bits.h
13d12
< DEBUG_BIT(DROP_POOLS, drop_pools, "Drop thread pools (testing)")
增加头文件: experimental_bits.h
改变 feature_bits.h
27a28,30
> FEATURE_BIT(ESI_INCLUDE_ONERROR, esi_include_onerror,
> "Parse the onerror attribute of <esi:include> tags."
> )
35a39,46
> )
> FEATURE_BIT	TRACE, trace,
> "Enable VCL tracing by default (enable (be)req.trace). "
> "Required for tracing vcl_init / vcl_fini"

```

```

> )
> FEATURE_BIT(VCL_REQ_RESET,          vcl_req_reset,
>     "Stop processing client VCL once the client is gone. "
>     "When this happens MAIN.req_reset is incremented."
改变 h2_error.h
98a99,105
> H2_ERROR(
>     RAPID_RESET,
>     11,
>     1,
>     SC_RAPID_RESET,
>     "http/2 rapid reset detected"
> )
改变 locks.h
1d0
< LOCK(backend)
4a4
> LOCK(director)
减少头文件: mgt_vcc.h
改变 obj_attr.h
1,2c1,2
<  OBJ_FIXATTR(LEN, len, 8)
<  OBJ_FIXATTR(VXID, vxid, 4)
---
>  OBJ_FIXATTR(LEN, len, sizeof(uint64_t))
>  OBJ_FIXATTR(VXID, vxid, sizeof(uint64_t))
5c5
<  OBJ_FIXATTR(LASTMODIFIED, lastmodified, 8)
---
>  OBJ_FIXATTR(LASTMODIFIED, lastmodified, sizeof(double))
改变 oc_flags.h
1,2c1
< OC_FLAG(PURGED,          purged,          (1<<0))OC_FLAG(BUSY,          busy,          (1<<1))
< OC_FLAG(HFM,          hfm,          (1<<2))
---
> OC_FLAG(BUSY,          busy,          (1<<1))OC_FLAG(HFM,          hfm,          (1<<2))
4c3
< OC_FLAG(ABANDON,      abandon,      (1<<4))
---
> OC_FLAG(CANCEL,      cancel,          (1<<4))
改变 params.h
225c225
<     "48k",
---

```

```

>         "64k",
238,239c238,248
<     "Timeout for the childs replies to CLI requests from the "
<     "mgt_param."
____
>     "Timeout for the child's replies to CLI requests."
> )
> PARAM_SIMPLE(
>     startup_timeout,
>     timeout,
>     "0.000",
>     NULL,
>     "0.000",
>     "seconds",
>     "Alternative timeout for the initial worker process startup.\n"
>     "If cli_timeout is longer than startup_timeout, it is used instead."
289c298
<         "10.000",
____
>         "10s",
294c303,306
<         OBJ_STICKY
____
>         OBJ_STICKY,
>         NULL,
>         NULL,
>         "10s"
301c313
<         "0.000",
____
>         "0s",
307c319,322
<         OBJ_STICKY
____
>         OBJ_STICKY,
>         NULL,
>         NULL,
>         "0s"
314c329
<         "120.000",
____
>         "2m",
318c333,336
<         OBJ_STICKY

```

```

-----
>     OBJ_STICKY,
>     NULL,
>     NULL,
>     "2m"
328c346,349
<     " Allocated from workspace_thread.",
-----
>     " Allocated from workspace_thread."
>     " This parameter affects only io vectors used for client delivery."
>     " For backend fetches, the maximum number of io vectors (up to IOV_MAX)"
>     " is allocated from available workspace_thread memory.",
479c500
<     PARAM_SIMPLE(
-----
> PARAM_SIMPLE(
491c512
<     PLATFORM_DEPENDENT | PLATFORM_FLAGS
-----
>     DELAYED_EFFECT
555a577,585
>     pcre2_jit_compilation,
>     boolean,
>     NULL,
>     NULL,
>     "on",
>     "bool",
>     "Use the pcre2 JIT compiler if available."
> )
> PARAM_SIMPLE(
609c639
<     PARAM_SIMPLE(
-----
> PARAM_SIMPLE(
622c652
<     PLATFORM_DEPENDENT | PLATFORM_FLAGS
-----
>     DELAYED_EFFECT
708c738
<     PARAM_SIMPLE(
-----
> PARAM_SIMPLE(
720,721c750
<     " before this timeout is reached.",

```



```

<          PLATFORM_DEPENDENT | PLATFORM_FLAGS
——
>    " before this timeout is reached."
739a769,790
>          transit_buffer,
>          bytes,
>          "Ok",
>          NULL,
>          "Ok",
>          "bytes",
>    "The number of bytes which Varnish buffers for uncacheable "
>    "backend streaming fetches – in other words, how many bytes "
>    "Varnish reads from the backend ahead of what has been sent to "
>    "the client.\n"
>    "A zero value means no limit, the object is fetched as fast as "
>    "possible.\n\n"
>    "When dealing with slow clients, setting this parameter to "
>    "non-zero can prevent large uncacheable objects from being "
>    "stored in full when the intent is to simply stream them to the "
>    "client. As a result, a slow client transaction holds onto a "
>    "backend connection until the end of the delivery.\n\n"
>    "This parameter is the default to the VCL variable "
>    "`beresp.transit_buffer`, which can be used to control the "
>    "transit buffer per backend request."
> )
> PARAM_SIMPLE(
796c847
<          "4k",
——
>          "16k",
812,813c863
<          NULL,
<          "255b",
——
>          "65535b",          "255b",
834,843d883
<          vsm_space,
<          bytes,
<          "1M",
<          "1G",
<          "1M",
<          "bytes",
<    "DEPRECATED: This parameter is ignored.\n"
<    "There is no global limit on amount of shared memory now."

```

```

<)
< PARAM_SIMPLE(
848c888
<      "64k",
____
>      "96k",
859c899
<      "64k",
____
>      "96k",
872c912
<      "0.25k",
____
>      "384b",
948c988
<      "0",
____
>      "65535b",
952c992
<      "HTTP2 initial flow control window size."
____
>      "HTTP2 initial flow control window size.",
971a1012,1047
> PARAM_SIMPLE(
>      h2_rapid_reset,
>      timeout,
>      "0.000",
>      NULL,
>      "1.000",
>      "seconds",
>      "The upper threshold for how soon an http/2 RST_STREAM frame has "
>      "to be parsed after a HEADERS frame for it to be treated as "
>      "suspect and subjected to the rate limits specified by "
>      "h2_rapid_reset_limit and h2_rapid_reset_period.\n" H2_RR_INFO,
>      EXPERIMENTAL|DELAYED_EFFECT,
>)
> PARAM_SIMPLE(
>      h2_rapid_reset_limit,
>      uint,
>      "0",
>      NULL,
>      "100",
>      NULL,
>      "HTTP2 RST Allowance.\n"

```

```

> "Specifies the maximum number of allowed stream resets issued by\n"
> "a client over a time period before the connection is closed.\n"
> "Setting this parameter to 0 disables the limit.\n" H2_RR_INFO,
>     EXPERIMENTAL|DELAYED_EFFECT,
> )
> PARAM_SIMPLE(
>     h2_rapid_reset_period,
>     timeout,
>     "1.000",
>     NULL,
>     "60.000",
>     "seconds",
>     "HTTP2 sliding window duration for h2_rapid_reset_limit.\n" H2_RR_INFO,
>     EXPERIMENTAL|DELAYED_EFFECT|WIZARD,
> )
1133c1209
<     "10e-3",
____
>     ".01",
1226a1303
>     "sysconf(_SC_THREAD_STACK_MIN)",
1228,1229c1305
<     NULL,
<     "sysconf(_SC_THREAD_STACK_MIN)"
____
>     "80k"
1232a1309
>     tweak_string,
1235,1239c1312,1354
<     "Command used for compiling the C source code to a "
<     "dlopen(3) loadable object. Any occurrence of %s in "
<     "the string will be replaced with the source file name, "
<     "and %o will be replaced with the output file name.",
<     MUST_RELOAD
____
>     "The command used for compiling the C source code to a "
>     "dlopen(3) loadable object. The following expansions can "
>     "be used:\n\n"
>     "- %s: the source file name\n"
>     "- %o: the output file name\n"
>     "- %w: the cc_warnings parameter\n"
>     "- %d: the raw default cc_command\n"
>     "- %D: the expanded default cc_command\n"
>     "- %n: the working directory (-n option)\n"

```

```

>  "- %%: a percent sign\n"
>  "\n"
>  "Unknown percent expansion sequences are ignored, and to "
>  "avoid future incompatibilities percent characters should "
>  "be escaped with a double percent sequence.\n\n"
>  "The %d and %D expansions allow passing the parameter's "
>  "default value to a wrapper script to perform additional "
>  "processing.",
>      MUST_RELOAD | BUILD_OPTIONS,
>      NULL,
>      NULL,
>      "exec $CC $CFLAGS %w -shared -o %o %s"
> )
> PARAM_STRING(
>     cc_warnings,
>     tweak_string,
>     &mgt_cc_warn,
>     VCC_WARN,
>     "Warnings used when compiling the C source code with "
>     "the cc_command parameter. By default, VCL is compiled "
>     "with the same set of warnings as Varnish itself.",
>     MUST_RELOAD | BUILD_OPTIONS,
>     NULL,
>     NULL,
>     "-Wall -Werror"
> )
> PARAM_STRING(
>     h2_rxbuf_storage,
>     tweak_storage,
>     &mgt_stv_h2_rxbuf,
>     "Transient",
>     "The name of the storage backend that HTTP/2 receive buffers "
>     "should be allocated from.",
>     MUST_RESTART
1242a1358
>     tweak_string,
1251c1367,1371
<  "would work."
—
>  "would work.",
>     BUILD_OPTIONS,
>     NULL,
>     NULL,
>     "${sysconfdir}/varnish:${datadir}/varnish/vcl"

```

1254a1375

```
>         tweak_string,
1258,1288c1379,1383
<     "where VMODs are to be found."
< )
< PARAM_VCC(
<         vcc_err_unref,
<         "on",
<     "Unreferenced VCL objects result in error."
< )
< PARAM_VCC(
<         vcc_acl_pedantic,
<         "off",
<     "Insist that network numbers used in ACLs have an "
<     "all-zero host part, e.g. make 1.2.3.4/24 an error.\n"
<     "With this option set to off (the default), the host "
<     "part of network numbers is being fixed to all-zeroes "
<     "(e.g. the above changed to 1.2.3.0/24), a warning is "
<     "output during VCL compilation and any ACL entry hits "
<     "are logged with the fixed address as \"fixed: ...\" "
<     "after the original VCL entry.\n"
<     "With this option set to on, any ACL entries with non-zero "
<     "host parts cause VCL compilation to fail."
< )
< PARAM_VCC(
<         vcc_allow_inline_c,
<         "off",
<     "Allow inline C code in VCL."
< )
< PARAM_VCC(
<         vcc_unsafe_path,
<         "on",
<     "Allow '/' in vmod & include paths.\n"
<     "Allow 'import ... from ...'."
—
>     "where VMODs are to be found.",
>     BUILD_OPTIONS,
>     NULL,
>     NULL,
>     "${libdir}/varnish/vmods"
1290,1291c1385,1386
< PARAM_PCRE(
<         pcre_match_limit,
—
```

```

> PARAM_PCRE2(
>     pcre2_match_limit,
1295,1297c1390,1392
<     "The limit for the number of calls to the internal match()"
<     " function in pcre_exec().\n\n"
<     "(See: PCRE_EXTRA_MATCH_LIMIT in pcre docs.)\n\n"
—
>     "The limit for the number of calls to the internal match"
>     " logic in pcre2_match().\n\n"
>     "(See: pcre2_set_match_limit() in pcre2 docs.)\n\n"
1301,1303c1396,1398
< PARAM_PCRE(
<     pcre_match_limit_recursion,
<     match_recursion,
—
> PARAM_PCRE2(
>     pcre2_depth_limit,
>     depth,
1306,1308c1401,1403
<     "The recursion depth-limit for the internal match() function"
<     " in a pcre_exec().\n\n"
<     "(See: PCRE_EXTRA_MATCH_LIMIT_RECURSION in pcre docs.)\n\n"
—
>     "The recursion depth-limit for the internal match logic"
>     " in a pcre2_match().\n\n"
>     "(See: pcre2_set_depth_limit() in pcre2 docs.)\n\n"
1310c1405
<     " by PCRE for certain classes of regular expressions.\n\n"
—
>     " by PCRE2 for certain classes of regular expressions.\n\n"
1315,1316c1410
<     " messages with regexp errors -27 or -21.\n\n"
<     "Testcase r01576 can be useful when tuning this parameter."
—
>     " messages."
1318c1412,1417
< PARAM(
—
> PARAM_ALIAS(deprecated_dummy, debug)
> PARAM_ALIAS(vcc_err_unref, vcc_feature)
> PARAM_ALIAS(vcc_allow_inline_c, vcc_feature)
> PARAM_ALIAS(vcc_unsafe_path, vcc_feature)
> PARAM_PRE
> PARAM_BITS(

```

1320,1324c1419,1420

```
<      debug,  
<      NULL,  
<      NULL,  
<      NULL,  
<      NULL,
```

——

```
>      debug_bits,  
>      "none",
```

1326,1341c1422,1435

```
<  "    none Disable all debugging\n"  
<  "\n"  
<  "Use +/- prefix to set/reset individual bits:\n"  
<  "    req_state  VSL Request state engine\n"  
<  "    workspace  VSL Workspace operations\n"  
<  "    waiter     VSL Waiter internals\n"  
<  "    waitinglist VSL Waitinglist events\n"  
<  "    syncvsl    Make VSL synchronous\n"  
<  "    hashedge   Edge cases in Hash\n"  
<  "    vclrelRapid VCL release\n"  
<  "    lurker     VSL Ban lurker\n"  
<  "    esi_chop   Chop ESI fetch to bits\n"  
<  "    flush_head Flush after http1 head\n"  
<  "    vtc_mode   Varnishtest Mode"
```

```
< )
```

```
< PARAM(  
——
```

```
>  "\tnone\tDisable all debugging\n\n"  
>  "Use +/- prefix to set/reset individual bits:")  
>  PARAM_POST  
> PARAM_PRE  
> PARAM_BITS(  
>      experimental,  
>      experimental_bits,  
>      "none",  
>  "Enable/Disable experimental features.\n"  
>  "\tnone\tDisable all experimental features\n\n"  
>  "Use +/- prefix to set/reset individual bits:")  
>  PARAM_POST  
> PARAM_PRE  
> PARAM_BITS(  
1343,1347c1437,1439
```

```
<      feature,  
<      NULL,
```

```

<      NULL,
<      NULL,
<      NULL,
----
>      feature_bits,
>      "+validate_headers,"
>      "+vcl_req_reset",
1349,1360c1441,1457
<      "      none  Disable all features.\n"
<      "\n"
<      "Use +/- prefix to enable/disable individual feature:\n"
<      "      short_panicShort panic message.\n"
<      "      wait_silo  Wait for persistent silo.\n"
<      "      no_coredump    No coredumps.\n"
<      "      esi_ignore_https  Treat HTTPS as HTTP in ESI:includes\n"
<      "      esi_disable_xml_check Don't check if body looks like XML\n"
<      "      esi_ignore_other_elements  Ignore non-esi XML-elements\n"
<      "      esi_remove_bom  Remove UTF-8 BOM"
< )
< PARAM(
----
>      "\tdefault\tSet default value\n"
>      "\tnone\tDisable all features.\n\n"
>      "Use +/- prefix to enable/disable individual feature:")
>      PARAM_POST
> PARAM_PRE
> PARAM_BITS(
>      vcc_feature,
>      vcc_feature_bits,
>      "+err_unref,"
>      "+unsafe_path",
>      "Enable/Disable various VCC behaviors.\n"
>      "\tdefault\tSet default value\n"
>      "\tnone\tDisable all behaviors.\n\n"
>      "Use +/- prefix to enable/disable individual behavior:")
>      PARAM_POST
> PARAM_PRE
> PARAM_BITS(
1363,1366c1460,1473
<      NULL,
<      NULL,
<      "default",
<      NULL,
----

```



```

> "-Debug,"
> "-ExpKill,"
> "-H2RxBdy,"
> "-H2RxHdr,"
> "-H2TxBdy,"
> "-H2TxHdr,"
> "-Hash,"
> "-ObjHeader,"
> "-ObjProtocol,"
> "-ObjReason,"
> "-ObjStatus,"
> "-VdpAcct,"
> "-VfpAcct,"
> "-WorkThread",
1368,1372c1475,1478
< "      default      Set default value\n"
< "\n"
< "Use +/- prefix in front of VSL tag name to unmask/mask "
< "individual VSL messages."
< )
---
> "\tdefault\tSet default value\n"
> "\nUse +/- prefix in front of VSL tag name to unmask/mask "
> "individual VSL messages.")
> PARAM_POST
增加头文件: req_bereq_flags.h
改变 req_flags.h
2a3
> REQ_FLAG(hash_ignore_vary, 1, 1, "")
5,6d5
< REQ_FLAG(is_hitmiss, 1, 0, "")
< REQ_FLAG(is_hitpass, 1, 0, "")
9a9
> REQ_FLAG(req_reset, 0, 0, "")
改变 sess_close.h
5c5
< SESS_CLOSE(RX_BODY, rx_body, 1, "Failure receiving req.body")
---
> SESS_CLOSE(RX_BODY, rx_body, 1, "Failure receiving body")
18a19
> SESS_CLOSE(RAPID_RESET, rapid_reset, 1, "HTTP2 rapid reset")
改变 symbol_kind.h
2a3
> VCC_KIND(ALIAS, alias)

```

改变 vas.h

1c1

```
< const char * vstrerror(int e);
```

```
> const char * VAS_errtxt(int e);
```

改变 vav.h

1a2

```
> char **VAV_ParseTxt(const char *b, const char *e, int *argc, int flag);
```

增加头文件: vcc_feature_bits.h

增加头文件: vcl_context.h

改变 vcl.h

21d20

```
< extern const struct vrt_type VCL_TYPE_STRING_LIST[1];
```

改变 vcli.h

4c4

```
<     char reponse[CLI_AUTH_RESPONSE_LEN + 1]);
```

```
>     char response[CLI_AUTH_RESPONSE_LEN + 1]);
```

改变 vdef.h

0a1,3

```
> int __gcov_flush(void);
```

```
> void __gcov_dump(void);
```

```
> int __llvm_gcov_flush(void);
```

增加头文件: vnum.h

改变 vre.h

3c3

```
< extern const unsigned VRE_has_jit;
```

```
> extern const int VRE_ERROR_NOMATCH;
```

5,8c5,11

```
< extern const unsigned VRE_NOTEMPTY;
```

```
< vre_t *VRE_compile(const char *, unsigned, const char **, int *);
```

```
< int VRE_exec(const vre_t *code, const char *subject, int length,
```

```
<     int startoffset, int options, int *ovector, int oveclsize,
```

```
> vre_t *VRE_compile(const char *, unsigned, int *, int *, unsigned);
```

```
> vre_t *VRE_export(const vre_t *, size_t *);
```

```
> int VRE_error(struct vsb *, int err);
```

```
> int VRE_match(const vre_t *code, const char *subject, size_t length,
```

```
>     int options, const volatile struct vre_limits *lim);
```

```
> int VRE_capture(const vre_t *code, const char *subject, size_t length,
```

```
>     int options, txt *groups, size_t count,
```

9a13,14

```
> int VRE_sub(const vre_t *code, const char *subject, const char *replacement,
```

```

> struct vsb *vsb, const volatile struct vre_limits *lim, int all);
增加头文件: vre_pcre2.h
改变 vrt.h
3a4
> struct lock;
5a7
> struct stream_close;
9a12
> struct vcl_sub;
17a21
> struct wrk_vpi;
19c23,24
< struct vcl_sub;
——
> extern const struct strands *vrt_null_strands;
> extern const struct vrt_blob *vrt_null_blob;
23a29
> int VRT_INT_is_valid(VCL_INT);
26a33
> int VRT_REAL_is_valid(VCL_REAL);
36d42
< VCL_STRING VRT_CollectStrands(VRT_CTX, VCL_STRANDS);
43d48
< VCL_STRING VRT_CollectString(VRT_CTX, const char *p, ...);
47a53
> VCL_STRING VRT_STRANDS_string(VRT_CTX, VCL_STRANDS);
51c57
< VCL_BACKEND VRT_new_backend(VRT_CTX, const struct vrt_backend *);
——
> VCL_BACKEND VRT_new_backend(VRT_CTX, const struct vrt_backend *, VCL_BACKEND);
53c59
< struct vsmw_cluster *, const struct vrt_backend *);
——
> struct vsmw_cluster *, const struct vrt_backend *, VCL_BACKEND);
65c71,72
< VCL_VOID VRT_SetHdr(VRT_CTX, VCL_HEADER, const char *, ...);
——
> VCL_VOID VRT_UnsetHdr(VRT_CTX, VCL_HEADER);
> VCL_VOID VRT_SetHdr(VRT_CTX, VCL_HEADER, const char *pfx, VCL_STRANDS);
68c75,76
< VCL_VOID VRT_fail(VRT_CTX, const char *fmt, ...) v_printflike_(2,3);
——
> VCL_VOID VRT_trace(VRT_CTX, VCL_BOOL);
> VCL_VOID VRT_fail(VRT_CTX, const char *fmt, ...) v_printflike_(2,3) v_cold_;

```

77a86

```
> void VRT_StaticDirector(VCL_BACKEND);
```

79a89

```
> void VRT_Assign_Backend(VCL_BACKEND *dst, VCL_BACKEND src);
```

98,99d107

```
< extern const void * const vrt_magic_string_end;
```

```
< extern const void * const vrt_magic_string_unset;
```

改变 vrt_obj.h

1,5d0

```
< VCL_IP VRT_r_local_ip(VRT_CTX);
```

```
< VCL_STRING VRT_r_local_endpoint(VRT_CTX);
```

```
< VCL_STRING VRT_r_local_socket(VRT_CTX);
```

```
< VCL_IP VRT_r_remote_ip(VRT_CTX);
```

```
< VCL_IP VRT_r_client_ip(VRT_CTX);
```

7,8c2,3

```
< void VRT_l_client_identity(VRT_CTX, const char *, ...);
```

```
< VCL_IP VRT_r_server_ip(VRT_CTX);
```

——

```
> void VRT_l_client_identity(VRT_CTX, const char *, VCL_STRANDS);
```

```
> VCL_IP VRT_r_client_ip(VRT_CTX);
```

10a6,10

```
> VCL_IP VRT_r_server_ip(VRT_CTX);
```

```
> VCL_IP VRT_r_remote_ip(VRT_CTX);
```

```
> VCL_STRING VRT_r_local_endpoint(VRT_CTX);
```

```
> VCL_IP VRT_r_local_ip(VRT_CTX);
```

```
> VCL_STRING VRT_r_local_socket(VRT_CTX);
```

12,22c12,16

```
< VCL_STRING VRT_r_req_method(VRT_CTX);
```

```
< void VRT_l_req_method(VRT_CTX, const char *, ...);
```

```
< VCL_BLOB VRT_r_req_hash(VRT_CTX);
```

```
< VCL_STRING VRT_r_req_url(VRT_CTX);
```

```
< void VRT_l_req_url(VRT_CTX, const char *, ...);
```

```
< VCL_STRING VRT_r_req_proto(VRT_CTX);
```

```
< void VRT_l_req_proto(VRT_CTX, const char *, ...);
```

```
< void VRT_u_req_http(VRT_CTX);
```

```
< VCL_INT VRT_r_req_restarts(VRT_CTX);
```

```
< VCL_STEVEDORE VRT_r_req_storage(VRT_CTX);
```

```
< void VRT_l_req_storage(VRT_CTX, VCL_STEVEDORE);
```

——

```
> VCL_BACKEND VRT_r_req_backend_hint(VRT_CTX);
```

```
> void VRT_l_req_backend_hint(VRT_CTX, VCL_BACKEND);
```

```
> VCL_BOOL VRT_r_req_can_gzip(VRT_CTX);
```

```
> VCL_BOOL VRT_r_req_esi(VRT_CTX);
```

```
> void VRT_l_req_esi(VRT_CTX, VCL_BOOL);
```

24,25d17

```
< VCL_DURATION VRT_r_req_ttl(VRT_CTX);  
< void VRT_l_req_ttl(VRT_CTX, VCL_DURATION);
```

28,35c20

```
< VCL_STRING VRT_r_req_xid(VRT_CTX);  
< VCL_BOOL VRT_r_req_esi(VRT_CTX);  
< void VRT_l_req_esi(VRT_CTX, VCL_BOOL);  
< VCL_BOOL VRT_r_req_can_gzip(VRT_CTX);  
< VCL_BACKEND VRT_r_req_backend_hint(VRT_CTX);  
< void VRT_l_req_backend_hint(VRT_CTX, VCL_BACKEND);  
< VCL_BOOL VRT_r_req_hash_ignore_busy(VRT_CTX);  
< void VRT_l_req_hash_ignore_busy(VRT_CTX, VCL_BOOL);  
____
```

```
> VCL_BLOB VRT_r_req_hash(VRT_CTX);
```

37a23,27

```
> VCL_BOOL VRT_r_req_hash_ignore_busy(VRT_CTX);  
> void VRT_l_req_hash_ignore_busy(VRT_CTX, VCL_BOOL);  
> VCL_BOOL VRT_r_req_hash_ignore_vary(VRT_CTX);  
> void VRT_l_req_hash_ignore_vary(VRT_CTX, VCL_BOOL);  
> void VRT_u_req_http(VRT_CTX);
```

39a30,45

```
> VCL_STRING VRT_r_req_method(VRT_CTX);  
> void VRT_l_req_method(VRT_CTX, const char *, VCL_STRANDS);  
> VCL_STRING VRT_r_req_proto(VRT_CTX);  
> void VRT_l_req_proto(VRT_CTX, const char *, VCL_STRANDS);  
> VCL_INT VRT_r_req_restarts(VRT_CTX);  
> VCL_STEVEDORE VRT_r_req_storage(VRT_CTX);  
> void VRT_l_req_storage(VRT_CTX, VCL_STEVEDORE);  
> VCL_TIME VRT_r_req_time(VRT_CTX);  
> VCL_BOOL VRT_r_req_trace(VRT_CTX);  
> void VRT_l_req_trace(VRT_CTX, VCL_BOOL);  
> VCL_STRING VRT_r_req_transport(VRT_CTX);  
> VCL_DURATION VRT_r_req_ttl(VRT_CTX);  
> void VRT_l_req_ttl(VRT_CTX, VCL_DURATION);  
> VCL_STRING VRT_r_req_url(VRT_CTX);  
> void VRT_l_req_url(VRT_CTX, const char *, VCL_STRANDS);  
> VCL_INT VRT_r_req_xid(VRT_CTX);
```

41d46

```
< VCL_STRING VRT_r_req_top_url(VRT_CTX);
```

42a48,49

```
> VCL_TIME VRT_r_req_top_time(VRT_CTX);  
> VCL_STRING VRT_r_req_top_url(VRT_CTX);
```

44,45d50

```
< VCL_STRING VRT_r_bereq_xid(VRT_CTX);
```

```

< VCL_INT VRT_r_bereq_retries(VRT_CTX);
47a53,54
> VCL_DURATION VRT_r_bereq_between_bytes_timeout(VRT_CTX);
> void VRT_l_bereq_between_bytes_timeout(VRT_CTX, VCL_DURATION);
49,57d55
< VCL_BLOB VRT_r_bereq_hash(VRT_CTX);
< VCL_STRING VRT_r_bereq_method(VRT_CTX);
< void VRT_l_bereq_method(VRT_CTX, const char *, ...);
< VCL_STRING VRT_r_bereq_url(VRT_CTX);
< void VRT_l_bereq_url(VRT_CTX, const char *, ...);
< VCL_STRING VRT_r_bereq_proto(VRT_CTX);
< void VRT_l_bereq_proto(VRT_CTX, const char *, ...);
< void VRT_u_bereq_http(VRT_CTX);
< VCL_BOOL VRT_r_bereq_uncacheable(VRT_CTX);
62,63c60,61
< VCL_DURATION VRT_r_bereq_between_bytes_timeout(VRT_CTX);
< void VRT_l_bereq_between_bytes_timeout(VRT_CTX, VCL_DURATION);
——
> VCL_BLOB VRT_r_bereq_hash(VRT_CTX);
> void VRT_u_bereq_http(VRT_CTX);
66a65,76
> VCL_STRING VRT_r_bereq_method(VRT_CTX);
> void VRT_l_bereq_method(VRT_CTX, const char *, VCL_STRANDS);
> VCL_STRING VRT_r_bereq_proto(VRT_CTX);
> void VRT_l_bereq_proto(VRT_CTX, const char *, VCL_STRANDS);
> VCL_INT VRT_r_bereq_retries(VRT_CTX);
> VCL_TIME VRT_r_bereq_time(VRT_CTX);
> VCL_BOOL VRT_r_bereq_trace(VRT_CTX);
> void VRT_l_bereq_trace(VRT_CTX, VCL_BOOL);
> VCL_BOOL VRT_r_bereq_uncacheable(VRT_CTX);
> VCL_STRING VRT_r_bereq_url(VRT_CTX);
> void VRT_l_bereq_url(VRT_CTX, const char *, VCL_STRANDS);
> VCL_INT VRT_r_bereq_xid(VRT_CTX);
68,75c78,82
< void VRT_l_beresp_body(VRT_CTX, enum lbody_e, const char *, ...);
< VCL_STRING VRT_r_beresp_proto(VRT_CTX);
< void VRT_l_beresp_proto(VRT_CTX, const char *, ...);
< VCL_INT VRT_r_beresp_status(VRT_CTX);
< void VRT_l_beresp_status(VRT_CTX, VCL_INT);
< VCL_STRING VRT_r_beresp_reason(VRT_CTX);
< void VRT_l_beresp_reason(VRT_CTX, const char *, ...);
< void VRT_u_beresp_http(VRT_CTX);
——
> VCL_DURATION VRT_r_beresp_age(VRT_CTX);

```

```

> VCL_BACKEND VRT_r_beresp_backend(VRT_CTX);
> VCL_IP VRT_r_beresp_backend_ip(VRT_CTX);
> VCL_STRING VRT_r_beresp_backend_name(VRT_CTX);
> void VRT_l_beresp_body(VRT_CTX, enum lbody_e, const char *, const void *);
78,81d84
< VCL_BOOL VRT_r_beresp_do_stream(VRT_CTX);
< void VRT_l_beresp_do_stream(VRT_CTX, VCL_BOOL);
< VCL_BOOL VRT_r_beresp_do_gzip(VRT_CTX);
< void VRT_l_beresp_do_gzip(VRT_CTX, VCL_BOOL);
84,89c87,92
< VCL_BOOL VRT_r_beresp_was_304(VRT_CTX);
< VCL_BOOL VRT_r_beresp_uncacheable(VRT_CTX);
< void VRT_l_beresp_uncacheable(VRT_CTX, VCL_BOOL);
< VCL_DURATION VRT_r_beresp_ttl(VRT_CTX);
< void VRT_l_beresp_ttl(VRT_CTX, VCL_DURATION);
< VCL_DURATION VRT_r_beresp_age(VRT_CTX);
-----
> VCL_BOOL VRT_r_beresp_do_gzip(VRT_CTX);
> void VRT_l_beresp_do_gzip(VRT_CTX, VCL_BOOL);
> VCL_BOOL VRT_r_beresp_do_stream(VRT_CTX);
> void VRT_l_beresp_do_stream(VRT_CTX, VCL_BOOL);
> VCL_STRING VRT_r_beresp_filters(VRT_CTX);
> void VRT_l_beresp_filters(VRT_CTX, const char *, VCL_STRANDS);
91a95
> void VRT_u_beresp_http(VRT_CTX);
94,96c98,103
< VCL_BACKEND VRT_r_beresp_backend(VRT_CTX);
< VCL_STRING VRT_r_beresp_backend_name(VRT_CTX);
< VCL_IP VRT_r_beresp_backend_ip(VRT_CTX);
-----
> VCL_STRING VRT_r_beresp_proto(VRT_CTX);
> void VRT_l_beresp_proto(VRT_CTX, const char *, VCL_STRANDS);
> VCL_STRING VRT_r_beresp_reason(VRT_CTX);
> void VRT_l_beresp_reason(VRT_CTX, const char *, VCL_STRANDS);
> VCL_INT VRT_r_beresp_status(VRT_CTX);
> void VRT_l_beresp_status(VRT_CTX, VCL_INT);
100,107c107,115
< void VRT_l_beresp_storage_hint(VRT_CTX, const char *, ...);
< VCL_STRING VRT_r_beresp_filters(VRT_CTX);
< void VRT_l_beresp_filters(VRT_CTX, const char *, ...);
< VCL_STRING VRT_r_obj_proto(VRT_CTX);
< VCL_INT VRT_r_obj_status(VRT_CTX);
< VCL_STRING VRT_r_obj_reason(VRT_CTX);
< VCL_INT VRT_r_obj_hits(VRT_CTX);

```

```

< VCL_DURATION VRT_r_obj_ttl(VRT_CTX);
-----
> void VRT_l_beresp_storage_hint(VRT_CTX, const char *, VCL_STRANDS);
> VCL_TIME VRT_r_beresp_time(VRT_CTX);
> VCL_BYTES VRT_r_beresp_transit_buffer(VRT_CTX);
> void VRT_l_beresp_transit_buffer(VRT_CTX, VCL_BYTES);
> VCL_DURATION VRT_r_beresp_ttl(VRT_CTX);
> void VRT_l_beresp_ttl(VRT_CTX, VCL_DURATION);
> VCL_BOOL VRT_r_beresp_uncacheable(VRT_CTX);
> void VRT_l_beresp_uncacheable(VRT_CTX, VCL_BOOL);
> VCL_BOOL VRT_r_beresp_was_304(VRT_CTX);
108a117
> VCL_BOOL VRT_r_obj_can_esi(VRT_CTX);
109a119
> VCL_INT VRT_r_obj_hits(VRT_CTX);
111c121,123
< VCL_BOOL VRT_r_obj_uncacheable(VRT_CTX);
-----
> VCL_STRING VRT_r_obj_proto(VRT_CTX);
> VCL_STRING VRT_r_obj_reason(VRT_CTX);
> VCL_INT VRT_r_obj_status(VRT_CTX);
113c125,127
< VCL_BOOL VRT_r_obj_can_esi(VRT_CTX);
-----
> VCL_TIME VRT_r_obj_time(VRT_CTX);
> VCL_DURATION VRT_r_obj_ttl(VRT_CTX);
> VCL_BOOL VRT_r_obj_uncacheable(VRT_CTX);
115,122c129
< void VRT_l_resp_body(VRT_CTX, enum lbody_e, const char *, ...);
< VCL_STRING VRT_r_resp_proto(VRT_CTX);
< void VRT_l_resp_proto(VRT_CTX, const char *, ...);
< VCL_INT VRT_r_resp_status(VRT_CTX);
< void VRT_l_resp_status(VRT_CTX, VCL_INT);
< VCL_STRING VRT_r_resp_reason(VRT_CTX);
< void VRT_l_resp_reason(VRT_CTX, const char *, ...);
< void VRT_u_resp_http(VRT_CTX);
-----
> void VRT_l_resp_body(VRT_CTX, enum lbody_e, const char *, const void *);
125d131
< VCL_BOOL VRT_r_resp_is_streaming(VRT_CTX);
127c133,142
< void VRT_l_resp_filters(VRT_CTX, const char *, ...);
-----
> void VRT_l_resp_filters(VRT_CTX, const char *, VCL_STRANDS);

```



```

> void VRT_u_resp_http(VRT_CTX);
> VCL_BOOL VRT_r_resp_is_streaming(VRT_CTX);
> VCL_STRING VRT_r_resp_proto(VRT_CTX);
> void VRT_l_resp_proto(VRT_CTX, const char *, VCL_STRANDS);
> VCL_STRING VRT_r_resp_reason(VRT_CTX);
> void VRT_l_resp_reason(VRT_CTX, const char *, VCL_STRANDS);
> VCL_INT VRT_r_resp_status(VRT_CTX);
> void VRT_l_resp_status(VRT_CTX, VCL_INT);
> VCL_TIME VRT_r_resp_time(VRT_CTX);
129c144,147
< VCL_STRING VRT_r_sess_xid(VRT_CTX);

```

```

——
> VCL_DURATION VRT_r_sess_idle_send_timeout(VRT_CTX);
> void VRT_l_sess_idle_send_timeout(VRT_CTX, VCL_DURATION);
> VCL_DURATION VRT_r_sess_send_timeout(VRT_CTX);
> void VRT_l_sess_send_timeout(VRT_CTX, VCL_DURATION);
134,140c152,155
< VCL_DURATION VRT_r_sess_send_timeout(VRT_CTX);
< void VRT_l_sess_send_timeout(VRT_CTX, VCL_DURATION);
< VCL_DURATION VRT_r_sess_idle_send_timeout(VRT_CTX);
< void VRT_l_sess_idle_send_timeout(VRT_CTX, VCL_DURATION);
< int64_t VRT_stevedore_free_space(VCL_STEVEDORE);
< int64_t VRT_stevedore_used_space(VCL_STEVEDORE);
< unsigned VRT_stevedore_happy(VCL_STEVEDORE);

```

```

——
> VCL_INT VRT_r_sess_xid(VRT_CTX);
> VCL_BYTES VRT_stevedore_free_space(VCL_STEVEDORE);
> VCL_BYTES VRT_stevedore_used_space(VCL_STEVEDORE);
> VCL_BOOL VRT_stevedore_happy(VCL_STEVEDORE);

```

改变 vsa.h

```

10,12c10,13
< struct suckaddr *VSA_Clone(const struct suckaddr *sua);
< struct suckaddr *VSA_getsockname(int, void *, size_t);
< struct suckaddr *VSA_getpeername(int, void *, size_t);
——
> const struct suckaddr *VSA_Clone(const struct suckaddr *sua);
> const struct suckaddr *VSA_getsockname(int, void *, size_t);
> const struct suckaddr *VSA_getpeername(int, void *, size_t);
> void VSA_free(const struct suckaddr **);
15,16c16,20
< struct suckaddr *VSA_Malloc(const void *s, unsigned sal);
< struct suckaddr *VSA_Build(void *d, const void *s, unsigned sal);
——
> const struct suckaddr *VSA_Malloc(const void *s, unsigned sal);

```

```
> const struct suckaddr *VSA_Build(void *d, const void *s, unsigned sal);
> const struct suckaddr * VSA_BuildFAP(void *d, sa_family_t fam,
>     const void *a, unsigned al, const void *p, unsigned pl);
> int VSA_GetPtr(const struct suckaddr *sua, const unsigned char ** dst);
```

改变 vsb.h

2d1

```
< struct vsb*VSB_new(struct vsb *, char *, int, int) v_deprecated_;
```

17d15

```
< void          VSB_delete(struct vsb *) v_deprecated_;
```

改变 vsl.h

26c26

```
<     vsl_check_valid = 2
```

```
>     vsl_check_valid      = 2
```

改变 vsl_tags.h

53,63d52

```
< SLTM(BackendReuse, 0, "Backend connection put up for reuse",
<     "Logged when a backend connection is put up for reuse by a later"
<     " connection.\n\n"
<     "The format is::\n\n"
<     "\t%d %s\n"
<     "\t|  |\n"
<     "\t| +- Backend display name\n"
<     "\t+----- Connection file descriptor\n"
<     "\n"
<     NOSUP_NOTICE
< )
```

95,106d83

```
< SLTM(Backend, 0, "Backend selected",
<     "Logged when a connection is selected for handling a backend"
<     " request.\n\n"
<     "The format is::\n\n"
<     "\t%d %s %s\n"
<     "\t|  |  |\n"
<     "\t|  | +- Backend display name\n"
<     "\t|  +----- VCL name\n"
<     "\t+----- Connection file descriptor\n"
<     "\n"
<     NOSUP_NOTICE
< )
```

169c146

```
<     "Logs VCL ACL evaluation results.\n\n"
```

```
>     "ACLs with the `+log` flag emits this record with the result.\n\n"
```

171,176c148,153

```
<  "\t%s [%s [%s [fixed: %s]]]\n"
<  "\t|   |   |           |\n"
<  "\t|   |   |           +- Fixed entry (see vcc_acl_pedantic parameter)\n"
<  "\t|   |   +----- Matching entry (only for MATCH)\n"
<  "\t|   +----- Name of the ACL for MATCH or NO_MATCH\n"
<  "\t+----- MATCH, NO_MATCH or NO_FAM\n"
```

```
>  "\t%s %s [%s [fixed: %s]]\n"
>  "\t|   |   |           |\n"
>  "\t|   |   |           +- Fixed entry (see acl +pedantic flag)\n"
>  "\t|   |   +----- Matching entry (only for MATCH)\n"
>  "\t|   +----- Name of the ACL\n"
>  "\t+----- MATCH or NO_MATCH\n"
```

180d156

```
<  "NO_FAM denotes a missing address family and should not occur.\n"
```

216,221c192,199

```
<  "\t%u %f %f %f\n"
<  "\t|   |   |   |\n"
<  "\t|   |   |   +- Keep period\n"
<  "\t|   |   +----- Grace period\n"
<  "\t|   +----- Remaining TTL\n"
<  "\t+----- VXID of the object\n"
```

```
>  "\t%u %f %f %f [%u [%u]]\n"
>  "\t|   |   |   |   |\n"
>  "\t|   |   |   |   +- Content length\n"
>  "\t|   |   |   +----- Fetched so far\n"
>  "\t|   |   +----- Keep period\n"
>  "\t|   +----- Grace period\n"
>  "\t+----- Remaining TTL\n"
>  "\t+----- VXID of the object\n"
```

356,360c334,339

```
<  "\t%s %d %s\n"
<  "\t|   |   |\n"
<  "\t|   +- Reason\n"
<  "\t|   +----- Child vxid\n"
<  "\t+----- Child type (\\"req\" or \\"breq\")\n"
```

```
>  "\t%s %d %s [%u]\n"
>  "\t|   |   |\n"
>  "\t|   |   +- Child task sub-level\n"
>  "\t|   +----- Reason\n"
>  "\t+----- Child vxid\n"
```

```

> "\t+----- Child type (\\"sess\\", \\"req\\" or \\"breq\\")\n"
366,370c345,350
< "\t%s %d %s\n"
< "\t| | |\n"
< "\t| | +- Reason\n"
< "\t| +----- Parent vxid\n"
< "\t+----- Type (\\"sess\\", \\"req\\" or \\"breq\\")\n"
-----
> "\t%s %d %s [%u]\n"
> "\t| | | |\n"
> "\t| | | +- Task sub-level\n"
> "\t| | +----- Reason\n"
> "\t| +----- Parent vxid\n"
> "\t+----- Type (\\"sess\\", \\"req\\" or \\"breq\\")\n"
459,469d438
<)
< SLTM(BackendStart, 0, "Backend request start",
< "Start of backend processing. Logs the backend IP address and port"
< " number.\n\n"
< "The format is::\n\n"
< "\t%s %s\n"
< "\t| |\n"
< "\t| +- Backend Port number\n"
< "\t+----- Backend IP4/6 address\n"
< "\n"
< NOSUP_NOTICE
改变 vsl_tags_http.h
26c26
< SLTH(Unset, HTTP_HDR_UNSET, 0, 0, "unset header",
-----
> SLTH(Unset, HTTP_HDR_UNSET, 1, 1, "unset header",
改变 vsm.h
17,18c17,18
< const char *class, const char *ident);
< char *VSM_Dup(struct vsm*, const char *class, const char *ident);
-----
> const char *category, const char *ident);
> char *VSM_Dup(struct vsm*, const char *category, const char *ident);
增加头文件: vte.h

```

nettle-devel

改变 hmac.h

95a96,105

```
> struct hmac_sm3_ctx HMAC_CTX(struct sm3_ctx);
```

```
> void
```

```
> hmac_sm3_set_key(struct hmac_sm3_ctx *ctx,
```

```
>         size_t key_length, const uint8_t *key);
```

```
> void
```

```
> hmac_sm3_update(struct hmac_sm3_ctx *ctx,
```

```
>         size_t length, const uint8_t *data);
```

```
> void
```

```
> hmac_sm3_digest(struct hmac_sm3_ctx *ctx,
```

```
>         size_t length, uint8_t *digest);
```

改变 nettle-meta.h

42a43

```
> extern const struct nettle_hash nettle_sm3;
```

68a70

```
> extern const struct nettle_mac nettle_hmac_sm3;
```

增加头文件: sm2.h

增加头文件: sm3.h

libtar-devel

改变 libtar.h

52a53

```
> void free_longlink_longname(struct tar_header th_buf);
```

eom-devel

改变 eom-dialog.h

5a6

```
> void      eom_dialog_delete      (EomDialog *dialog);
```

libdwarf-devel

改变 libdwarf.h

6,7c6

```
< struct Dwarf_Loc_c_s;
```

```
< struct Dwarf_Locdesc_c_s;
```

——

```
> struct  Dwarf_Locdesc_c_s;
```

8a8

```
> struct Dwarf_Gnu_Index_Head_s;
```

```

11c11,12
< int dwarf_frame_get_reg_register(struct Dwarf_Regtable_Entry3_s *reg_in,
-----
> int dwarf_frame_get_reg_register(
>     struct Dwarf_Regtable_Entry3_s *reg_in,
15c16,17
< int dwarf_frame_get_reg_expression(struct Dwarf_Regtable_Entry3_s *reg_in,
-----
> int dwarf_frame_get_reg_expression(
>     struct Dwarf_Regtable_Entry3_s *reg_in,
20a23,32
>     Dwarf_Obj_Access_Interface;
> struct Dwarf_Obj_Access_Sections {
>     Dwarf_Addr      addr;
>     Dwarf_Unsigned type;
>     Dwarf_Unsigned size;
>     const char*     name;
>     Dwarf_Unsigned link;
>     Dwarf_Unsigned info;
>     Dwarf_Unsigned entrysize;
> };
23c35
<     unsigned          ,
-----
>     unsigned int      ,
25c37
<     unsigned          ,
-----
>     unsigned int      ,
32a45,59
> int dwarf_init_path_dl(const char * ,
>     char *             ,
>     unsigned int       ,
>     Dwarf_Unsigned     ,
>     unsigned int       ,
>     Dwarf_Handler      ,
>     Dwarf_Ptr          ,
>     Dwarf_Debug*       ,
>     char **            ,
>     unsigned int       ,
>     unsigned char      * ,
>     const char         * ,
>     Dwarf_Unsigned     ,
>     Dwarf_Unsigned     * ,

```

```

> Dwarf_Error* );
35c62
< unsigned ,
____
> unsigned int ,
48c75
< unsigned ,
____
> unsigned int ,
70c97
< unsigned ,
____
> unsigned int ,
169a197,201
> int dwarf_find_die_given_sig8(Dwarf_Debug ,
> Dwarf_Sig8 * ,
> Dwarf_Die * ,
> Dwarf_Bool * ,
> Dwarf_Error * );
170a203,213
> int dwarf_cu_header_basics(Dwarf_Die die,
> Dwarf_Half * ,
> Dwarf_Bool * ,
> Dwarf_Bool * ,
> Dwarf_Half * ,
> Dwarf_Half * ,
> Dwarf_Half * ,
> Dwarf_Sig8 ** ,
> Dwarf_Off * ,
> Dwarf_Unsigned * ,
> Dwarf_Error * );
184a228
> Dwarf_Bool dwarf_addr_form_is_indexed(int form);
209c253,254
< int dwarf_validate_die_sibling(Dwarf_Die ,Dwarf_Off*);
____
> int dwarf_validate_die_sibling(Dwarf_Die ,
> Dwarf_Off* );
219a265,287
> int dwarf_get_gnu_index_head(Dwarf_Debug ,
> Dwarf_Bool ,
> Dwarf_Gnu_Index_Head * ,
> Dwarf_Unsigned * ,
> Dwarf_Error * );

```

```

> void dwarf_gnu_index_dealloc(Dwarf_Gnu_Index_Head );
> int dwarf_get_gnu_index_block(Dwarf_Gnu_Index_Head ,
>     Dwarf_Unsigned      ,
>     Dwarf_Unsigned      *,
>     Dwarf_Half          *,
>     Dwarf_Unsigned      *,
>     Dwarf_Unsigned      *,
>     Dwarf_Unsigned      *,
>     Dwarf_Error         * );
> int dwarf_get_gnu_index_block_entry(Dwarf_Gnu_Index_Head ,
>     Dwarf_Unsigned      ,
>     Dwarf_Unsigned      ,
>     Dwarf_Unsigned      *,
>     const char          **,
>     unsigned char       *,
>     unsigned char       *,
>     unsigned char       *,
>     Dwarf_Error         * );
221,223c289,309
<     Dwarf_Loc_Head_c *,
<     Dwarf_Unsigned      *,
<     Dwarf_Error         * );
____
>     Dwarf_Loc_Head_c *,
>     Dwarf_Unsigned      *,
>     Dwarf_Error         * );
> int
> dwarf_get_loclist_head_kind(Dwarf_Loc_Head_c ,
>     unsigned int      *,
>     Dwarf_Error       * );
> int dwarf_get_locdesc_entry_d(Dwarf_Loc_Head_c ,
>     Dwarf_Unsigned      ,
>     Dwarf_Small         * ,
>     Dwarf_Unsigned * ,
>     Dwarf_Unsigned * ,
>     Dwarf_Bool          * ,
>     Dwarf_Addr          * ,
>     Dwarf_Addr          * ,
>     Dwarf_Unsigned * ,
>     Dwarf_Locdesc_c *,
>     Dwarf_Small         * ,
>     Dwarf_Unsigned * ,
>     Dwarf_Unsigned * ,
>     Dwarf_Error         * );

```


225,234c311,332

```
< Dwarf_Unsigned    ,
< Dwarf_Small      *  ,
< Dwarf_Addr       *  ,
< Dwarf_Addr       *  ,
< Dwarf_Unsigned *   ,
< Dwarf_Locdesc_c *  ,
< Dwarf_Small      *  ,
< Dwarf_Unsigned *   ,
< Dwarf_Unsigned *   ,
< Dwarf_Error      *  );
```

```
> Dwarf_Unsigned    ,
> Dwarf_Small      *  ,
> Dwarf_Addr       *  ,
> Dwarf_Addr       *  ,
> Dwarf_Unsigned *   ,
> Dwarf_Locdesc_c *  ,
> Dwarf_Small      *  ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Error      *  );
```

> int

> dwarf_get_location_op_value_d(Dwarf_Locdesc_c ,

```
> Dwarf_Unsigned    ,
> Dwarf_Small      *  ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Unsigned *   ,
> Dwarf_Error*      );
```

236,242c334,340

```
< Dwarf_Unsigned    ,
< Dwarf_Small      *  ,
< Dwarf_Unsigned *   ,
< Dwarf_Unsigned *   ,
< Dwarf_Unsigned *   ,
< Dwarf_Unsigned *   ,
< Dwarf_Error*      );
```

```
> Dwarf_Unsigned    ,
```

```

> Dwarf_Small *,
> Dwarf_Unsigned *,
> Dwarf_Unsigned *,
> Dwarf_Unsigned *,
> Dwarf_Unsigned *,
> Dwarf_Error* );
442c540,541
< int dwarf_srclines_include_dir_count(Dwarf_Line_Context ,
____
> int dwarf_srclines_include_dir_count(
> Dwarf_Line_Context ,
445c544,545
< int dwarf_srclines_include_dir_data(Dwarf_Line_Context ,
____
> int dwarf_srclines_include_dir_data(
> Dwarf_Line_Context ,
605,610c705
< char **,
< unsigned char **,
< char **,
< unsigned * ,
< unsigned * ,
< char **,
____
> char **,
612,615c707,715
< unsigned * ,
< char ***,
< unsigned * ,
< Dwarf_Error* );
____
> char **,
> unsigned int * ,
> unsigned int * ,
> char **,
> unsigned char **,
> unsigned int * ,
> char ***,
> unsigned int * ,
> Dwarf_Error* );
618a719,722
> int dwarf_crc32 (Dwarf_Debug ,unsigned char * ,
> Dwarf_Error * );
> unsigned int dwarf_basic_crc32 (const unsigned char * ,

```

```

>     unsigned long , unsigned int );
791,798c895,902
< int dwarf_get_abbrev_entry_b(Dwarf_Abbrev abbrev,
<     Dwarf_Unsigned indx,
<     Dwarf_Bool      filter_outliers,
<     Dwarf_Unsigned * returned_attr_num,
<     Dwarf_Unsigned * returned_form,
<     Dwarf_Signed   * returned_implicit_const,
<     Dwarf_Off      * offset,
<     Dwarf_Error    * error);
-----
> int dwarf_get_abbrev_entry_b(Dwarf_Abbrev ,
>     Dwarf_Unsigned      ,
>     Dwarf_Bool          ,
>     Dwarf_Unsigned * ,
>     Dwarf_Unsigned * ,
>     Dwarf_Signed   * ,
>     Dwarf_Off      * ,
>     Dwarf_Error    * );
814,815c918,919
<     const char ** ,
<     Dwarf_Error * );
-----
>     const char ** ,
>     Dwarf_Error * );
817,818c921,922
<     const char ** ,
<     Dwarf_Error * );
-----
>     const char ** ,
>     Dwarf_Error * );
1019a1124,1126
> int dwarf_macro_context_total_length(Dwarf_Macro_Context ,
>     Dwarf_Unsigned * ,
>     Dwarf_Error    * );
1029c1136
<     unsigned      * ,
-----
>     unsigned int   * ,
1057c1164,1165
< int dwarf_get_macro_startend_file(Dwarf_Macro_Context ,
-----
> int dwarf_get_macro_startend_file(
>     Dwarf_Macro_Context ,

```

```

1069c1177
< int *);
-----
> int *);
1072c1180,1184
< void dwarf_check_lineheader(Dwarf_Die ,int *errcount_out);
-----
> int dwarf_check_lineheader_b(Dwarf_Die ,
> int *,
> Dwarf_Error *);
> void dwarf_check_lineheader(Dwarf_Die ,
> int *);
1177c1289,1290
< int dwarf_gdbindex_cuvector_inner_attributes(Dwarf_Gdbindex ,
-----
> int dwarf_gdbindex_cuvector_inner_attributes(
> Dwarf_Gdbindex ,
1233a1347
> Dwarf_Debug_Fission_Per_CU;
1244a1359,1360
> void dwarf_error_creation(Dwarf_Debug ,
> Dwarf_Error * , char *);
1247c1363,1364
< Dwarf_Handler dwarf_seterrhand(Dwarf_Debug , Dwarf_Handler);
-----
> Dwarf_Handler dwarf_seterrhand(Dwarf_Debug ,
> Dwarf_Handler );
1250a1368,1370
> void dwarf_dealloc_error(Dwarf_Debug , Dwarf_Error);
> void dwarf_dealloc_die( Dwarf_Die );
> void dwarf_dealloc_attribute(Dwarf_Attribute );
1316a1437,1443
> int dwarf_add_debug_sup(Dwarf_P_Debug ,
> Dwarf_Half ,
> Dwarf_Small ,
> char *,
> Dwarf_Unsigned ,
> Dwarf_Small *,
> Dwarf_Error *);
1479c1606,1607
< Dwarf_P_Attribute dwarf_add_AT_any_value_sleb(Dwarf_P_Die ,
-----
> Dwarf_P_Attribute dwarf_add_AT_any_value_sleb(
> Dwarf_P_Die ,

```

1488c1616,1617

```
< Dwarf_P_Attribute dwarf_add_AT_const_value_signedint(Dwarf_P_Die ,  
——
```

```
> Dwarf_P_Attribute dwarf_add_AT_const_value_signedint(  
>     Dwarf_P_Die ,
```

1495c1624,1625

```
< Dwarf_P_Attribute dwarf_add_AT_any_value_uleb(Dwarf_P_Die ,  
——
```

```
> Dwarf_P_Attribute dwarf_add_AT_any_value_uleb(  
>     Dwarf_P_Die ,
```

1528,1531c1658,1661

```
<     Dwarf_P_Die     ,  
<     Dwarf_Half      ,  
<     const Dwarf_Sig8 * ,  
<     Dwarf_Error * );  
——
```

```
>     Dwarf_P_Die     ,  
>     Dwarf_Half      ,  
>     const Dwarf_Sig8 * ,  
>     Dwarf_Error * );
```

1533,1537c1663,1667

```
<     Dwarf_P_Die     ,  
<     Dwarf_Half      ,  
<     const Dwarf_Sig8 * ,  
<     Dwarf_P_Attribute * ,  
<     Dwarf_Error * );  
——
```

```
>     Dwarf_P_Die     ,  
>     Dwarf_Half      ,  
>     const Dwarf_Sig8 * ,  
>     Dwarf_P_Attribute * ,  
>     Dwarf_Error * );
```

2072a2203,2210

```
> int dwarf_get_ranges_b(Dwarf_Debug ,
```

```
>     Dwarf_Off      ,  
>     Dwarf_Die      ,  
>     Dwarf_Off *     ,  
>     Dwarf_Ranges ** ,  
>     Dwarf_Signed *  ,  
>     Dwarf_Unsigned * ,  
>     Dwarf_Error *   );
```

2075a2214,2379

```
> int dwarf_get_debug_sup(Dwarf_Debug ,
```

```
>     Dwarf_Half      * ,
```

```

> Dwarf_Small    *,
> char           **,
> Dwarf_Unsigned *,
> Dwarf_Small    **,
> Dwarf_Error * );
> struct Dwarf_Rnglists_Entry_s;
> struct Dwarf_Rnglists_Head_s;
> int dwarf_rnglists_get_rle_head(Dwarf_Attribute      ,
> Dwarf_Half      ,
> Dwarf_Unsigned   ,
> Dwarf_Rnglists_Head *,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Error *    );
> int dwarf_get_rnglists_entry_fields_a(
> Dwarf_Rnglists_Head ,
> Dwarf_Unsigned   ,
> unsigned         * ,
> unsigned         * ,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Bool      * ,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Error *    );
> int dwarf_get_rnglists_entry_fields(Dwarf_Rnglists_Head ,
> Dwarf_Unsigned   ,
> unsigned int    * ,
> unsigned int    * ,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Error *    );
> int dwarf_dealloc_rnglists_head(Dwarf_Rnglists_Head );
> int dwarf_load_rnglists(Dwarf_Debug ,
> Dwarf_Unsigned * ,
> Dwarf_Error * );
> int dwarf_get_rnglist_offset_index_value(Dwarf_Debug ,
> Dwarf_Unsigned   ,
> Dwarf_Unsigned   ,
> Dwarf_Unsigned * ,
> Dwarf_Unsigned * ,
> Dwarf_Error * );

```

```

> int dwarf_get_rnglist_head_basics(Dwarf_Rnglists_Head ,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Half      *,
>     Dwarf_Half      *,
>     Dwarf_Half      *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Bool      *,
>     Dwarf_Unsigned *,
>     Dwarf_Bool      *,
>     Dwarf_Unsigned *,
>     Dwarf_Bool      *,
>     Dwarf_Unsigned *,
>     Dwarf_Error     *);

> int dwarf_get_rnglist_context_basics(Dwarf_Debug ,
>     Dwarf_Unsigned ,
>     Dwarf_Unsigned *,
>     Dwarf_Small  * ,
>     Dwarf_Small  * ,
>     unsigned int * ,
>     Dwarf_Small  * ,
>     Dwarf_Small  * ,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Error * );

> int dwarf_get_rnglist_raw_entry_detail(Dwarf_Debug ,
>     Dwarf_Unsigned ,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned * ,
>     Dwarf_Error * );

> int dwarf_get_rnglist_rle( Dwarf_Debug ,
>     Dwarf_Unsigned ,
>     Dwarf_Unsigned ,
>     Dwarf_Unsigned ,
>     unsigned int    * ,

```

```

>     unsigned int    *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Error * );
> struct Dwarf_Loclists_Entry_s;
> int dwarf_get_loclists_entry_fields(Dwarf_Loc_Head_c ,
>     Dwarf_Unsigned    ,
>     unsigned int    *,
>     unsigned int    *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Error      * );
> int dwarf_load_loclists(Dwarf_Debug ,
>     Dwarf_Unsigned *,
>     Dwarf_Error * );
> int dwarf_get_loclist_offset_index_value(Dwarf_Debug ,
>     Dwarf_Unsigned    ,
>     Dwarf_Unsigned    ,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Error * );
> int dwarf_get_loclist_head_basics(Dwarf_Loc_Head_c ,
>     Dwarf_Small      *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Half      *,
>     Dwarf_Half      *,
>     Dwarf_Half      *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Bool      *,
>     Dwarf_Unsigned *,
>     Dwarf_Bool      *,
>     Dwarf_Unsigned *,
>     Dwarf_Bool      *,
>     Dwarf_Unsigned *,
>     Dwarf_Unsigned *,
>     Dwarf_Error      * );

```



```

> int dwarf_get_loclist_context_basics(Dwarf_Debug ,
>   Dwarf_Unsigned ,
>   Dwarf_Unsigned *,
>   Dwarf_Small * ,
>   Dwarf_Small * ,
>   unsigned int * ,
>   Dwarf_Small * ,
>   Dwarf_Small * ,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Error * );
> int dwarf_get_loclist_raw_entry_detail(Dwarf_Debug ,
>   Dwarf_Unsigned ,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned * ,
>   Dwarf_Error * );
> int dwarf_get_loclist_lle( Dwarf_Debug ,
>   Dwarf_Unsigned ,
>   Dwarf_Unsigned ,
>   Dwarf_Unsigned ,
>   unsigned int * ,
>   unsigned int * ,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Unsigned *,
>   Dwarf_Small **,
>   Dwarf_Error * );
2083c2387,2388
< int dwarf_next_str_offsets_table( Dwarf_Str_Offsets_Table ,
____
> int dwarf_next_str_offsets_table(
>   Dwarf_Str_Offsets_Table ,
2092c2397,2398
< int dwarf_str_offsets_value_by_index(Dwarf_Str_Offsets_Table ,
____
> int dwarf_str_offsets_value_by_index(
>   Dwarf_Str_Offsets_Table ,
2096c2402,2403
< int dwarf_str_offsets_statistics(Dwarf_Str_Offsets_Table ,

```

```

-----
> int dwarf_str_offsets_statistics(
>     Dwarf_Str_Offsets_Table ,
2101,2103c2408,2410
<     unsigned ,
<     const char **,
<     unsigned * );
-----
>     unsigned int ,
>     const char ** ,
>     unsigned int * );
2106,2143c2413,2492
< unsigned dwarf_set_harmless_error_list_size(Dwarf_Debug ,
<     unsigned );
< extern int dwarf_get_ACCESS_name(unsigned int , const char ** );
< extern int dwarf_get_ADDR_name(unsigned int , const char ** );
< extern int dwarf_get_ATCF_name(unsigned int , const char ** );
< extern int dwarf_get_ATE_name(unsigned int , const char ** );
< extern int dwarf_get_AT_name(unsigned int , const char ** );
< extern int dwarf_get_CC_name(unsigned int , const char ** );
< extern int dwarf_get_CFA_name(unsigned int , const char ** );
< extern int dwarf_get_children_name(unsigned int , const char ** );
< extern int dwarf_get_CHILDREN_name(unsigned int , const char ** );
< extern int dwarf_get_DEFAULTED_name(unsigned int , const char ** );
< extern int dwarf_get_DSC_name(unsigned int , const char ** );
< extern int dwarf_get_DS_name(unsigned int , const char ** );
< extern int dwarf_get_EH_name(unsigned int , const char ** );
< extern int dwarf_get_END_name(unsigned int , const char ** );
< extern int dwarf_get_FORM_name(unsigned int , const char ** );
< extern int dwarf_get_FRAME_name(unsigned int , const char ** );
< extern int dwarf_get_ID_name(unsigned int , const char ** );
< extern int dwarf_get_IDX_name(unsigned int , const char ** );
< extern int dwarf_get_INL_name(unsigned int , const char ** );
< extern int dwarf_get_ISA_name(unsigned int , const char ** );
< extern int dwarf_get_LANG_name(unsigned int , const char ** );
< extern int dwarf_get_LLE_name(unsigned int , const char ** );
< extern int dwarf_get_LLEX_name(unsigned int , const char ** );
< extern int dwarf_get_LNCT_name(unsigned int , const char ** );
< extern int dwarf_get_LNE_name(unsigned int , const char ** );
< extern int dwarf_get_LNS_name(unsigned int , const char ** );
< extern int dwarf_get_MACINFO_name(unsigned int , const char ** );
< extern int dwarf_get_MACRO_name(unsigned int , const char ** );
< extern int dwarf_get_OP_name(unsigned int , const char ** );
< extern int dwarf_get_ORD_name(unsigned int , const char ** );

```

```

< extern int dwarf_get_RLE_name(unsigned int , const char ** );
< extern int dwarf_get_SECT_name(unsigned int , const char ** );
< extern int dwarf_get_TAG_name(unsigned int , const char ** );
< extern int dwarf_get_UT_name(unsigned int , const char ** );
< extern int dwarf_get_VIRTUALITY_name(unsigned int , const char ** );
< extern int dwarf_get_VIS_name(unsigned int , const char ** );
——
> unsigned int dwarf_set_harmless_error_list_size(Dwarf_Debug ,
>     unsigned int );
> extern int dwarf_get_ACCESS_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_ADDR_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_ATCF_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_ATE_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_AT_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_CC_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_CFA_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_children_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_CHILDREN_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_DEFAULTED_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_DSC_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_DS_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_EH_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_END_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_FORM_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_FRAME_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_GNUKIND_name(unsigned int ,
>     const char ** );
> extern int dwarf_get_GNUIVIS_name(unsigned int ,

```

```

>     const char **);
> extern int dwarf_get_ID_name(unsigned int ,
>     const char **);
> extern int dwarf_get_IDX_name(unsigned int ,
>     const char **);
> extern int dwarf_get_INL_name(unsigned int ,
>     const char **);
> extern int dwarf_get_ISA_name(unsigned int ,
>     const char **);
> extern int dwarf_get_LANG_name(unsigned int ,
>     const char **);
> extern int dwarf_get_LLE_name(unsigned int ,
>     const char **);
> extern int dwarf_get_LLEX_name(unsigned int ,
>     const char **);
> extern int dwarf_get_LNCT_name(unsigned int ,
>     const char **);
> extern int dwarf_get_LNE_name(unsigned int ,
>     const char **);
> extern int dwarf_get_LNS_name(unsigned int ,
>     const char **);
> extern int dwarf_get_MACINFO_name(unsigned int ,
>     const char **);
> extern int dwarf_get_MACRO_name(unsigned int ,
>     const char **);
> extern int dwarf_get_OP_name(unsigned int ,
>     const char **);
> extern int dwarf_get_ORD_name(unsigned int ,
>     const char **);
> extern int dwarf_get_RLE_name(unsigned int ,
>     const char **);
> extern int dwarf_get_SECT_name(unsigned int ,
>     const char **);
> extern int dwarf_get_TAG_name(unsigned int ,
>     const char **);
> extern int dwarf_get_UT_name(unsigned int ,
>     const char **);
> extern int dwarf_get_VIRTUALITY_name(unsigned int ,
>     const char **);
> extern int dwarf_get_VIS_name(unsigned int ,
>     const char **);
> extern int dwarf_get_FORM_CLASS_name(enum Dwarf_Form_Class ,
>     const char **);

```

2197c2546,2547

```
< void (*dwarf_get_endian_copy_function(Dwarf_Debug))(void *, const void *, unsigned long );
```

```
> void (*dwarf_get_endian_copy_function(Dwarf_Debug ))
```

```
>     (void *, const void *, unsigned long );
```

```
2205a2556,2564
```

```
> int dwarf_decode_leb128(char *,
```

```
>     Dwarf_Unsigned *,
```

```
>     Dwarf_Unsigned *,
```

```
>     char                * );
```

```
> int dwarf_decode_signed_leb128(char *,
```

```
>     Dwarf_Unsigned *,
```

```
>     Dwarf_Signed    *,
```

```
>     char                * );
```

```
> } Dwarf_Cmdline_Options;
```

```
2216,2224c2575,2599
```

```
< int dwarf_object_detector_path(const char  *path,
```

```
<     char *outpath,
```

```
<     unsigned long,
```

```
<     unsigned *ftype,
```

```
<     unsigned *endian,
```

```
<     unsigned *offsetsize,
```

```
<     Dwarf_Unsigned  *filesize,
```

```
<     int * errcode);
```

```
< int dwarf_object_detector_fd(int fd,
```

```
> int dwarf_set_de_alloc_flag(int v);
```

```
> int dwarf_object_detector_path(const char  *,
```

```
>     char                *,
```

```
>     unsigned long      ,
```

```
>     unsigned int       *,
```

```
>     unsigned int       *,
```

```
>     unsigned int       *,
```

```
>     Dwarf_Unsigned *,
```

```
>     int *   );
```

```
> int dwarf_object_detector_path_b(const char  *,
```

```
>     char                *,
```

```
>     unsigned long      ,
```

```
>     char **            ,
```

```
>     unsigned           ,
```

```
>     unsigned int *,
```

```
>     unsigned int *,
```

```
>     unsigned int *,
```

```
>     Dwarf_Unsigned *,
```

```
>     unsigned char *,
```

```

>     int * );
> int dwarf_object_detector_path_dSYM(
>     const char    *path,
>     char *outpath, unsigned long outpath_len,
>     char ** gl_pathnames,
>     unsigned gl_pathcount,
2229c2604,2614
<     int * errcode);
-----
>     unsigned char *pathsource,
>     int *errcode);
> int dwarf_get_path_source_type(Dwarf_Debug ,
>     unsigned char * ,
>     Dwarf_Error    * );
> int dwarf_object_detector_fd(int ,
>     unsigned int * ,
>     unsigned int * ,
>     unsigned int * ,
>     Dwarf_Unsigned * ,
>     int *    );

```

nodejs-devel

改变 config.gypi

3c3,4

```

<                                     'defines': ['NODE_OPENSSL_CERT_STORE'],
-----
>                                     'defines': [ 'NODE_OPENSSL_CERT_STORE',
>                                                 'ICU_NO_USER_DATA_OVERRIDE'],

```

virglrenderer-devel

改变 virglrenderer.h

12a13,14

```

> VIRGL_EXPORT void virgl_renderer_resource_set_priv(uint32_t res_handle, void *priv);
> VIRGL_EXPORT void *virgl_renderer_resource_get_priv(uint32_t res_handle);
43a46
> VIRGL_EXPORT virgl_debug_callback_type virgl_set_debug_callback(virgl_debug_callback_type cb);
48a52
> VIRGL_EXPORT int virgl_renderer_execute(void *execute_args, uint32_t execute_size);

```

libkysecwhlist-devel

增加头文件: kysec_pblk_list.h

增加头文件: kysec_ppro_list.h

改变 kysec_whlist.h

23a24

```
> int kysec_whlist_update_rmmodpro_sm(int status, char *path);
```

29a31

```
> int kysec_whlist_load_kdk(int kysec_type, const char *key, struct object_list *objects , int data_source_type , char *data_source_name);
```

30a33

```
> int kysec_whlist_load_sm(int kysec_type, const char *key, struct object_list *objects);
```

41a45,47

```
> int kysec_kmod_whlist_load_sm(const char *key, struct kmod_list *objects);
```

```
> int kysec_kmod_whlist_load_kdk(const char *key, const int data_source_type, const char *data_source_name, struct kmod_list *objects);
```

```
> int kysec_kmod_clear_policy_kdk(int data_source_type, char *data_source_name);
```

48a55,67

```
> int kysec_pkglist_create();
```

```
> int kysec_pkglist_get_switch_status();
```

```
> int kysec_pkglist_set_switch_status(int value);
```

```
> int kysec_pkglist_add(const char* debname);
```

```
> int kysec_pkglist_add_kdk(const char* debname, int data_source_type, const char* data_source_name);
```

```
> int kysec_pkglist_del(const char* debname);
```

```
> int kysec_pkglist_del_kdk(const char* debname, int data_source_type, const char* data_source_name);
```

```
> char** kysec_pkglist_read_all_data(int* count);
```

```
> int kysec_pkglist_clear();
```

```
> int kysec_pkglist_clear_policy_kdk(int data_source_type, char *data_source_name);
```

```
> int kysec_clear_all_policy_kdk(int data_source_type, char *data_source_name);
```

```
> int kysec_bgset_set(const char* picture_path);
```

```
> int kysec_bgset_unset();
```

php-devel

改变 credits_ext.h

19c19

```
< CREDIT_LINE("Iconv", "Rui Hirokawa, Stig Bakken, Moriyoshi Koizumi ");
```

——

```
> CREDIT_LINE("Iconv", "Rui Hirokawa, Stig Bakken, Moriyoshi Koizumi");
```

改变 cyr_convert.h

1a2

>

改变 file.h

54a55

>

改变 info.h

27a28

```

>
改变 mbfl_memory_device.h
15a16
>
改变 mysqlnd_ext_plugin.h
1,2c1,2
< extern struct st_mysqlnd_plugin__plugin_area_getters mysqlnd_plugin_area_getters;
< extern struct st_mysqlnd_plugin_methods_xetters mysqlnd_plugin_methods_xetters;
-----
> PHPAPI extern struct st_mysqlnd_plugin__plugin_area_getters mysqlnd_plugin_area_getters;
> PHPAPI extern struct st_mysqlnd_plugin_methods_xetters mysqlnd_plugin_methods_xetters;
改变 mysqlnd.h
31c31
< void mysqlnd_local_infile_default(MYSQLND_CONN_DATA * conn);
-----
> PHPAPI void mysqlnd_local_infile_default(MYSQLND_CONN_DATA * conn);
改变 mysqlnd_structs.h
208d207
< struct st_mysqlnd_packet_cached_sha2_result;
改变 php_date.h
70a71
> PHP_METHOD(DatePeriod, getRecurrences);
99a101
> PHPAPI time_t php_time();
改变 php_json_parser.h
18a19
>
改变 php_json_scanner.h
2a3
>
改变 php_libxml.h
29a30,39
> static zend_always_inline void php_libxml_sanitize_parse_ctxt_options(xmlParserCtxtPtr ctxt)
> {
>     ctxt->loadsubset = 0;
>     ctxt->validate = 0;
>     ctxt->pedantic = 0;
>     ctxt->replaceEntities = 0;
>     ctxt->linenumbers = 0;
>     ctxt->keepBlanks = 1;
>     ctxt->options = 0;
> }
改变 php_mt_rand.h
5a6

```



```

>
改变 php_pcre.h
5a6
> PHPAPI pcre_cache_entry* pcre_get_compiled_regex_cache_ex(zend_string *regex, int locale_aware);
改变 php_string.h
97,98c97,98
< PHPAPI size_t php_strspn(char *s1, char *s2, char *s1_end, char *s2_end);
< PHPAPI size_t php_strcspn(char *s1, char *s2, char *s1_end, char *s2_end);
---
> PHPAPI size_t php_strspn(char *s1, char *s2, char *s1_end, char *s2_end);
> PHPAPI size_t php_strcspn(char *s1, char *s2, char *s1_end, char *s2_end);
改变 zend_hash.h
106c106
<     return (zv && Z_TYPE_P(zv) == IS_INDIRECT) ?
---
>     return (zv && Z_TYPE_P(zv) == IS_INDIRECT) ?
120c120
<     return (zv && Z_TYPE_P(zv) == IS_INDIRECT) ?
---
>     return (zv && Z_TYPE_P(zv) == IS_INDIRECT) ?
526c526
<         zend_string_hash_val(key);
---
>         zend_string_hash_val(key);
546c546
<         zend_string_hash_val(key);
---
>         zend_string_hash_val(key);
566c566
<         zend_string_hash_val(key);
---
>         zend_string_hash_val(key);
改变 zend_signal.h
11c11
< void zend_signal_startup(void);
---
> ZEND_API void zend_signal_startup(void);
改变 zend_vm_def.h
666a667
>         const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
668c669
<         if (OP1_TYPE == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
---
>         if (OP1_TYPE == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {

```

```

2058a2060
> zend_vm_stack_free_extra_args_ex(call_info, execute_data);
2070d2071
< zend_vm_stack_free_extra_args_ex(call_info, execute_data);
2677a2679
> FREE_UNFETCHED_OP2();
2787c2789
< if ((opline->op1.num & ZEND_FETCH_CLASS_MASK) == ZEND_FETCH_CLASS_PARENT ||
----
> if ((opline->op1.num & ZEND_FETCH_CLASS_MASK) == ZEND_FETCH_CLASS_PARENT ||
4074a4077
> const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
4076c4079
< if (OP1_TYPE == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
----
> if (OP1_TYPE == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
5478c5481
< }
----
> }
7338c7341
< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_INC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_PRE_INC_LONG_NO_OVERFLOW, TMPVARCV, ANY, SPEC(RETV))
----
> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_INC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_PRE_INC_LONG_NO_OVERFLOW, CV, ANY, SPEC(RETV))
7349c7352
< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_INC, (op1_info == MAY_BE_LONG),
ZEND_PRE_INC_LONG, TMPVARCV, ANY, SPEC(RETV))
----
> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_INC, (op1_info == MAY_BE_LONG),
ZEND_PRE_INC_LONG, CV, ANY, SPEC(RETV))
7360c7363
< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_INC, (op1_info ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_PRE_INC_LONG_OR_DOUBLE, TMPVARCV, ANY,
SPEC(RETV))
----
> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_INC, (op1_info ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_PRE_INC_LONG_OR_DOUBLE, CV, ANY, SPEC(RETV))
7375c7378
< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_DEC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_PRE_DEC_LONG_NO_OVERFLOW, TMPVARCV, ANY, SPEC(RETV))
----
> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_DEC, (res_info == MAY_BE_LONG && op1_info ==

```

MAY_BE_LONG), ZEND_PRE_DEC_LONG_NO_OVERFLOW, CV, ANY, SPEC(RETVAL))

7386c7389

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_DEC, (op1_info == MAY_BE_LONG),
ZEND_PRE_DEC_LONG, TMPVARCV, ANY, SPEC(RETVAL))

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_DEC, (op1_info == MAY_BE_LONG),
ZEND_PRE_DEC_LONG, CV, ANY, SPEC(RETVAL))

7397c7400

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_DEC, (op1_info ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_PRE_DEC_LONG_OR_DOUBLE, TMPVARCV, ANY,
SPEC(RETVAL))

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_PRE_DEC, (op1_info ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_PRE_DEC_LONG_OR_DOUBLE, CV, ANY, SPEC(RETVAL))

7412c7415

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_INC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_POST_INC_LONG_NO_OVERFLOW, TMPVARCV, ANY)

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_INC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_POST_INC_LONG_NO_OVERFLOW, CV, ANY)

7421c7424

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_INC, (op1_info == MAY_BE_LONG),
ZEND_POST_INC_LONG, TMPVARCV, ANY)

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_INC, (op1_info == MAY_BE_LONG),
ZEND_POST_INC_LONG, CV, ANY)

7430c7433

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_INC, (op1_info ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_POST_INC_LONG_OR_DOUBLE, TMPVARCV, ANY)

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_INC, (op1_info ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_POST_INC_LONG_OR_DOUBLE, CV, ANY)

7443c7446

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_DEC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_POST_DEC_LONG_NO_OVERFLOW, TMPVARCV, ANY)

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_DEC, (res_info == MAY_BE_LONG && op1_info ==
MAY_BE_LONG), ZEND_POST_DEC_LONG_NO_OVERFLOW, CV, ANY)

7452c7455

< ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_DEC, (op1_info == MAY_BE_LONG),
ZEND_POST_DEC_LONG, TMPVARCV, ANY)

> ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_DEC, (op1_info == MAY_BE_LONG),
ZEND_POST_DEC_LONG, CV, ANY)

```

7461c7464
<      ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_DEC,      (op1_info      ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_POST_DEC_LONG_OR_DOUBLE, TMPVARCV, ANY)
-----
>      ZEND_VM_HOT_TYPE_SPEC_HANDLER(ZEND_POST_DEC,      (op1_info      ==
(MAY_BE_LONG|MAY_BE_DOUBLE)), ZEND_POST_DEC_LONG_OR_DOUBLE, CV, ANY)
改变 zend_vm_execute.h
328a329
>      zend_vm_stack_free_extra_args_ex(call_info, execute_data);
340d340
<      zend_vm_stack_free_extra_args_ex(call_info, execute_data);
2112a2113
>      const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
2114c2115
<      if (IS_CONST == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
-----
>      if (IS_CONST == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
2550a2552
>      const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
2552c2554
<      if (IS_CONST == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
-----
>      if (IS_CONST == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
9368a9371
>      zval_ptr_dtor_nogc(EX_VAR(opline->op2.var));
21947a21951
>      zval_ptr_dtor_nogc(EX_VAR(opline->op2.var));
27097a27102
>      zval_ptr_dtor_nogc(EX_VAR(opline->op2.var));
27317a27323
>      const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
27319c27325
<      if (IS_CV == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
-----
>      if (IS_CV == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
27991a27998
>      const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
27993c28000
<      if (IS_CV == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
-----
>      if (IS_CV == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
28661a28669,28878
>      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HAND

```

```

LER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     Z_LVAL_P(var_ptr)++;
>     if (UNEXPECTED(0)) {
>         ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>
>     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLE
R_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     Z_LVAL_P(var_ptr)++;
>     if (UNEXPECTED(1)) {
>         ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>
>     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_INC_LONG_SPEC_CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     fast_long_increment_function(var_ptr);
>     if (UNEXPECTED(0)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>
>     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_INC_LONG_SPEC_CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     fast_long_increment_function(var_ptr);
>     if (UNEXPECTED(1)) {

```

```

>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLE
R_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
>         fast_long_increment_function(var_ptr);
>     } else {
>         Z_DVAL_P(var_ptr)++;
>     }
>     if (UNEXPECTED(0)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_A
RGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
>         fast_long_increment_function(var_ptr);
>     } else {
>         Z_DVAL_P(var_ptr)++;
>     }
>     if (UNEXPECTED(1)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HAN
DLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;

```

```

>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     Z_LVAL_P(var_ptr)--;
>     if (UNEXPECTED(0)) {
>         ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>     static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     Z_LVAL_P(var_ptr)--;
>     if (UNEXPECTED(1)) {
>         ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>     static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_DEC_LONG_SPEC_CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     fast_long_decrement_function(var_ptr);
>     if (UNEXPECTED(0)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
>     static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_DEC_LONG_SPEC_CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     fast_long_decrement_function(var_ptr);
>     if (UNEXPECTED(1)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }

```

```

> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_
R_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
>         fast_long_decrement_function(var_ptr);
>     } else {
>         Z_DVAL_P(var_ptr)--;
>     }
>     if (UNEXPECTED(0)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }

> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_
ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
>         fast_long_decrement_function(var_ptr);
>     } else {
>         Z_DVAL_P(var_ptr)--;
>     }
>     if (UNEXPECTED(1)) {
>         ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     }
>     ZEND_VM_NEXT_OPCODE();
> }

> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     Z_LVAL_P(var_ptr)++;
>     ZEND_VM_NEXT_OPCODE();
> }

```



```

> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_POST_INC_LONG_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     fast_long_increment_function(var_ptr);
>     ZEND_VM_NEXT_OPCODE();
> }
> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
>         fast_long_increment_function(var_ptr);
>     } else {
>         Z_DVAL_P(var_ptr)++;
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     Z_LVAL_P(var_ptr)--;
>     ZEND_VM_NEXT_OPCODE();
> }
> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_POST_DEC_LONG_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
>     fast_long_decrement_function(var_ptr);
>     ZEND_VM_NEXT_OPCODE();
> }

```

```

> static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
> {
>     USE_OPLINE
>     zval *var_ptr;
>     var_ptr = _get_zval_ptr_cv_undef_BP_VAR_RW(opline->op1.var EXECUTE_DATA_CC);
>     ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
>     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
>         fast_long_decrement_function(var_ptr);
>     } else {
>         Z_DVAL_P(var_ptr)--;
>     }
>     ZEND_VM_NEXT_OPCODE();
> }
40073a40291
>     const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
40075c40293
<     if ((IS_TMP_VAR|IS_VAR) == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
----
>     if ((IS_TMP_VAR|IS_VAR) == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
40303a40522
>     const uint32_t orig_val_type = Z_TYPE_INFO_P(val);
40305c40524
<     if ((IS_TMP_VAR|IS_VAR) == IS_CV && UNEXPECTED(Z_TYPE_INFO_P(val) == IS_UNDEF)) {
----
>     if ((IS_TMP_VAR|IS_VAR) == IS_CV && UNEXPECTED(orig_val_type == IS_UNDEF)) {
44981,45190d45199
< static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_
E_HANDLER_ARGS)
< {
<     USE_OPLINE
<     zval *var_ptr;
<     var_ptr = EX_VAR(opline->op1.var);
<     Z_LVAL_P(var_ptr)++;
<     if (UNEXPECTED(0)) {
<         ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<     }
<     ZEND_VM_NEXT_OPCODE();
< }
< static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_
HANDLER_ARGS)
< {

```

```

< USE_OPLINE
< zval *var_ptr;
< var_ptr = EX_VAR(opline->op1.var);
< Z_LVAL_P(var_ptr)++;
< if (UNEXPECTED(1)) {
<     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
< }
< ZEND_VM_NEXT_OPCODE();
<}

< static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_INC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_AR
GS)
<{
< USE_OPLINE
< zval *var_ptr;
< var_ptr = EX_VAR(opline->op1.var);
< fast_long_increment_function(var_ptr);
< if (UNEXPECTED(0)) {
<     ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
< }
< ZEND_VM_NEXT_OPCODE();
<}

< static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_INC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
<{
< USE_OPLINE
< zval *var_ptr;
< var_ptr = EX_VAR(opline->op1.var);
< fast_long_increment_function(var_ptr);
< if (UNEXPECTED(1)) {
<     ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
< }
< ZEND_VM_NEXT_OPCODE();
<}

< static ZEND_VM_HOT ZEND_OPCODE_HANDLER_RET ZEND_FASTCALL
ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_
HANDLER_ARGS)
<{
< USE_OPLINE
< zval *var_ptr;
< var_ptr = EX_VAR(opline->op1.var);
< if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
<     fast_long_increment_function(var_ptr);
< } else {

```

```

<      Z_DVAL_P(var_ptr)++;
<  }
<  if (UNEXPECTED(0)) {
<      ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<  }
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
<      fast_long_increment_function(var_ptr);
<  } else {
<      Z_DVAL_P(var_ptr)++;
<  }
<  if (UNEXPECTED(1)) {
<      ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<  }
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  Z_LVAL_P(var_ptr)--;
<  if (UNEXPECTED(0)) {
<      ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<  }
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);

```

```

<  Z_LVAL_P(var_ptr)--;
<  if (UNEXPECTED(1)) {
<      ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<  }
<  ZEND_VM_NEXT_OPCODE();
<}

<      static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  fast_long_decrement_function(var_ptr);
<  if (UNEXPECTED(0)) {
<      ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<  }
<  ZEND_VM_NEXT_OPCODE();
<}

<      static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS
)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  fast_long_decrement_function(var_ptr);
<  if (UNEXPECTED(1)) {
<      ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<  }
<  ZEND_VM_NEXT_OPCODE();
<}

<      static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_
HANDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
<      fast_long_decrement_function(var_ptr);
<  } else {
<      Z_DVAL_P(var_ptr)--;
<  }

```

```

<  if (UNEXPECTED(0)) {
<      ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<  }
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HA
NDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
<      fast_long_decrement_function(var_ptr);
<  } else {
<      Z_DVAL_P(var_ptr)--;
<  }
<  if (UNEXPECTED(1)) {
<      ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<  }
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARG
S)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<  Z_LVAL_P(var_ptr)++;
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL
ZEND_POST_INC_LONG_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<  USE_OPLINE
<  zval *var_ptr;
<  var_ptr = EX_VAR(opline->op1.var);
<  ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<  fast_long_increment_function(var_ptr);
<  ZEND_VM_NEXT_OPCODE();
<}
<      static      ZEND_VM_HOT      ZEND_OPCODE_HANDLER_RET      ZEND_FASTCALL

```

```

ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<     USE_OPLINE
<     zval *var_ptr;
<     var_ptr = EX_VAR(opline->op1.var);
<     ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
<         fast_long_increment_function(var_ptr);
<     } else {
<         Z_DVAL_P(var_ptr)++;
<     }
<     ZEND_VM_NEXT_OPCODE();
< }
<
<     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_AR
GS)
< {
<     USE_OPLINE
<     zval *var_ptr;
<     var_ptr = EX_VAR(opline->op1.var);
<     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<     Z_LVAL_P(var_ptr)--;
<     ZEND_VM_NEXT_OPCODE();
< }
<
<     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_POST_DEC_LONG_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<     USE_OPLINE
<     zval *var_ptr;
<     var_ptr = EX_VAR(opline->op1.var);
<     ZVAL_LONG(EX_VAR(opline->result.var), Z_LVAL_P(var_ptr));
<     fast_long_decrement_function(var_ptr);
<     ZEND_VM_NEXT_OPCODE();
< }
<
<     static          ZEND_VM_HOT          ZEND_OPCODE_HANDLER_RET          ZEND_FASTCALL
ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARGS)
< {
<     USE_OPLINE
<     zval *var_ptr;
<     var_ptr = EX_VAR(opline->op1.var);
<     ZVAL_COPY_VALUE(EX_VAR(opline->result.var), var_ptr);
<     if (EXPECTED(Z_TYPE_P(var_ptr) == IS_LONG)) {
<         fast_long_decrement_function(var_ptr);
<     } else {

```

```

<      Z_DVAL_P(var_ptr)---;
<  }
<  ZEND_VM_NEXT_OPCODE();
<}
50497,50500d50505
<
      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCHV_RETVAL_UNUSED_LABEL,
<
      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCHV_RETVAL_USED_LABEL,
<
      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCHV_RETVAL_UNUSED_LABEL,
<
      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCHV_RETVAL_USED_LABEL,
50503,50504d50507
<
      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCHV_RETVAL_UNUSED_LABEL,
<
      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCHV_RETVAL_USED_LABEL,
50507,50510d50509
<      (void*)&&ZEND_PRE_INC_LONG_SPEC_TMPVARCHV_RETVAL_UNUSED_LABEL,
<      (void*)&&ZEND_PRE_INC_LONG_SPEC_TMPVARCHV_RETVAL_USED_LABEL,
<      (void*)&&ZEND_PRE_INC_LONG_SPEC_TMPVARCHV_RETVAL_UNUSED_LABEL,
<      (void*)&&ZEND_PRE_INC_LONG_SPEC_TMPVARCHV_RETVAL_USED_LABEL,
50513,50514c50512,50530
<      (void*)&&ZEND_PRE_INC_LONG_SPEC_TMPVARCHV_RETVAL_UNUSED_LABEL,
<      (void*)&&ZEND_PRE_INC_LONG_SPEC_TMPVARCHV_RETVAL_USED_LABEL,
---
>      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED_LABEL,
>      (void*)&&ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_PRE_INC_LONG_SPEC_CV_RETVAL_UNUSED_LABEL,
>      (void*)&&ZEND_PRE_INC_LONG_SPEC_CV_RETVAL_USED_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,
>      (void*)&&ZEND_NULL_LABEL,

```



```

>         (void*)&&ZEND_NULL_LABEL,
>         (void*)&&ZEND_NULL_LABEL,
50515a50532,50533
>         (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_UNUSED_LABEL,
>         (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_USED_LABEL,
50517,50520d50534
<
    (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_LABEL,
<
    (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50523,50524d50536
<
    (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50527,50530d50538
<
    (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50533,50534d50540
<
    (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50535a50542,50543
>         (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED_LABEL,
>         (void*)&&ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED_LABEL,
50537,50540d50544
<
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50543,50544d50546
<
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_LABEL,

```

```

50547,50550d50548
<
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50553,50554d50550
<
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_LABEL,
<
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_LABEL,
50556,50557c50552,50553
<
    (void*)&&ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_LABEL,
<
    (void*)&&ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_LABEL,
-----
>
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_CV_RETVAL_UNUSED_LABEL,
>
    (void*)&&ZEND_PRE_DEC_LONG_SPEC_CV_RETVAL_USED_LABEL,
50559d50554
<
    (void*)&&ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_LABEL,
50561,50562d50555
<
    (void*)&&ZEND_POST_INC_LONG_SPEC_TMPVARCV_LABEL,
<
    (void*)&&ZEND_POST_INC_LONG_SPEC_TMPVARCV_LABEL,
50564d50556
<
    (void*)&&ZEND_POST_INC_LONG_SPEC_TMPVARCV_LABEL,
50566,50567d50557
<
    (void*)&&ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_LABEL,
<
    (void*)&&ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_LABEL,
50569d50558
<
    (void*)&&ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_LABEL,
50571,50572d50559
<
    (void*)&&ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_LABEL,
<
    (void*)&&ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_LABEL,
50574d50560
<
    (void*)&&ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_LABEL,
50576,50577c50562,50563
<
    (void*)&&ZEND_POST_DEC_LONG_SPEC_TMPVARCV_LABEL,
<
    (void*)&&ZEND_POST_DEC_LONG_SPEC_TMPVARCV_LABEL,
-----
>
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_UNUSED_LABEL,
>
    (void*)&&ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_USED_LABEL,
50579d50564

```

```

<          (void*)&&ZEND_POST_DEC_LONG_SPEC_TMPVARCV_LABEL,
50581,50582d50565
<          (void*)&&ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_LABEL,
<          (void*)&&ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_LABEL,
50584c50567,50593
<          (void*)&&ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_LABEL,
____
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_CV_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_POST_INC_LONG_SPEC_CV_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_CV_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_CV_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_POST_DEC_LONG_SPEC_CV_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_NULL_LABEL,
>          (void*)&&ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_CV_LABEL,
52983a52993,53046
>          HYBRID_CASE(ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED):
>
>          ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_
HANDLER_ARGS_PASSTHRU);
>          HYBRID_BREAK();
>          HYBRID_CASE(ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED):
>
>          ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED_HANDLER(ZEND_OPCODE_HA
NDLER_ARGS_PASSTHRU);

```

```

>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_INC_LONG_SPEC_CV_RETVAl_UNUSED):
>
>         ZEND_PRE_INC_LONG_SPEC_CV_RETVAl_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_
PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_INC_LONG_SPEC_CV_RETVAl_USED):
>
>         ZEND_PRE_INC_LONG_SPEC_CV_RETVAl_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PA
SSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAl_UNUSED):
>
>         ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAl_UNUSED_HANDLER(ZEND_OPCODE_HAN
DLER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAl_USED):
>
>         ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_CV_RETVAl_USED_HANDLER(ZEND_OPCODE_HANDL
ER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAl_UNUSED):
>
>         ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAl_UNUSED_HANDLER(ZEND_OPCODE_
HANDLER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAl_USED):
>
>         ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAl_USED_HANDLER(ZEND_OPCODE_HA
NDLER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_DEC_LONG_SPEC_CV_RETVAl_UNUSED):
>
>         ZEND_PRE_DEC_LONG_SPEC_CV_RETVAl_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS
_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_DEC_LONG_SPEC_CV_RETVAl_USED):
>
>         ZEND_PRE_DEC_LONG_SPEC_CV_RETVAl_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PA
SSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAl_UNUSED):
>
>         ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAl_UNUSED_HANDLER(ZEND_OPCODE_HA

```

```

NDLER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAIL_USED):
>
>         ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAIL_USED_HANDLER(ZEND_OPCODE_HAND
LER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_CV):
>
>         ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_P
ASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_POST_INC_LONG_SPEC_CV):
>
>         ZEND_POST_INC_LONG_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_CV):
>
>         ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PAS
STHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_CV):
>
>         ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_
PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_POST_DEC_LONG_SPEC_CV):
>
>         ZEND_POST_DEC_LONG_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
>         HYBRID_BREAK();
>         HYBRID_CASE(ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_CV):
>
>         ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_CV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PAS
STHRU);
>         HYBRID_BREAK();
54481,54534d54543
<
>         HYBRID_CASE(ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAIL_UNUSED):
<
>         ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAIL_UNUSED_HANDLER(ZEND_O
PCODE_HANDLER_ARGS_PASSTHRU);
<
>         HYBRID_BREAK();
<
>         HYBRID_CASE(ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAIL_USED):

```

```

<
    ZEND_PRE_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_INC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED):
<
    ZEND_PRE_INC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_INC_LONG_SPEC_TMPVARCV_RETVAL_USED):
<
    ZEND_PRE_INC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED):
<
    ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED):
<
    ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED):
<
    ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED):
<
    ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED):
<
    ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHRU);
<
    HYBRID_BREAK();
<
    HYBRID_CASE(ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED):
<

```

```

        ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE_HANDLER_
ARGS_PASSTHRU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED):
    <
        ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER(ZEND_OPC
ODE_HANDLER_ARGS_PASSTHRU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED):
    <
        ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER(ZEND_OPCODE
E_HANDLER_ARGS_PASSTHRU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV):
    <
        ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER
_ARGS_PASSTHRU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_POST_INC_LONG_SPEC_TMPVARCV):
    <
        ZEND_POST_INC_LONG_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTHR
U);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV):
    <
        ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_A
RGS_PASSTHRU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV):
    <
        ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLE
R_ARGS_PASSTHRU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_POST_DEC_LONG_SPEC_TMPVARCV):
    <
        ZEND_POST_DEC_LONG_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_ARGS_PASSTH
RU);
    <
        HYBRID_BREAK();
    <
        HYBRID_CASE(ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV):
    <
        ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER(ZEND_OPCODE_HANDLER_A
RGS_PASSTHRU);

```

[illegible]


```

> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_UNUSED_HANDLER,
> ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_CV_RETVAL_USED_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_PRE_DEC_LONG_SPEC_CV_RETVAL_UNUSED_HANDLER,
> ZEND_PRE_DEC_LONG_SPEC_CV_RETVAL_USED_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
> ZEND_NULL_HANDLER,
59480,59483d59522
< ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
< ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
59486,59487c59525,59526
< ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
——
> ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_UNUSED_HANDLER,
> ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_CV_RETVAL_USED_HANDLER,
59490,59493d59528
< ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
< ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
59496,59497c59531
< ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
——
> ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_CV_HANDLER,
59500,59503d59533
< ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
< ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER,

```

```

<      ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
<      ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
59506,59507c59536
<      ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
<      ZEND_PRE_DEC_LONG_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
____
>      ZEND_POST_INC_LONG_SPEC_CV_HANDLER,
59510,59513d59538
<      ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
<      ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
<      ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
<      ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
59516,59517c59541
<      ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_UNUSED_HANDLER,
<      ZEND_PRE_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_RETVAL_USED_HANDLER,
____
>      ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_CV_HANDLER,
59519,59520d59542
<      ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER,
<      ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER,
59522d59543
<      ZEND_POST_INC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER,
59524,59525d59544
<      ZEND_POST_INC_LONG_SPEC_TMPVARCV_HANDLER,
<      ZEND_POST_INC_LONG_SPEC_TMPVARCV_HANDLER,
59527c59546
<      ZEND_POST_INC_LONG_SPEC_TMPVARCV_HANDLER,
____
>      ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_CV_HANDLER,
59529,59530d59547
<      ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER,
<      ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER,
59532d59548
<      ZEND_POST_INC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER,
59534,59535d59549
<      ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER,
<      ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER,
59537c59551
<      ZEND_POST_DEC_LONG_NO_OVERFLOW_SPEC_TMPVARCV_HANDLER,
____
>      ZEND_POST_DEC_LONG_SPEC_CV_HANDLER,
59539,59540d59552
<      ZEND_POST_DEC_LONG_SPEC_TMPVARCV_HANDLER,
<      ZEND_POST_DEC_LONG_SPEC_TMPVARCV_HANDLER,

```

59542d59553

```
<      ZEND_POST_DEC_LONG_SPEC_TMPVARCV_HANDLER,  
59544,59545d59554  
<      ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER,  
<      ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER,  
59547c59556  
<      ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_TMPVARCV_HANDLER,  
_____  
>      ZEND_POST_DEC_LONG_OR_DOUBLE_SPEC_CV_HANDLER,
```

hdf5-devel

变动很大，略

libnghttp2-devel

改变 nghttp2.h

127a128,130

```
> NGHTTP2_EXTERN void  
> nghttp2_option_set_stream_reset_rate_limit(nghttp2_option *option,  
>      uint64_t burst, uint64_t rate);
```

libwebsockets-devel

改变 lws-adopt.h

22c22

```
<      const lws_retry_bo_t *retry_policy);  
_____  
>      const lws_retry_bo_t *retry_policy, const char *fi_wsi_name);
```

改变 lws-async-dns.h

0a1

> struct addrinfo;

增加头文件：lws-bb-i2c.h

增加头文件：lws-bb-spi.h

增加头文件：lws-button.h

增加头文件：lws-cache-ttl.h

改变 lws-client.h

20a21,22

```
> LWS_VISIBLE LWS_EXTERN int  
> lws_tls_session_is_reused(struct lws *wsi);
```

增加头文件：lws-common.h

改变 lws-context-vhost.h

3a4

```

> struct lws_metric_policy;
49a51,54
> LWS_VISIBLE LWS_EXTERN const char *
> lws_vh_tag(struct lws_vhost *vh);
> LWS_VISIBLE LWS_EXTERN int
> lws_context_is_being_destroyed(struct lws_context *context);
增加头文件: lws-cose.h
减少头文件: lws-detailed-latency.h
改变 lws-diskcache.h
7c7
< lws_diskcache_prepare(const char *cache_base_dir, int mode, int uid);
——
> lws_diskcache_prepare(const char *cache_base_dir, int mode, uid_t uid);
增加头文件: lws-display.h
改变 lws-dll2.h
1,18d0
< LWS_VISIBLE LWS_EXTERN void
< lws_dll_add_head(struct lws_dll *d, struct lws_dll *phead);
< LWS_VISIBLE LWS_EXTERN void
< lws_dll_add_tail(struct lws_dll *d, struct lws_dll *phead);
< LWS_VISIBLE LWS_EXTERN void
< lws_dll_insert(struct lws_dll *d, struct lws_dll *target,
<         struct lws_dll *phead, int before);
< static LWS_INLINE struct lws_dll *
< lws_dll_get_head(struct lws_dll *phead) { return phead->next; }
< static LWS_INLINE struct lws_dll *
< lws_dll_get_tail(struct lws_dll *phead) { return phead->prev; }
< LWS_VISIBLE LWS_EXTERN void
< lws_dll_remove(struct lws_dll *d) LWS_WARN_DEPRECATED;
< LWS_VISIBLE LWS_EXTERN void
< lws_dll_remove_track_tail(struct lws_dll *d, struct lws_dll *phead);
< LWS_VISIBLE LWS_EXTERN int
< lws_dll_foreach_safe(struct lws_dll *phead, void *user,
<         int (*cb)(struct lws_dll *d, void *user));
21,22c3,4
< static LWS_INLINE int
< lws_dll2_is_detached(const struct lws_dll2 *d) { return !d->owner; }
——
> LWS_VISIBLE LWS_EXTERN int
> lws_dll2_is_detached(const struct lws_dll2 *d);
37c19
<         int (*cb)(struct lws_dll2 *d, void *user));
——
>         lws_dll2_foreach_cb_t cb);

```

46a29,35

```
> LWS_VISIBLE LWS_EXTERN void
> lws_dll2_add_sorted_priv(lws_dll2_t *d, lws_dll2_owner_t *own, void *priv,
>
>         int (*compare3)(void *priv, const lws_dll2_t *d,
>
>         const lws_dll2_t *i));
> LWS_VISIBLE LWS_EXTERN void *
> _lws_dll2_search_sz_pl(lws_dll2_owner_t *own, const char *name, size_t namelen,
>
>         size_t dll2_ofs, size_t ptr_ofs);
```

改变 lws-dsh.h

9a10,11

```
> LWS_VISIBLE LWS_EXTERN size_t
> lws_dsh_get_size(struct lws_dsh *dsh, int kind);
```

减少头文件: lws-esp32.h

增加头文件: lws-eventlib-exports.h

增加头文件: lws-fault-injection.h

改变 lws-freertos.h

0a1

```
> extern const lws_gpio_ops_t lws_gpio_plat;
```

改变 lws-genec.h

21c21

```
<
struct lws_gencrypto_keyelem *el);
-----
>
const struct lws_gencrypto_keyelem *el);
```

改变 lws-genrsa.h

2c2,3

```
< lws_genrsa_create(struct lws_genrsa_ctx *ctx, struct lws_gencrypto_keyelem *el,
-----
> lws_genrsa_create(struct lws_genrsa_ctx *ctx,
>
const struct lws_gencrypto_keyelem *el,
```

增加头文件: lws-gpio.h

改变 lws-http.h

27a28,31

```
> LWS_VISIBLE LWS_EXTERN int
> lws_hdr_custom_name_foreach(struct lws *wsi, lws_hdr_custom_fe_cb_t cb, void *opaque);
> LWS_VISIBLE LWS_EXTERN int
> lws_get_urlarg_by_name_safe(struct lws *wsi, const char *name, char *buf, int len);
```

29c33,34

```
< lws_get_urlarg_by_name(struct lws *wsi, const char *name, char *buf, int len);
-----
> lws_get_urlarg_by_name(struct lws *wsi, const char *name, char *buf, int len)
> ;
```

62a68,73

```
> lws_http_date_render_from_unix(char *buf, size_t len, const time_t *t);
> LWS_VISIBLE LWS_EXTERN int
```

```

> lws_http_date_parse_unix(const char *b, size_t len, time_t *t);
> LWS_VISIBLE LWS_EXTERN int
> lws_http_check_retry_after(struct lws *wsi, lws_usec_t *us_interval_in_out);
> LWS_VISIBLE LWS_EXTERN int
82c93,95
< lws_h2_update_peer_txcredit(struct lws *wsi, int sid, int bump);
—
> lws_http_cookie_get(struct lws *wsi, const char *name, char *buf, size_t *max);
> LWS_VISIBLE LWS_EXTERN int
> lws_h2_update_peer_txcredit(struct lws *wsi, unsigned int sid, int bump);
增加头文件: lws-i2c.h
增加头文件: lws-ili9341-spi.h
改变 lws-jws.h
66a67,89
> LWS_VISIBLE LWS_EXTERN int
> lws_jwt_signed_validate(struct lws_context *ctx, struct lws_jwk *jwk,
>
>         const char *alg_list, const char *com, size_t len,
>         char *temp, int tl, char *out, size_t *out_len);
> LWS_VISIBLE LWS_EXTERN int
> lws_jwt_sign_compact(struct lws_context *ctx, struct lws_jwk *jwk,
>
>         const char *alg, char *out, size_t *out_len, char *temp,
>         int tl, const char *format, ...) LWS_FORMAT(8);
> LWS_VISIBLE LWS_EXTERN int
> lws_jwt_sign_via_info(struct lws_context *ctx, struct lws_jwk *jwk,
>
>         const struct lws_jwt_sign_info *info, const char *format, ...) LWS_FORMAT(4);
> LWS_VISIBLE LWS_EXTERN int
> lws_jwt_token_sanity(const char *in, size_t in_len,
>
>         const char *iss, const char *aud, const char *csrf_in,
>         char *sub, size_t sub_len, unsigned long *exp_unix_time);
> LWS_VISIBLE LWS_EXTERN int
> lws_jwt_sign_token_set_http_cookie(struct lws *wsi,
>
>         const struct lws_jwt_sign_set_cookie *i,
>         uint8_t **p, uint8_t *end);
> LWS_VISIBLE LWS_EXTERN int
> lws_jwt_get_http_cookie_validate_jwt(struct lws *wsi,
>
>         struct lws_jwt_sign_set_cookie *i,
>         char *out, size_t *out_len);
增加头文件: lws-lecp.h
增加头文件: lws-led.h
改变 lws-logs.h
0a1
> struct lws_log_ctx;
2,4c3,39
< lwsl_timestamp(int level, char *p, int len);

```

< LWS_VISIBLE LWS_EXTERN void _lws_log(int filter, const char *format, ...) LWS_FORMAT(2);

< LWS_VISIBLE LWS_EXTERN void _lws_logv(int filter, const char *format, va_list vl);

——

> lws_timestamp(int level, char *p, size_t len);

> LWS_VISIBLE LWS_EXTERN void

> _lws_log(int filter, const char *format, ...) LWS_FORMAT(2);

> LWS_VISIBLE LWS_EXTERN void

> _lws_logv(int filter, const char *format, va_list vl);

> struct lws_vhost;

> struct lws;

> LWS_VISIBLE LWS_EXTERN struct lws_log_cx *

> lws_context_get_cx(struct lws_context *cx);

> LWS_VISIBLE LWS_EXTERN struct lws_log_cx *

> lws_vhost_get_cx(struct lws_vhost *vh);

> LWS_VISIBLE LWS_EXTERN struct lws_log_cx *

> lws_wsi_get_cx(struct lws *wsi);

> struct lws_ss_handle;

> struct lws_sspc_handle;

> LWS_VISIBLE LWS_EXTERN struct lws_log_cx *

> lws_ss_get_cx(struct lws_ss_handle *ss);

> LWS_VISIBLE LWS_EXTERN struct lws_log_cx *

> lws_sspc_get_cx(struct lws_sspc_handle *ss);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_emit_cx_file(struct lws_log_cx *cx, int level, const char *line,

> size_t len);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_use_cx_file(struct lws_log_cx *cx, int _new);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_prepend_context(struct lws_log_cx *cx, void *obj, char **p, char *e);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_prepend_vhost(struct lws_log_cx *cx, void *obj, char **p, char *e);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_prepend_wsi(struct lws_log_cx *cx, void *obj, char **p, char *e);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_prepend_ss(struct lws_log_cx *cx, void *obj, char **p, char *e);

> LWS_VISIBLE LWS_EXTERN void

> lws_log_prepend_sspc(struct lws_log_cx *cx, void *obj, char **p, char *e);

> LWS_VISIBLE LWS_EXTERN void

> _lws_log_cx(lws_log_cx_t *cx, lws_log_prepend_cx_t prep, void *obj,

> int filter, const char *_fun, const char *format, ...) LWS_FORMAT(6);

7a43,45

> lws_hexdump_level_cx(lws_log_cx_t *cx, lws_log_prepend_cx_t prep, void *obj,

> int hexdump_level, const void *vbuf, size_t len);

> LWS_VISIBLE LWS_EXTERN void

14,15c52

```
< lws_set_log_level(int level,  
<         void (*log_emit_function)(int level, const char *line));  
____
```

```
> lws_set_log_level(int level, lws_log_emit_t log_emit_function);
```

23a61,65

```
> struct lws;  
> LWS_VISIBLE LWS_EXTERN const char *  
> lws_wsi_tag(struct lws *wsi);  
> LWS_VISIBLE LWS_EXTERN void  
> lws_l_refcount_cx(lws_log_cx_t *cx, int _new);
```

改变 lws-lwsac.h

20c20

```
< lwsac_extend(struct lwsac *head, int amount);  
____
```

```
> lwsac_extend(struct lwsac *head, size_t amount);
```

增加头文件: lws-map.h

增加头文件: lws-metrics.h

改变 lws-misc.h

13a14,18

```
> LWS_VISIBLE LWS_EXTERN int  
> lws_buflist_linear_use(struct lws_buflist **head, uint8_t *buf, size_t len);  
> LWS_VISIBLE LWS_EXTERN int  
> lws_buflist_fragment_use(struct lws_buflist **head, uint8_t *buf,  
>         size_t len, char *frag_first, char *frag_fin);
```

21a27,32

```
> LWS_VISIBLE LWS_EXTERN const char *  
> lws_nstrstr(const char *buf, size_t len, const char *name, size_t nl);  
> LWS_VISIBLE LWS_EXTERN const char *  
> lws_json_simple_find(const char *buf, size_t len, const char *name, size_t *alen);  
> LWS_VISIBLE LWS_EXTERN int  
> lws_json_simple_strcmp(const char *buf, size_t len, const char *name, const char *comp);
```

23a35,38

```
> LWS_VISIBLE LWS_EXTERN void  
> lws_hex_from_byte_array(const uint8_t *src, size_t slen, char *dest, size_t len);  
> LWS_VISIBLE LWS_EXTERN int  
> lws_hex_random(struct lws_context *context, char *dest, size_t len);
```

33a49,50

```
> LWS_VISIBLE LWS_EXTERN int  
> lws_wsi_tsi(struct lws *wsi);
```

59c76

```
< lws_get_effective_uid_gid(struct lws_context *context, int *uid, int *gid);  
____  
> lws_get_effective_uid_gid(struct lws_context *context, uid_t *uid, gid_t *gid);
```


90a108,111

```
> LWS_VISIBLE LWS_EXTERN int
> lws_dir_rm_rf_cb(const char *dirpath, void *user, struct lws_dir_entry *lde);
> LWS_VISIBLE LWS_EXTERN int
> lws_dir_glob_cb(const char *dirpath, void *user, struct lws_dir_entry *lde);
```

98a120,123

```
> lws_tls_jit_trust_blob_queuey_skid(const void *_blob, size_t blen,
>
> const uint8_t *skid, size_t skid_len,
>
> const uint8_t **prpder, size_t *prder_len);
> LWS_VISIBLE LWS_EXTERN int
```

104,106c129,131

```
< LWS_VISIBLE LWS_EXTERN const lws_humanize_unit_t humanize_schema_si[7];
< LWS_VISIBLE LWS_EXTERN const lws_humanize_unit_t humanize_schema_si_bytes[7];
< LWS_VISIBLE LWS_EXTERN const lws_humanize_unit_t humanize_schema_us[8];
```

```
> LWS_VISIBLE extern const lws_humanize_unit_t humanize_schema_si[7];
> LWS_VISIBLE extern const lws_humanize_unit_t humanize_schema_si_bytes[7];
> LWS_VISIBLE extern const lws_humanize_unit_t humanize_schema_us[8];
```

108c133

```
< lws_humanize(char *buf, int len, uint64_t value,
```

```
> lws_humanize(char *buf, size_t len, uint64_t value,
```

132c157

```
< lws_spawn_stdwsi_closed(struct lws_spawn_piped *lsp);
```

```
> lws_spawn_stdwsi_closed(struct lws_spawn_piped *lsp, struct lws *wsi);
```

增加头文件: lws-netdev.h

改变 lws-network-helper.h

15a16,18

```
> lws_sa46_on_net(const lws_sockaddr46 *sa46a, const lws_sockaddr46 *sa46_net,
>
> int net_len);
> LWS_VISIBLE LWS_EXTERN int
```

减少头文件: lws-plugin-generic-sessions.h

减少头文件: lws-plugin-ssh.h

改变 lws-protocols-plugins.h

13a14,16

```
> lws_vhd_find_by_pvo(struct lws_context *cx, const char *protname,
>
> const char *pvo_name, const char *pvo_value);
> LWS_VISIBLE LWS_EXTERN void *
```

22a26,50

```
> LWS_VISIBLE LWS_EXTERN int
> lws_plugins_init(struct lws_plugin **ppplugin, const char * const *d,
>
> const char *_class, const char *filter,
>
> each_plugin_cb_t each, void *each_user);
```

```

> LWS_VISIBLE LWS_EXTERN int
> lws_plugins_destroy(struct lws_plugin **pplugin, each_plugin_cb_t each,
> void *each_user);
> extern const struct lws_protocols post_demo_protocols[1];
> extern const struct lws_protocols lws_raw_proxy_protocols[1];
> extern const struct lws_protocols lws_status_protocols[1];
> extern const struct lws_protocols lws_mirror_protocols[1];
> extern const struct lws_protocols lws_ssh_base_protocols[2];
> extern const struct lws_protocols post_demo_protocols[1];
> extern const struct lws_protocols dumb_increment_protocols[1];
> extern const struct lws_protocols deaddrop_protocols[1];
> extern const struct lws_protocols lws_raw_test_protocols[1];
> extern const struct lws_protocols lws_sshd_demo_protocols[1];
> extern const struct lws_protocols lws_acme_client_protocols[1];
> extern const struct lws_protocols client_loopback_test_protocols[1];
> extern const struct lws_protocols fulltext_demo_protocols[1];
> extern const struct lws_protocols lws_openmetrics_export_protocols[
> 4
> 3
> 1
> ];

```

改变 lws-purify.h

2c2

```
< lws_sql_purify(char *escaped, const char *string, int len);
```

```
> lws_sql_purify(char *escaped, const char *string, size_t len);
```

13c13

```
< int len);
```

```
> size_t len);
```

15c15

```
< lws_plat_write_file(const char *filename, void *buf, int len);
```

```
> lws_plat_write_file(const char *filename, void *buf, size_t len);
```

17c17

```
< lws_plat_read_file(const char *filename, void *buf, int len);
```

```
> lws_plat_read_file(const char *filename, void *buf, size_t len);
```

增加头文件: lws-pwm.h

改变 lws-secure-streams-client.h

1a2,5

```
> LWS_VISIBLE LWS_EXTERN void
```

```
> lws_log_prepend_ssps(struct lws_log_cx *cx, void *obj, char **p, char *e);
```

```
> LWS_VISIBLE LWS_EXTERN struct lws_log_cx *
```

```

> lws_sspc_get_cx(struct lws_sspc_handle *ss);
4,5c8,9
<
    void *opaque_user_data, struct lws_sspc_handle **ppss,
<
    struct lws_sequencer *seq_owner, const char **ppayload_fmt);
----
>
    void *opaque_user_data, struct lws_sspc_handle **ppss,
>
    struct lws_sequencer *seq_owner, const char **ppayload_fmt);
8c12
< LWS_VISIBLE LWS_EXTERN void
----
> LWS_VISIBLE LWS_EXTERN lws_ss_state_return_t
10c14,16
< LWS_VISIBLE LWS_EXTERN int
----
> LWS_VISIBLE LWS_EXTERN lws_ss_state_return_t
> lws_sspc_request_tx_len(struct lws_sspc_handle *h, unsigned long len);
> LWS_VISIBLE LWS_EXTERN lws_ss_state_return_t
18c24
< LWS_VISIBLE LWS_EXTERN const struct lws_protocols lws_sspc_protocols[];
----
> LWS_VISIBLE extern const struct lws_protocols lws_sspc_protocols[2];
23c29,32
<
    void *value, size_t len);
----
>
    const void *value, size_t len);
> LWS_VISIBLE LWS_EXTERN int
> lws_sspc_get_metadata(struct lws_sspc_handle *h, const char *name,
>
    const void **value, size_t *len);
27a37,54
> LWS_VISIBLE LWS_EXTERN void
> lws_sspc_start_timeout(struct lws_sspc_handle *h, unsigned int timeout_ms);
> LWS_VISIBLE LWS_EXTERN void
> lws_sspc_cancel_timeout(struct lws_sspc_handle *h);
> LWS_VISIBLE LWS_EXTERN void *
> lws_sspc_to_user_object(struct lws_sspc_handle *h);
> LWS_VISIBLE LWS_EXTERN void
> lws_sspc_change_handlers(struct lws_sspc_handle *h,
>
    lws_ss_state_return_t (*rx)(void *userobj, const uint8_t *buf,
>
        size_t len, int flags),
>
    lws_ss_state_return_t (*tx)(void *userobj, lws_ss_tx_ordinal_t ord,
>
        uint8_t *buf, size_t *len, int *flags),
>
    lws_ss_state_return_t (*state)(void *userobj, void *h_src
>
        ,
>
        lws_ss_constate_t state,

```

```

>                                lws_ss_tx_ordinal_t ack));
> const char *
> lws_sspc_tag(struct lws_sspc_handle *h);
改变 lws-secure-streams.h
2c2,3
< LWS_VISIBLE LWS_EXTERN int
——
> struct lws_ss_policy;
> LWS_VISIBLE LWS_EXTERN int LWS_WARN_UNUSED_RESULT
8c9
< LWS_VISIBLE LWS_EXTERN void
——
> LWS_VISIBLE LWS_EXTERN lws_ss_state_return_t LWS_WARN_UNUSED_RESULT
10c11
< LWS_VISIBLE LWS_EXTERN void
——
> LWS_VISIBLE LWS_EXTERN lws_ss_state_return_t LWS_WARN_UNUSED_RESULT
12c13
< LWS_VISIBLE LWS_EXTERN int
——
> LWS_VISIBLE LWS_EXTERN lws_ss_state_return_t LWS_WARN_UNUSED_RESULT
21a23,28
> LWS_VISIBLE LWS_EXTERN void
> lws_ss_start_timeout(struct lws_ss_handle *h, unsigned int timeout_ms);
> LWS_VISIBLE LWS_EXTERN void
> lws_ss_cancel_timeout(struct lws_ss_handle *h);
> LWS_VISIBLE LWS_EXTERN void *
> lws_ss_to_user_object(struct lws_ss_handle *h);
24c31
< LWS_VISIBLE LWS_EXTERN int
——
> LWS_VISIBLE LWS_EXTERN int LWS_WARN_UNUSED_RESULT
26c33,47
<                                void *value, size_t len);
——
>                                const void *value, size_t len);
> LWS_VISIBLE LWS_EXTERN int LWS_WARN_UNUSED_RESULT
> lws_ss_alloc_set_metadata(struct lws_ss_handle *h, const char *name,
>                                const void *value, size_t len);
> LWS_VISIBLE LWS_EXTERN int
> lws_ss_get_metadata(struct lws_ss_handle *h, const char *name,
>                                const void **value, size_t *len);
> LWS_VISIBLE LWS_EXTERN void
> lws_ss_server_ack(struct lws_ss_handle *h, int nack);

```

```

> LWS_VISIBLE LWS_EXTERN void
> lws_ss_server_foreach_client(struct lws_ss_handle *h, lws_ssfec_cb cb,
>                             void *arg);
> LWS_VISIBLE LWS_EXTERN void
> lws_ss_change_handlers(struct lws_ss_handle *h, lws_sscb_rx rx, lws_sscb_tx tx,
>                        lws_sscb_state state);
30a52,60
> LWS_VISIBLE LWS_EXTERN const char *
> lws_ss_tag(struct lws_ss_handle *h);
> LWS_VISIBLE LWS_EXTERN int
> lws_ss_sigv4_set_aws_key(struct lws_context* context, uint8_t idx,
>                          const char * keyid, const char * key);
> LWS_VISIBLE LWS_EXTERN int
> lws_aws_filesystem_credentials_helper(const char *path, const char *kid,
>                                     const char *ak, char **aws_keyid,
>                                     char **aws_key);
改变 lws-secure-streams-policy.h
0a1,12
> LWS_VISIBLE LWS_EXTERN int
> lws_ss_policy_parse_begin(struct lws_context *context, int overlay);
> LWS_VISIBLE LWS_EXTERN int
> lws_ss_policy_parse_abandon(struct lws_context *context);
> LWS_VISIBLE LWS_EXTERN int
> lws_ss_policy_parse(struct lws_context *context, const uint8_t *buf, size_t len);
> LWS_VISIBLE LWS_EXTERN int
> lws_ss_policy_overlay(struct lws_context *context, const char *overlay);
> LWS_VISIBLE LWS_EXTERN const lws_ss_policy_t *
> lws_ss_policy_get(struct lws_context *context);
> LWS_VISIBLE LWS_EXTERN const lws_ss_auth_t *
> lws_ss_auth_get(struct lws_context *context);
改变 lws-service.h
21c21,22
< lws_libuv_static_refcount_add(uv_handle_t *, struct lws_context *context);
——
> lws_libuv_static_refcount_add(uv_handle_t *, struct lws_context *context,
>                               int tsi);
增加头文件: lws-settings.h
增加头文件: lws-smd.h
增加头文件: lws-spi.h
增加头文件: lws-ssd1306-i2c.h
减少头文件: lws-stats.h
改变 lws-struct.h
28c28
<         sqlite3 **pdb);

```

> char create_if_missing, sqlite3 **pdb);

改变 lws-system.h

18a19,22

> LWS_EXTERN LWS_VISIBLE int

> lws_tls_jit_trust_get_cert_cb(struct lws_context *cx, void *got_opaque,

> const uint8_t *skid, size_t skid_len,

> const uint8_t *der, size_t der_len);

29c33

< int

> LWS_EXTERN LWS_VISIBLE int

32c36

< int

> LWS_EXTERN LWS_VISIBLE int

34c38

< int

> LWS_EXTERN LWS_VISIBLE int

35a40,47

> LWS_EXTERN LWS_VISIBLE int

> lws_system_cpd_start(struct lws_context *context);

> LWS_EXTERN LWS_VISIBLE void

> lws_system_cpd_start_defer(struct lws_context *cx, lws_usec_t defer_us);

> LWS_EXTERN LWS_VISIBLE void

> lws_system_cpd_set(struct lws_context *context, lws_cpd_result_t result);

> LWS_EXTERN LWS_VISIBLE lws_cpd_result_t

> lws_system_cpd_state_get(struct lws_context *context);

改变 lws-threadpool.h

16c16,18

< lws_threadpool_dequeue(struct lws *wsi);

> lws_threadpool_dequeue(struct lws *wsi) LWS_WARN_DEPRECATED;

> LWS_VISIBLE LWS_EXTERN int

> lws_threadpool_dequeue_task(struct lws_threadpool_task *task);

19c21,26

< struct lws_threadpool_task **task, void **user);

> struct lws_threadpool_task **task, void **user)

> LWS_WARN_DEPRECATED;

> LWS_VISIBLE LWS_EXTERN enum lws_threadpool_task_status

> lws_threadpool_task_status(struct lws_threadpool_task *task, void **user);

> LWS_VISIBLE LWS_EXTERN enum lws_threadpool_task_status

```

> lws_threadpool_task_status_noreap(struct lws_threadpool_task *task);
23a31,41
> LWS_VISIBLE LWS_EXTERN struct lws_threadpool_task *
> lws_threadpool_get_task_wsi(struct lws *wsi);
> LWS_VISIBLE LWS_EXTERN struct lws_threadpool_task *
> lws_threadpool_get_task_ss(struct lws_ss_handle *ss);
> LWS_VISIBLE LWS_EXTERN int
> lws_threadpool_foreach_task_wsi(struct lws *wsi, void *user,
>                                int (*cb)(struct lws_threadpool_task *task,
>                                void *user));
> LWS_VISIBLE LWS_EXTERN int
> lws_threadpool_foreach_task_ss(struct lws_ss_handle *ss, void *user,
>                                int (*cb)(struct lws_threadpool_task *task, void *user));
改变 lws-timeout-timer.h
7,14d6
< LWS_VISIBLE LWS_EXTERN int
< lws_timed_callback_vh_protocol(struct lws_vhost *vh,
<                                const struct lws_protocols *prot,
<                                int reason, int secs);
< LWS_VISIBLE LWS_EXTERN int
< lws_timed_callback_vh_protocol_us(struct lws_vhost *vh,
<                                const struct lws_protocols *prot, int reason,
<                                lws_usec_t us);
17,18c9,24
< lws_sul_schedule(struct lws_context *context, int tsi,
<                lws_sorted_usec_list_t *sul, sul_cb_t cb, lws_usec_t us);
——
> lws_sul2_schedule(struct lws_context *context, int tsi, int flags,
>                lws_sorted_usec_list_t *sul);
> LWS_VISIBLE LWS_EXTERN void
> lws_sul_cancel(lws_sorted_usec_list_t *sul);
> LWS_VISIBLE LWS_EXTERN int
> lws_sul_earliest_wakeable_event(struct lws_context *ctx, lws_usec_t *pearliest);
> LWS_VISIBLE LWS_EXTERN void
> lws_sul_schedule(struct lws_context *ctx, int tsi, lws_sorted_usec_list_t *sul,
>                sul_cb_t _cb, lws_usec_t _us);
> LWS_VISIBLE LWS_EXTERN void
> lws_sul_schedule_wakesuspend(struct lws_context *ctx, int tsi,
>                lws_sorted_usec_list_t *sul, sul_cb_t _cb,
>                lws_usec_t _us);
> LWS_VISIBLE LWS_EXTERN void
> lws_sul_debug_zombies(struct lws_context *ctx, void *po, size_t len,
>                const char *destroy_description);
22,23c28

```

```

< __lws_sul_insert(lws_dll2_owner_t *own, lws_sorted_usec_list_t *sul,
<         lws_usec_t us);
____
> __lws_sul_insert(lws_dll2_owner_t *own, lws_sorted_usec_list_t *sul);
25c30
< __lws_sul_service_ripe(lws_dll2_owner_t *own, lws_usec_t usnow);
____
> __lws_sul_service_ripe(lws_dll2_owner_t *own, int own_len, lws_usec_t usnow);
增加头文件: lws-tls-sessions.h
改变 lws-tokenize.h
14a15,17
> LWS_VISIBLE LWS_EXTERN int
> lws_strcmp_wildcard(const char *wildcard, size_t wlen, const char *check,
>         size_t clen);
改变 lws-writeable.h
22c22
< lws_callback_vhost_protocols(struct lws *wsi, int reason, void *in, int len)
____
> lws_callback_vhost_protocols(struct lws *wsi, int reason, void *in, size_t len)
改变 lws-x509.h
14,15c14,15
< lws_x509_jwk_privkey_pem(struct lws_jwk *jwk, void *pem, size_t len,
<         const char *passphrase);
____
> lws_x509_jwk_privkey_pem(struct lws_context *cx, struct lws_jwk *jwk,
>         void *pem, size_t len, const char *passphrase);

```

libreofficekit-devel

改变 LibreOfficeKitEnums.h

3c3

改变 LibreOfficeKit.hxx

67c67,68

```

<         const double dpiscale = 1.0)
____
>         const double dpiscale = 1.0,
>         const int viewId = -1)
69,70c70,71
<         return mpDoc->pClass->paintWindowDPI(mpDoc, nWindowId, pBuffer,
<         x, y, width, height, dpiscale);
____
>         return mpDoc->pClass->paintWindowForView(mpDoc, nWindowId, pBuffer, x, y,

```



```

>                                     width, height, dpiscale, viewId);
107a109,112
>     void sendDialogEvent(unsigned nWindowId, const char* pArguments = NULL)
>     {
>         mpDoc->pClass->sendDialogEvent(mpDoc, nWindowId, pArguments);
>     }
119a125,143
>     int getSelectionType()
>     {
>         return mpDoc->pClass->getSelectionType(mpDoc);
>     }
>     bool getClipboard(const char **pMimeTypes,
>                       size_t      *pOutCount,
>                       char        ***pOutMimeTypes,
>                       size_t      **pOutSizes,
>                       char        ***pOutStreams)
>     {
>         return mpDoc->pClass->getClipboard(mpDoc, pMimeTypes, pOutCount, pOutMimeTypes, pOutSizes,
pOutStreams);
>     }
>     bool setClipboard(const size_t  nInCount,
>                       const char    **pInMimeTypes,
>                       const size_t *pInSizes,
>                       const char    **pInStreams)
>     {
>         return mpDoc->pClass->setClipboard(mpDoc, nInCount, pInMimeTypes, pInSizes, pInStreams);
>     }
178c202,203
<                                     int *pFontHeight)
-----
>                                     int *pFontHeight,
>                                     int pOrientation=0)
180c205,208
<         return mpDoc->pClass->renderFont(mpDoc, pFontName, pChar, pFontWidth, pFontHeight);
-----
>         if (LIBREOFFICEKIT_DOCUMENT_HAS(mpDoc, renderFontOrientation))
>             return mpDoc->pClass->renderFontOrientation(mpDoc, pFontName, pChar, pFontWidth,
pFontHeight, pOrientation);
>         else
>             return mpDoc->pClass->renderFont(mpDoc, pFontName, pChar, pFontWidth, pFontHeight);
247a276,293
>     }
>     void selectPart(int nPart, int nSelect)
>     {

```

```

>         mpDoc->pClass->selectPart(mpDoc, nPart, nSelect);
>     }
>     void moveSelectedParts(int nPosition, bool bDuplicate)
>     {
>         mpDoc->pClass->moveSelectedParts(mpDoc, nPosition, bDuplicate);
>     }
>     void resizeWindow(unsigned nWindowId,
>                         const int width,
>                         const int height)
>     {
>         return mpDoc->pClass->resizeWindow(mpDoc, nWindowId, width, height);
>     }
>     void removeTextContext(unsigned nWindowId, int nBefore, int nAfter)
>     {
>         mpDoc->pClass->removeTextContext(mpDoc, nWindowId, nBefore, nAfter);

```

lcr-devel

增加头文件: container_rename_request.h

增加头文件: container_rename_response.h

增加头文件: container_resize_request.h

增加头文件: container_resize_response.h

改变 defs.h

9a10,13

```
> defs_blkio_entry;
```

```
> void free_defs_blkio_entry(defs_blkio_entry *ptr);
```

```
> defs_blkio_entry *make_defs_blkio_entry(yajl_val tree, const struct parser_context *ctx, parser_error *err);
```

```
> yajl_gen_status gen_defs_blkio_entry(yajl_gen g, const defs_blkio_entry *ptr, const struct parser_context *ctx,
parser_error *err);
```

143a148,151

```
> defs_resources_files;
```

```
> void free_defs_resources_files(defs_resources_files *ptr);
```

```
> defs_resources_files *make_defs_resources_files(yajl_val tree, const struct parser_context *ctx, parser_error *err);
```

```
> yajl_gen_status gen_defs_resources_files(yajl_gen g, const defs_resources_files *ptr, const struct parser_context
*ctx, parser_error *err);
```

增加头文件: image_pull_image_request.h

增加头文件: image_pull_image_response.h

改变 json_common.h

7a8

```
> void *smart_safe_malloc(size_t count, size_t extra, size_t unit_size);
```

改变 lcrcontainer.h

2,5d1

```
< struct lcr_container_info *lcr_container_info_get(const char *name, const char *lcrpath);
```

```

< void lcr_container_info_free(struct lcr_container_info *info);
< int lcr_list_all_containers(const char *lcrpath, struct lcr_container_info **info_arr);
< void lcr_containers_info_free(struct lcr_container_info **info_arr, size_t size);
增加头文件： shim_client_cgroup_resources.h
改变 shim_client_runtime_stats.h
25a26,29
> shim_client_runtime_stats_data_blkio;
> void free_shim_client_runtime_stats_data_blkio(shim_client_runtime_stats_data_blkio *ptr);
> shim_client_runtime_stats_data_blkio *make_shim_client_runtime_stats_data_blkio(yajl_val tree, const struct
parser_context *ctx, parser_error *err);
> yajl_gen_status gen_shim_client_runtime_stats_data_blkio(yajl_gen g, const
shim_client_runtime_stats_data_blkio *ptr, const struct parser_context *ctx, parser_error *err);

```

libdvdnav-devel

```

改变 dvdnav.h
4c4,10
< dvdnav_open_stream(dvdnav_t **dest, void *stream, dvdnav_stream_cb *stream_cb);
----
> dvdnav_open_stream(dvdnav_t **dest, void *priv, dvdnav_stream_cb *stream_cb);
> dvdnav_status_t dvdnav_open2(dvdnav_t **dest,
> void *, const dvdnav_logger_cb *,
> const char *path);
> dvdnav_status_t dvdnav_open_stream2(dvdnav_t **dest,
> void *priv, const dvdnav_logger_cb *,
> dvdnav_stream_cb *stream_cb);
10a17
> const char* dvdnav_version(void);
27a35
> dvdnav_status_t dvdnav_get_number_of_angles(dvdnav_t *self, int32_t title, int32_t *angles);
增加头文件： version.h

```

e2fsprogs-devel

```

改变 ext2fs.h
539a540
> extern int ext2fs_current_level_get(ext2_extent_handle_t handle);

```

wireshark-devel

变动很大，略

multipath-tools-devel

改变 mpath_persist.h

2a3

> extern int libmpathpersist_init (void);

4a6

> extern int libmpathpersist_exit (void);

增加头文件: mpath_valid.h

hiredis-devel

改变 ae.h

47c47

< free(e);

> hi_free(e);

54c54,56

< e = (redisAeEvents*)malloc(sizeof(*e));

> e = (redisAeEvents*)hi_malloc(sizeof(*e));

> if (e == NULL)

> return REDIS_ERR;

增加头文件: alloc.h

改变 async.h

3a4

> redisAsyncContext *redisAsyncConnectWithOptions(const redisOptions *options);

10a12,13

> redisAsyncPushFn *redisAsyncSetPushCallback(redisAsyncContext *ac, redisAsyncPushFn *fn);

> int redisAsyncSetTimeout(redisAsyncContext *ac, struct timeval tv);

14a18,20

> void redisAsyncHandleTimeout(redisAsyncContext *ac);

> void redisAsyncRead(redisAsyncContext *ac);

> void redisAsyncWrite(redisAsyncContext *ac);

改变 glib.h

99a100,101

> if (source == NULL)

> return NULL;

改变 hiredis.h

0a1,3

> struct timeval;

> struct redisAsyncContext;

> struct redisContext;

9a13,16

> struct redisSsl;

```

> typedef unsigned long long redisFD;
> typedef unsigned long redisFD;
> redisContext *redisConnectWithOptions(const redisOptions *options);
20c27
< redisContext *redisConnectFd(int fd);
——
> redisContext *redisConnectFd(redisFD fd);
21a29
> redisPushFn *redisSetPushCallback(redisContext *c, redisPushFn *fn);
25c33
< int redisFreeKeepFd(redisContext *c);
——
> redisFD redisFreeKeepFd(redisContext *c);
改变 ivykis.h
28c28
<     free(e);
——
>     hi_free(e);
35c35,37
<     e = (redisIvykisEvents*)malloc(sizeof(*e));
——
>     e = (redisIvykisEvents*)hi_malloc(sizeof(*e));
>     if (e == NULL)
>         return REDIS_ERR;
改变 libevent.h
1,4c1,2
< static void redisLibeventReadEvent(int fd, short event, void *arg) {
<     ((void)fd); ((void)event);
<     redisLibeventEvents *e = (redisLibeventEvents*)arg;
<     redisAsyncHandleRead(e->context);
——
> static void redisLibeventDestroy(redisLibeventEvents *e) {
>     hi_free(e);
6,7c4,5
< static void redisLibeventWriteEvent(int fd, short event, void *arg) {
<     ((void)fd); ((void)event);
——
> static void redisLibeventHandler(int fd, short event, void *arg) {
>     ((void)fd);
9c7,41
<     redisAsyncHandleWrite(e->context);
——
>     e->state |= REDIS_LIBEVENT_ENTERED;
>     if ((event & EV_TIMEOUT) && (e->state & REDIS_LIBEVENT_DELETED) == 0) {

```

```

>     redisAsyncHandleTimeout(e->context);
>     CHECK_DELETED();
> }
> if ((event & EV_READ) && e->context && (e->state & REDIS_LIBEVENT_DELETED) == 0) {
>     redisAsyncHandleRead(e->context);
>     CHECK_DELETED();
> }
> if ((event & EV_WRITE) && e->context && (e->state & REDIS_LIBEVENT_DELETED) == 0) {
>     redisAsyncHandleWrite(e->context);
>     CHECK_DELETED();
> }
> e->state &= ~REDIS_LIBEVENT_ENTERED;
> }
> static void redisLibeventUpdate(void *privdata, short flag, int isRemove) {
>     redisLibeventEvents *e = (redisLibeventEvents *)privdata;
>     const struct timeval *tv = e->tv.tv_sec || e->tv.tv_usec ? &e->tv : NULL;
>     if (isRemove) {
>         if ((e->flags & flag) == 0) {
>             return;
>         } else {
>             e->flags &= ~flag;
>         }
>     } else {
>         if (e->flags & flag) {
>             return;
>         } else {
>             e->flags |= flag;
>         }
>     }
>     event_del(e->ev);
>     event_assign(e->ev, e->base, e->context->c.fd, e->flags | EV_PERSIST,
>         redisLibeventHandler, privdata);
>     event_add(e->ev, tv);
12,13c44
<     redisLibeventEvents *e = (redisLibeventEvents *)privdata;
<     event_add(&e->rev,NULL);
-----
>     redisLibeventUpdate(privdata, EV_READ, 0);
16,17c47
<     redisLibeventEvents *e = (redisLibeventEvents *)privdata;
<     event_del(&e->rev);
-----
>     redisLibeventUpdate(privdata, EV_READ, 1);
20,21c50

```

```

<     redisLibeventEvents *e = (redisLibeventEvents*)privdata;
<     event_add(&e->wev,NULL);
-----
>     redisLibeventUpdate(privdata, EV_WRITE, 0);
24,25c53
<     redisLibeventEvents *e = (redisLibeventEvents*)privdata;
<     event_del(&e->wev);
-----
>     redisLibeventUpdate(privdata, EV_WRITE, 1);
29,31c57,74
<     event_del(&e->rev);
<     event_del(&e->wev);
<     free(e);
-----
>     if (!e) {
>         return;
>     }
>     event_del(e->ev);
>     event_free(e->ev);
>     e->ev = NULL;
>     if (e->state & REDIS_LIBEVENT_ENTERED) {
>         e->state |= REDIS_LIBEVENT_DELETED;
>     } else {
>         redisLibeventDestroy(e);
>     }
> }
> static void redisLibeventSetTimeout(void *privdata, struct timeval tv) {
>     redisLibeventEvents *e = (redisLibeventEvents *)privdata;
>     short flags = e->flags;
>     e->flags = 0;
>     e->tv = tv;
>     redisLibeventUpdate(e, flags, 0);
38c81,83
<     e = (redisLibeventEvents*)malloc(sizeof(*e));
-----
>     e = (redisLibeventEvents*)hi_calloc(1, sizeof(*e));
>     if (e == NULL)
>         return REDIS_ERR;
44a90
>     ac->ev.scheduleTimer = redisLibeventSetTimeout;
46,49c92,93
<     event_set(&e->rev,c->fd,EV_READ,redisLibeventReadEvent,e);
<     event_set(&e->wev,c->fd,EV_WRITE,redisLibeventWriteEvent,e);
<     event_base_set(base,&e->rev);

```

```

<     event_base_set(base,&e->wbev);
-----
>     e->ev = event_new(base, c->fd, EV_READ | EV_WRITE, redisLibeventHandler, e);
>     e->base = base;
改变 libev.h
48a49,54
> static void redisLibevStopTimer(void *privdata) {
>     redisLibevEvents *e = (redisLibevEvents*)privdata;
>     struct ev_loop *loop = e->loop;
>     ((void)loop);
>     ev_timer_stop(EV_A_ &e->timer);
> }
53c59,76
<     free(e);
-----
>     redisLibevStopTimer(privdata);
>     hi_free(e);
> }
> static void redisLibevTimeout(EV_P_ ev_timer *timer, int revents) {
>     ((void)revents);
>     redisLibevEvents *e = (redisLibevEvents*)timer->data;
>     redisAsyncHandleTimeout(e->context);
> }
> static void redisLibevSetTimeout(void *privdata, struct timeval tv) {
>     redisLibevEvents *e = (redisLibevEvents*)privdata;
>     struct ev_loop *loop = e->loop;
>     ((void)loop);
>     if (!ev_is_active(&e->timer)) {
>         ev_init(&e->timer, redisLibevTimeout);
>         e->timer.data = e;
>     }
>     e->timer.repeat = tv.tv_sec + tv.tv_usec / 1000000.00;
>     ev_timer_again(EV_A_ &e->timer);
60c83,85
<     e = (redisLibevEvents*)malloc(sizeof(*e));
-----
>     e = (redisLibevEvents*)hi_calloc(1, sizeof(*e));
>     if (e == NULL)
>         return REDIS_ERR;
64d88
<     e->reading = e->writing = 0;
71a96
>     ac->ev.scheduleTimer = redisLibevSetTimeout;
改变 libuv.h

```


3,6c3,4

```
<  if (status != 0) {
<      return;
<  }
<  if (events & UV_READABLE) {
____
>  int ev = (status ? p->events : events);
>  if (p->context != NULL && (ev & UV_READABLE)) {
```

9c7

```
<  if (events & UV_WRITABLE) {
____
>  if (p->context != NULL && (ev & UV_WRITABLE)) {
```

43c41

```
<  free(p);
____
```

```
>  hi_free(p);
```

47c45

```
<  uv_close((uv_handle_t*)&p->handle, on_close);
____
>  p->context = NULL;  uv_close((uv_handle_t*)&p->handle, on_close);
```

59,62c57,59

```
<  redisLibuvEvents* p = (redisLibuvEvents*)malloc(sizeof(*p));
<  if (!p) {
<      return REDIS_ERR;
<  }
____
```

```
>  redisLibuvEvents* p = (redisLibuvEvents*)hi_malloc(sizeof(*p));
>  if (p == NULL)
>      return REDIS_ERR;
```

64c61

```
<  if (uv_poll_init(loop, &p->handle, c->fd) != 0) {
____
>  if (uv_poll_init_socket(loop, &p->handle, c->fd) != 0) {
```

改变 macosx.h

11c11

```
<      free(redisRunLoop);
____
```

```
>      hi_free(redisRunLoop);
```

51,52c51,53

```
<  RedisRunLoop* redisRunLoop = (RedisRunLoop*) calloc(1, sizeof(RedisRunLoop));
<  if( !redisRunLoop ) return REDIS_ERR;
____
>  RedisRunLoop* redisRunLoop = (RedisRunLoop*) hi_calloc(1, sizeof(RedisRunLoop));
>  if (redisRunLoop == NULL)
```

```

>         return REDIS_ERR;
改变 sds.h
2d1
< Error:inline has no {
3a3,4
>         return 0;
> }
5d5
< Error:inline has no {
6a7,66
>         case SDS_TYPE_8: {
>             SDS_HDR_VAR(8,s);
>             return sh->alloc - sh->len;
>         }
>         case SDS_TYPE_16: {
>             SDS_HDR_VAR(16,s);
>             return sh->alloc - sh->len;
>         }
>         case SDS_TYPE_32: {
>             SDS_HDR_VAR(32,s);
>             return sh->alloc - sh->len;
>         }
>         case SDS_TYPE_64: {
>             SDS_HDR_VAR(64,s);
>             return sh->alloc - sh->len;
>         }
>     }
>     return 0;
> }
> static inline void sdssetlen(sds s, size_t newlen) {
>
>         break;
>     case SDS_TYPE_8:
>         SDS_HDR(8,s)->len = (uint8_t)newlen;
>         break;
>     case SDS_TYPE_16:
>         SDS_HDR(16,s)->len = (uint16_t)newlen;
>         break;
>     case SDS_TYPE_32:
>         SDS_HDR(32,s)->len = (uint32_t)newlen;
>         break;
>     case SDS_TYPE_64:
>         SDS_HDR(64,s)->len = (uint64_t)newlen;
>         break;

```

```

>     }
> }
> static inline void sdsinclen(sds s, size_t inc) {
>
>     break;
>     case SDS_TYPE_8:
>         SDS_HDR(8,s)->len += (uint8_t)inc;
>         break;
>     case SDS_TYPE_16:
>         SDS_HDR(16,s)->len += (uint16_t)inc;
>         break;
>     case SDS_TYPE_32:
>         SDS_HDR(32,s)->len += (uint32_t)inc;
>         break;
>     case SDS_TYPE_64:
>         SDS_HDR(64,s)->len += (uint64_t)inc;
>         break;
>     }
> }
> static inline size_t sdsalloc(const sds s) {
>
>     return 0;
> }
> static inline void sdssetalloc(sds s, size_t newlen) {
>
> }
10d69
< size_t sdslen(const sds s);
13d71
< size_t sdsavail(const sds s);
25,26c83,84
< void sdstrim(sds s, const char *cset);
< void sdsrange(sds s, int start, int end);
——
> sds sdstrim(sds s, const char *cset);
> int sdsrange(sds s, ssize_t start, ssize_t end);
38c96
< sds sdsjoin(char **argv, int argc, char *sep, size_t seplen);
——
> sds sdsjoin(char **argv, int argc, char *sep);
43a102,106
> void *sdsAllocPtr(sds s);
> void *sds_malloc(size_t size);
> void *sds_realloc(void *ptr, size_t size);

```

```
> void sds_free(void *ptr);
> int sdsTest(int argc, char *argv[]);
```

asio-devel

变动很大，略

libevhttp-devel

增加头文件：log.h

增加头文件：tree.h

hdf5-openmpi-devel

改变 H5Apublic.h

2,17c2,18

```
< H5_DLL hid_t H5Acreate2(hid_t loc_id, const char *attr_name, hid_t type_id,
<     hid_t space_id, hid_t acpl_id, hid_t aapl_id);
< H5_DLL hid_t H5Acreate_by_name(hid_t loc_id, const char *obj_name, const char *attr_name,
<     hid_t type_id, hid_t space_id, hid_t acpl_id, hid_t aapl_id, hid_t lapl_id);
< H5_DLL hid_t H5Aopen(hid_t obj_id, const char *attr_name, hid_t aapl_id);
< H5_DLL hid_t H5Aopen_by_name(hid_t loc_id, const char *obj_name,
<     const char *attr_name, hid_t aapl_id, hid_t lapl_id);
< H5_DLL hid_t H5Aopen_by_idx(hid_t loc_id, const char *obj_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t aapl_id,
<     hid_t lapl_id);
< H5_DLL herr_t H5Awrite(hid_t attr_id, hid_t type_id, const void *buf);
< H5_DLL herr_t H5Aread(hid_t attr_id, hid_t type_id, void *buf);
< H5_DLL herr_t H5Aclose(hid_t attr_id);
< H5_DLL hid_t H5Aget_space(hid_t attr_id);
< H5_DLL hid_t H5Aget_type(hid_t attr_id);
< H5_DLL hid_t H5Aget_create_plist(hid_t attr_id);
——
> H5_DLL herr_t H5Aclose(hid_t attr_id);
> H5_DLL hid_t H5Acreate2(hid_t loc_id, const char *attr_name, hid_t type_id, hid_t space_id, hid_t acpl_id,
>     hid_t aapl_id);
> H5_DLL hid_t H5Acreate_by_name(hid_t loc_id, const char *obj_name, const char *attr_name, hid_t type_id,
>     hid_t space_id, hid_t acpl_id, hid_t aapl_id, hid_t lapl_id);
> H5_DLL herr_t H5Adelete(hid_t loc_id, const char *attr_name);
> H5_DLL herr_t H5Adelete_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type, H5_iter_order_t
order,
>     hsize_t n, hid_t lapl_id);
> H5_DLL herr_t H5Adelete_by_name(hid_t loc_id, const char *obj_name, const char *attr_name, hid_t lapl_id);
> H5_DLL htri_t H5Aexists(hid_t obj_id, const char *attr_name);
```

```

> H5_DLL htri_t H5Aexists_by_name(hid_t obj_id, const char *obj_name, const char *attr_name, hid_t lapl_id);
> H5_DLL hid_t H5Aget_create_plist(hid_t attr_id);
> H5_DLL herr_t H5Aget_info(hid_t attr_id, H5A_info_t *ainfo );
> H5_DLL herr_t H5Aget_info_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, H5A_info_t *ainfo , hid_t lapl_id);
> H5_DLL herr_t H5Aget_info_by_name(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     H5A_info_t *ainfo , hid_t lapl_id);
19,21c20,23
< H5_DLL ssize_t H5Aget_name_by_idx(hid_t loc_id, const char *obj_name,
<
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<
<     char *name , size_t size, hid_t lapl_id);
____
> H5_DLL ssize_t H5Aget_name_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, char *name , size_t size,
>
>                                     hid_t lapl_id);
> H5_DLL hid_t H5Aget_space(hid_t attr_id);
23,51c25,45
< H5_DLL herr_t  H5Aget_info(hid_t attr_id, H5A_info_t *ainfo );
< H5_DLL herr_t  H5Aget_info_by_name(hid_t loc_id, const char *obj_name,
<
<     const char *attr_name, H5A_info_t *ainfo , hid_t lapl_id);
< H5_DLL herr_t  H5Aget_info_by_idx(hid_t loc_id, const char *obj_name,
<
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<
<     H5A_info_t *ainfo , hid_t lapl_id);
< H5_DLL herr_t  H5Arename(hid_t loc_id, const char *old_name, const char *new_name);
< H5_DLL herr_t  H5Arename_by_name(hid_t loc_id, const char *obj_name,
<
<     const char *old_attr_name, const char *new_attr_name, hid_t lapl_id);
< H5_DLL herr_t  H5Aiterate2(hid_t loc_id, H5_index_t idx_type,
<
<     H5_iter_order_t order, hsize_t *idx, H5A_operator2_t op, void *op_data);
< H5_DLL herr_t  H5Aiterate_by_name(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
<
<     H5_iter_order_t order, hsize_t *idx, H5A_operator2_t op, void *op_data,
<
<     hid_t lapd_id);
< H5_DLL herr_t  H5Adelete(hid_t loc_id, const char *name);
< H5_DLL herr_t  H5Adelete_by_name(hid_t loc_id, const char *obj_name,
<
<     const char *attr_name, hid_t lapl_id);
< H5_DLL herr_t  H5Adelete_by_idx(hid_t loc_id, const char *obj_name,
<
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t lapl_id);
< H5_DLL htri_t H5Aexists(hid_t obj_id, const char *attr_name);
< H5_DLL htri_t H5Aexists_by_name(hid_t obj_id, const char *obj_name,
<
<     const char *attr_name, hid_t lapl_id);
< H5_DLL hid_t  H5Acreate1(hid_t loc_id, const char *name, hid_t type_id,
<
<     hid_t space_id, hid_t acpl_id);
< H5_DLL hid_t  H5Aopen_name(hid_t loc_id, const char *name);
< H5_DLL hid_t  H5Aopen_idx(hid_t loc_id, unsigned idx);
< H5_DLL int     H5Aget_num_attrs(hid_t loc_id);

```

```

< H5_DLL herr_t H5Aiterate1(hid_t loc_id, unsigned *attr_num, H5A_operator1_t op,
<     void *op_data);
____
> H5_DLL hid_t H5Aget_type(hid_t attr_id);
> H5_DLL herr_t H5Aiterate2(hid_t loc_id, H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
>     H5A_operator2_t op, void *op_data);
> H5_DLL herr_t H5Aiterate_by_name(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t *idx, H5A_operator2_t op, void
*op_data,
>     hid_t lapl_id);
> H5_DLL hid_t H5Aopen(hid_t obj_id, const char *attr_name, hid_t aapl_id);
> H5_DLL hid_t H5Aopen_by_idx(hid_t loc_id, const char *obj_name, H5_index_t idx_type, H5_iter_order_t order,
>     hsize_t n, hid_t aapl_id, hid_t lapl_id);
> H5_DLL hid_t H5Aopen_by_name(hid_t loc_id, const char *obj_name, const char *attr_name, hid_t aapl_id,
>     hid_t lapl_id);
> H5_DLL herr_t H5Aread(hid_t attr_id, hid_t type_id, void *buf);
> H5_DLL herr_t H5Arename(hid_t loc_id, const char *old_name, const char *new_name);
> H5_DLL herr_t H5Awrite(hid_t attr_id, hid_t type_id, const void *buf);
> H5_DLL herr_t H5Arename_by_name(hid_t loc_id, const char *obj_name, const char *old_attr_name,
>     const char *new_attr_name, hid_t lapl_id);
> H5_DLL hid_t H5Acreate1(hid_t loc_id, const char *name, hid_t type_id, hid_t space_id, hid_t acpl_id);
> H5_DLL int H5Aget_num_attrs(hid_t loc_id);
> H5_DLL herr_t H5Aiterate1(hid_t loc_id, unsigned *idx, H5A_operator1_t op, void *op_data);
> H5_DLL hid_t H5Aopen_idx(hid_t loc_id, unsigned idx);
> H5_DLL hid_t H5Aopen_name(hid_t loc_id, const char *name);

```

52a47

>

改变 H5DOpublic.h

2,12c2,7

```

< H5_HLDLL herr_t H5DOWrite_chunk(hid_t dset_id,
<     hid_t dxpl_id,
<     uint32_t filters,
<     const hsize_t *offset,
<     size_t data_size,
<     const void *buf);
< H5_HLDLL herr_t H5DORead_chunk(hid_t dset_id,
<     hid_t dxpl_id,
<     const hsize_t *offset,
<     uint32_t *filters,
<     void *buf);
____
> H5_HLDLL herr_t H5DOappend(hid_t dset_id, hid_t dxpl_id, unsigned axis, size_t extension, hid_t memtype,
>     const void *buf);
> H5_HLDLL herr_t H5DOWrite_chunk(hid_t dset_id, hid_t dxpl_id, uint32_t filters, const hsize_t *offset,

```

```

>                                     size_t data_size, const void *buf);
> H5_HLDLL herr_t H5Doread_chunk(hid_t dset_id, hid_t dxpl_id, const hsize_t *offset, uint32_t *filters ,
>
>                                     void *buf );
改变 H5Dpublic.h
2,7c2,5
< H5_DLL hid_t H5Dcreate2(hid_t loc_id, const char *name, hid_t type_id,
<     hid_t space_id, hid_t lcpl_id, hid_t dcpl_id, hid_t dapl_id);
< H5_DLL hid_t H5Dcreate_anon(hid_t file_id, hid_t type_id, hid_t space_id,
<     hid_t plist_id, hid_t dapl_id);
< H5_DLL hid_t H5Dopen2(hid_t file_id, const char *name, hid_t dapl_id);
< H5_DLL herr_t H5Dclose(hid_t dset_id);
——
> H5_DLL hid_t H5Dcreate2(hid_t loc_id, const char *name, hid_t type_id, hid_t space_id, hid_t lcpl_id,
>
>                                     hid_t dcpl_id, hid_t dapl_id);
> H5_DLL hid_t H5Dcreate_anon(hid_t loc_id, hid_t type_id, hid_t space_id, hid_t dcpl_id, hid_t dapl_id);
> H5_DLL hid_t H5Dopen2(hid_t loc_id, const char *name, hid_t dapl_id);
14a13,17
> H5_DLL herr_t H5Dget_num_chunks(hid_t dset_id, hid_t fspace_id, hsize_t *nchunks);
> H5_DLL herr_t H5Dget_chunk_info_by_coord(hid_t dset_id, const hsize_t *offset, unsigned *filter_mask,
>
>                                     haddr_t *addr, hsize_t *size);
> H5_DLL herr_t H5Dget_chunk_info(hid_t dset_id, hid_t fspace_id, hsize_t chk_idx, hsize_t *offset,
>
>                                     unsigned *filter_mask, haddr_t *addr, hsize_t *size);
16,25c19,29
< H5_DLL herr_t H5Dread(hid_t dset_id, hid_t mem_type_id, hid_t mem_space_id,
<
<     hid_t file_space_id, hid_t plist_id, void *buf);
< H5_DLL herr_t H5Dwrite(hid_t dset_id, hid_t mem_type_id, hid_t mem_space_id,
<
<     hid_t file_space_id, hid_t plist_id, const void *buf);
< H5_DLL herr_t H5Diterate(void *buf, hid_t type_id, hid_t space_id,
<
<     H5D_operator_t op, void *operator_data);
< H5_DLL herr_t H5Dvlen_reclaim(hid_t type_id, hid_t space_id, hid_t plist_id, void *buf);
< H5_DLL herr_t H5Dvlen_get_buf_size(hid_t dataset_id, hid_t type_id, hid_t space_id, hsize_t *size);
< H5_DLL herr_t H5Dfill(const void *fill, hid_t fill_type, void *buf,
<
<     hid_t buf_type, hid_t space);
——
> H5_DLL herr_t H5Dread(hid_t dset_id, hid_t mem_type_id, hid_t mem_space_id, hid_t file_space_id,
>
>                                     hid_t dxpl_id, void *buf );
> H5_DLL herr_t H5Dwrite(hid_t dset_id, hid_t mem_type_id, hid_t mem_space_id, hid_t file_space_id,
>
>                                     hid_t dxpl_id, const void *buf);
> H5_DLL herr_t H5Dwrite_chunk(hid_t dset_id, hid_t dxpl_id, uint32_t filters, const hsize_t *offset,
>
>                                     size_t data_size, const void *buf);
> H5_DLL herr_t H5Dread_chunk(hid_t dset_id, hid_t dxpl_id, const hsize_t *offset, uint32_t *filters,
>
>                                     void *buf);
> H5_DLL herr_t H5Diterate(void *buf, hid_t type_id, hid_t space_id, H5D_operator_t op, void *operator_data);
> H5_DLL herr_t H5Dvlen_get_buf_size(hid_t dset_id, hid_t type_id, hid_t space_id, hsize_t *size);

```

> H5_DLL herr_t H5Dfill(const void *fill, hid_t fill_type_id, void *buf, hid_t buf_type_id, hid_t space_id);
27,30c31,37

< H5_DLL herr_t H5Dscatter(H5D_scatter_func_t op, void *op_data, hid_t type_id,
< hid_t dst_space_id, void *dst_buf);

< H5_DLL herr_t H5Dgather(hid_t src_space_id, const void *src_buf, hid_t type_id,
< size_t dst_buf_size, void *dst_buf, H5D_gather_func_t op, void *op_data);

> H5_DLL herr_t H5Dflush(hid_t dset_id);

> H5_DLL herr_t H5Drefresh(hid_t dset_id);

> H5_DLL herr_t H5Dscatter(H5D_scatter_func_t op, void *op_data, hid_t type_id, hid_t dst_space_id,
> void *dst_buf);

> H5_DLL herr_t H5Dgather(hid_t src_space_id, const void *src_buf, hid_t type_id, size_t dst_buf_size,
> void *dst_buf, H5D_gather_func_t op, void *op_data);

> H5_DLL herr_t H5Dclose(hid_t dset_id);

32,34c39,42

< H5_DLL hid_t H5Dcreate1(hid_t file_id, const char *name, hid_t type_id,

< hid_t space_id, hid_t dcpl_id);

< H5_DLL hid_t H5Dopen1(hid_t file_id, const char *name);

> H5_DLL herr_t H5Dformat_convert(hid_t dset_id);

> H5_DLL herr_t H5Dget_chunk_index_type(hid_t did, H5D_chunk_index_t *idx_type);

> H5_DLL hid_t H5Dcreate1(hid_t loc_id, const char *name, hid_t type_id, hid_t space_id, hid_t dcpl_id);

> H5_DLL hid_t H5Dopen1(hid_t loc_id, const char *name);

35a44

> H5_DLL herr_t H5Dvlen_reclaim(hid_t type_id, hid_t space_id, hid_t dxpl_id, void *buf);

改变 H5DSpublic.h

2,30c2,12

< H5_HLDLL herr_t H5DSattach_scale(hid_t did,

< hid_t dsid,

< unsigned int idx);

< H5_HLDLL herr_t H5DSdetach_scale(hid_t did,

< hid_t dsid,

< unsigned int idx);

< H5_HLDLL herr_t H5DSset_scale(hid_t dsid,

< const char *dimname);

< H5_HLDLL int H5DSget_num_scales(hid_t did,

< unsigned int dim);

< H5_HLDLL herr_t H5DSset_label(hid_t did,

< unsigned int idx,

< const char *label);

< H5_HLDLL ssize_t H5DSget_label(hid_t did,

< unsigned int idx,

< char *label,

< size_t size);


```

< H5_HLDLL ssize_t H5DSget_scale_name( hid_t did,
<                                     char *name,
<                                     size_t size);
< H5_HLDLL htri_t H5DSis_scale( hid_t did);
< H5_HLDLL herr_t H5DSiterate_scales( hid_t did,
<                                     unsigned int dim,
<                                     int *idx,
<                                     H5DS_iterate_t visitor,
<                                     void *visitor_data);
< H5_HLDLL htri_t H5DSis_attached( hid_t did,
<                                     hid_t dsid,
<                                     unsigned int idx);
____
> H5_HLDLL herr_t H5DSattach_scale(hid_t did, hid_t dsid, unsigned int idx);
> H5_HLDLL herr_t H5DSdetach_scale(hid_t did, hid_t dsid, unsigned int idx);
> H5_HLDLL herr_t H5DSset_scale(hid_t dsid, const char *dimname);
> H5_HLDLL int H5DSget_num_scales(hid_t did, unsigned int dim);
> H5_HLDLL herr_t H5DSset_label(hid_t did, unsigned int idx, const char *label);
> H5_HLDLL ssize_t H5DSget_label(hid_t did, unsigned int idx, char *label, size_t size);
> H5_HLDLL ssize_t H5DSget_scale_name(hid_t did, char *name, size_t size);
> H5_HLDLL htri_t H5DSis_scale(hid_t did);
> H5_HLDLL herr_t H5DSiterate_scales(hid_t did, unsigned int dim, int *idx, H5DS_iterate_t visitor,
>                                     void *visitor_data);
> H5_HLDLL htri_t H5DSis_attached(hid_t did, hid_t dsid, unsigned int idx);
改变 H5Epubgen.h
2d1
< H5_DLLVAR hid_t H5E_DATASET_g;
4d2
< H5_DLLVAR hid_t H5E_STORAGE_g;
5a4
> H5_DLLVAR hid_t H5E_VOL_g;
15,17d13
< H5_DLLVAR hid_t H5E_PLIST_g;
< H5_DLLVAR hid_t H5E_LINK_g;
< H5_DLLVAR hid_t H5E_DATATYPE_g;
18a15
> H5_DLLVAR hid_t H5E_FARRAY_g;
20,21c17
< H5_DLLVAR hid_t H5E_OHDR_g;
< H5_DLLVAR hid_t H5E_ATOM_g;
____
> H5_DLLVAR hid_t H5E_MAP_g;
23d18
< H5_DLLVAR hid_t H5E_NONE_MAJOR_g;

```

25d19

< H5_DLLVAR hid_t H5E_SLIST_g;

27a22,33

> H5_DLLVAR hid_t H5E_LIB_g;

> H5_DLLVAR hid_t H5E_PAGEBUF_g;

> H5_DLLVAR hid_t H5E_FSPACE_g;

> H5_DLLVAR hid_t H5E_DATASET_g;

> H5_DLLVAR hid_t H5E_STORAGE_g;

> H5_DLLVAR hid_t H5E_LINK_g;

> H5_DLLVAR hid_t H5E_PLIST_g;

> H5_DLLVAR hid_t H5E_DATATYPE_g;

> H5_DLLVAR hid_t H5E_OHDR_g;

> H5_DLLVAR hid_t H5E_ATOM_g;

> H5_DLLVAR hid_t H5E_NONE_MAJOR_g;

> H5_DLLVAR hid_t H5E_SLIST_g;

29c35,36

< H5_DLLVAR hid_t H5E_ERROR_g;

> H5_DLLVAR hid_t H5E_CONTEXT_g;

> H5_DLLVAR hid_t H5E_EARRAY_g;

31c38

< H5_DLLVAR hid_t H5E_FSPACE_g;

> H5_DLLVAR hid_t H5E_ERROR_g;

95a103,105

> H5_DLLVAR hid_t H5E_CANTDELETEFILE_g;

> H5_DLLVAR hid_t H5E_CANTLOCKFILE_g;

> H5_DLLVAR hid_t H5E_CANTUNLOCKFILE_g;

102a113

> H5_DLLVAR hid_t H5E_CANTUNSERIALIZE_g;

103a115

> H5_DLLVAR hid_t H5E_CANTTAG_g;

113a126,128

> H5_DLLVAR hid_t H5E_CANTMARKCLEAN_g;

> H5_DLLVAR hid_t H5E_CANTMARKUNSERIALIZED_g;

> H5_DLLVAR hid_t H5E_CANTMARKSERIALIZED_g;

114a130

> H5_DLLVAR hid_t H5E_CANTCLEAN_g;

116a133,138

> H5_DLLVAR hid_t H5E_CANTDEPEND_g;

> H5_DLLVAR hid_t H5E_CANTUNDEPEND_g;

> H5_DLLVAR hid_t H5E_CANTNOTIFY_g;

> H5_DLLVAR hid_t H5E_LOGGING_g;

> H5_DLLVAR hid_t H5E_CANTCORK_g;

```

> H5_DLLVAR hid_t H5E_CANTUNCORK_g;
124a147,148
> H5_DLLVAR hid_t H5E_CANTGATHER_g;
> H5_DLLVAR hid_t H5E_NO_INDEPENDENT_g;
130a155,156
> H5_DLLVAR hid_t H5E_INCONSISTENTSTATE_g;
> H5_DLLVAR hid_t H5E_CANTAPPEND_g;
改变 H5Epublic.h
3,4c3
< H5_DLL hid_t H5Eregister_class(const char *cls_name, const char *lib_name,
<     const char *version);
____
> H5_DLL hid_t H5Eregister_class(const char *cls_name, const char *lib_name, const char *version);
7,9c6,8
< H5_DLL hid_t H5Ecreate_msg(hid_t cls, H5E_type_t msg_type, const char *msg);
< H5_DLL hid_t H5Ecreate_stack(void);
< H5_DLL hid_t H5Eget_current_stack(void);
____
> H5_DLL hid_t H5Ecreate_msg(hid_t cls, H5E_type_t msg_type, const char *msg);
> H5_DLL hid_t H5Ecreate_stack(void);
> H5_DLL hid_t H5Eget_current_stack(void);
13,14c12,13
< H5_DLL herr_t H5Epush2(hid_t err_stack, const char *file, const char *func, unsigned line,
<     hid_t cls_id, hid_t maj_id, hid_t min_id, const char *msg, ...);
____
> H5_DLL herr_t H5Epush2(hid_t err_stack, const char *file, const char *func, unsigned line, hid_t cls_id,
>     hid_t maj_id, hid_t min_id, const char *msg, ...);
17,18c16
< H5_DLL herr_t H5Ewalk2(hid_t err_stack, H5E_direction_t direction, H5E_walk2_t func,
<     void *client_data);
____
> H5_DLL herr_t H5Ewalk2(hid_t err_stack, H5E_direction_t direction, H5E_walk2_t func, void *client_data);
23,24c21
< H5_DLL ssize_t H5Eget_msg(hid_t msg_id, H5E_type_t *type, char *msg,
<     size_t size);
____
> H5_DLL ssize_t H5Eget_msg(hid_t msg_id, H5E_type_t *type, char *msg, size_t size);
28,29c25,26
< H5_DLL herr_t H5Epush1(const char *file, const char *func, unsigned line,
<     H5E_major_t maj, H5E_minor_t min, const char *str);
____
> H5_DLL herr_t H5Epush1(const char *file, const char *func, unsigned line, H5E_major_t maj, H5E_minor_t min,
>     const char *str);
32,33c29

```

```

< H5_DLL herr_t H5Ewalk1(H5E_direction_t direction, H5E_walk1_t func,
<     void *client_data);
——
> H5_DLL herr_t H5Ewalk1(H5E_direction_t direction, H5E_walk1_t func, void *client_data);
36a33
>
增加头文件: H5ESpublic.h
改变 H5f90i_gen.h
1d0
<
改变 H5FDcore.h
3,7c3,4
< H5_DLL void H5FD_core_term(void);
< H5_DLL herr_t H5Pset_fapl_core(hid_t fapl_id, size_t increment,
<     hbool_t backing_store);
< H5_DLL herr_t H5Pget_fapl_core(hid_t fapl_id, size_t *increment,
<     hbool_t *backing_store);
——
> H5_DLL herr_t H5Pset_fapl_core(hid_t fapl_id, size_t increment, hbool_t backing_store);
> H5_DLL herr_t H5Pget_fapl_core(hid_t fapl_id, size_t *increment, hbool_t *backing_store);
改变 H5FDdirect.h
3,7c3,5
< H5_DLL void H5FD_direct_term(void);
< H5_DLL herr_t H5Pset_fapl_direct(hid_t fapl_id, size_t alignment, size_t block_size,
<     size_t cbuf_size);
< H5_DLL herr_t H5Pget_fapl_direct(hid_t fapl_id, size_t *boundary,
<     size_t *block_size, size_t *cbuf_size);
——
> H5_DLL herr_t H5Pset_fapl_direct(hid_t fapl_id, size_t alignment, size_t block_size, size_t cbuf_size);
> H5_DLL herr_t H5Pget_fapl_direct(hid_t fapl_id, size_t *boundary, size_t *block_size,
>     size_t *cbuf_size);
改变 H5FDfamily.h
3,7c3,4
< H5_DLL void H5FD_family_term(void);
< H5_DLL herr_t H5Pset_fapl_family(hid_t fapl_id, hsize_t memb_size,
<     hid_t memb_fapl_id);
< H5_DLL herr_t H5Pget_fapl_family(hid_t fapl_id, hsize_t *memb_size,
<     hid_t *memb_fapl_id);
——
> H5_DLL herr_t H5Pset_fapl_family(hid_t fapl_id, hsize_t memb_size, hid_t memb_fapl_id);
> H5_DLL herr_t H5Pget_fapl_family(hid_t fapl_id, hsize_t *memb_size, hid_t *memb_fapl_id);
增加头文件: H5FDhdfs.h
改变 H5FDlog.h
3d2

```

< H5_DLL void H5FD_log_term(void);

增加头文件: H5FDmirror.h

改变 H5FDmpi.h

0a1

>

改变 H5FDmpio.h

4d3

< H5_DLL void H5FD_mpio_term(void);

6,7c5

< H5_DLL herr_t H5Pget_fapl_mpio(hid_t fapl_id, MPI_Comm *comm,

< MPI_Info *info);

> H5_DLL herr_t H5Pget_fapl_mpio(hid_t fapl_id, MPI_Comm *comm, MPI_Info *info);

9c7

< H5_DLL herr_t H5Pget_dxpl_mpio(hid_t dxpl_id, H5FD_mpio_xfer_t *xfer_mode);

> H5_DLL herr_t H5Pget_dxpl_mpio(hid_t dxpl_id, H5FD_mpio_xfer_t *xfer_mode);

改变 H5FDmulti.h

3,12c3,8

< H5_DLL void H5FD_multi_term(void);

< H5_DLL herr_t H5Pset_fapl_multi(hid_t fapl_id, const H5FD_mem_t *memb_map,

< const hid_t *memb_fapl, const char * const *memb_name,

< const haddr_t *memb_addr, hbool_t relax);

< H5_DLL herr_t H5Pget_fapl_multi(hid_t fapl_id, H5FD_mem_t *memb_map,

< hid_t *memb_fapl, char **memb_name,

< haddr_t *memb_addr, hbool_t *relax);

< H5_DLL herr_t H5Pset_fapl_split(hid_t fapl, const char *meta_ext,

< hid_t meta_plist_id, const char *raw_ext,

< hid_t raw_plist_id);

> H5_DLL herr_t H5Pset_fapl_multi(hid_t fapl_id, const H5FD_mem_t *memb_map, const hid_t *memb_fapl,

> const char *const *memb_name, const haddr_t *memb_addr, hbool_t relax);

> H5_DLL herr_t H5Pget_fapl_multi(hid_t fapl_id, H5FD_mem_t *memb_map, hid_t *memb_fapl,

> char **memb_name, haddr_t *memb_addr, hbool_t *relax);

> H5_DLL herr_t H5Pset_fapl_split(hid_t fapl, const char *meta_ext, hid_t meta_plist_id, const char *raw_ext,

> hid_t raw_plist_id);

改变 H5FDpublic.h

2c2

< H5_DLL hid_t H5FDregister(const H5FD_class_t *cls);

> H5_DLL hid_t H5FDregister(const H5FD_class_t *cls);

4,8c4,7

< H5_DLL H5FD_t *H5FDopen(const char *name, unsigned flags, hid_t fapl_id,

```

<                                haddr_t maxaddr);
< H5_DLL herr_t H5FDclose(H5FD_t *file);
< H5_DLL int H5FDcmp(const H5FD_t *f1, const H5FD_t *f2);
< H5_DLL int H5FDquery(const H5FD_t *f, unsigned long *flags);
____
> H5_DLL H5FD_t *H5FDopen(const char *name, unsigned flags, hid_t fapl_id, haddr_t maxaddr);
> H5_DLL herr_t H5FDclose(H5FD_t *file);
> H5_DLL int H5FDcmp(const H5FD_t *f1, const H5FD_t *f2);
> H5_DLL int H5FDquery(const H5FD_t *f, unsigned long *flags);
10,11c9
< H5_DLL herr_t H5FDfree(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id,
<                                haddr_t addr, hsize_t size);
____
> H5_DLL herr_t H5FDfree(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id, haddr_t addr, hsize_t size);
13,21c11,22
< H5_DLL herr_t H5FDset_eoa(H5FD_t *file, H5FD_mem_t type, haddr_t eoa);
< H5_DLL haddr_t H5FDget_eof(H5FD_t *file);
< H5_DLL herr_t H5FDget_vfd_handle(H5FD_t *file, hid_t fapl, void**file_handle);
< H5_DLL herr_t H5FDread(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id,
<                                haddr_t addr, size_t size, void *buf);
< H5_DLL herr_t H5FDwrite(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id,
<                                haddr_t addr, size_t size, const void *buf);
< H5_DLL herr_t H5FDflush(H5FD_t *file, hid_t dxpl_id, unsigned closing);
< H5_DLL herr_t H5FDtruncate(H5FD_t *file, hid_t dxpl_id, hbool_t closing);
____
> H5_DLL herr_t H5FDset_eoa(H5FD_t *file, H5FD_mem_t type, haddr_t eoa);
> H5_DLL haddr_t H5FDget_eof(H5FD_t *file, H5FD_mem_t type);
> H5_DLL herr_t H5FDget_vfd_handle(H5FD_t *file, hid_t fapl, void **file_handle);
> H5_DLL herr_t H5FDread(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id, haddr_t addr, size_t size,
>                                void *buf);
> H5_DLL herr_t H5FDwrite(H5FD_t *file, H5FD_mem_t type, hid_t dxpl_id, haddr_t addr, size_t size,
>                                const void *buf);
> H5_DLL herr_t H5FDflush(H5FD_t *file, hid_t dxpl_id, hbool_t closing);
> H5_DLL herr_t H5FDtruncate(H5FD_t *file, hid_t dxpl_id, hbool_t closing);
> H5_DLL herr_t H5FDlock(H5FD_t *file, hbool_t rw);
> H5_DLL herr_t H5FDunlock(H5FD_t *file);
> H5_DLL herr_t H5FDdriver_query(hid_t driver_id, unsigned long *flags);
增加头文件: H5FDros3.h
改变 H5FDsec2.h
2,3c2
< H5_DLL hid_t H5FD_sec2_init(void);
< H5_DLL void H5FD_sec2_term(void);
____
> H5_DLL hid_t H5FD_sec2_init(void);

```

增加头文件: H5FDsplitter.h

改变 H5FDstdio.h

3d2

< H5_DLL void H5FD_stdio_term(void);

增加头文件: H5FDwindows.h

改变 H5Fpublic.h

2,7c2,5

< H5_DLL htri_t H5Fis_hdf5(const char *filename);

< H5_DLL hid_t H5Fcreate(const char *filename, unsigned flags,
< hid_t create_plist, hid_t access_plist);

< H5_DLL hid_t H5Fopen(const char *filename, unsigned flags,
< hid_t access_plist);

< H5_DLL hid_t H5Freopen(hid_t file_id);

> H5_DLL htri_t H5Fis_accessible(const char *container_name, hid_t fapl_id);

> H5_DLL hid_t H5Fcreate(const char *filename, unsigned flags, hid_t fcpl_id, hid_t fapl_id);

> H5_DLL hid_t H5Fopen(const char *filename, unsigned flags, hid_t fapl_id);

> H5_DLL hid_t H5Freopen(hid_t file_id);

10,12c8,12

< H5_DLL hid_t H5Fget_create_plist(hid_t file_id);

< H5_DLL hid_t H5Fget_access_plist(hid_t file_id);

< H5_DLL herr_t H5Fget_intent(hid_t file_id, unsigned * intent);

> H5_DLL herr_t H5Fdelete(const char *filename, hid_t fapl_id);

> H5_DLL hid_t H5Fget_create_plist(hid_t file_id);

> H5_DLL hid_t H5Fget_access_plist(hid_t file_id);

> H5_DLL herr_t H5Fget_intent(hid_t file_id, unsigned *intent);

> H5_DLL herr_t H5Fget_fileno(hid_t file_id, unsigned long *fileno);

20,30c20,27

< H5_DLL ssize_t H5Fget_file_image(hid_t file_id, void * buf_ptr, size_t buf_len);

< H5_DLL herr_t H5Fget_mdc_config(hid_t file_id,

< H5AC_cache_config_t * config_ptr);

< H5_DLL herr_t H5Fset_mdc_config(hid_t file_id,

< H5AC_cache_config_t * config_ptr);

< H5_DLL herr_t H5Fget_mdc_hit_rate(hid_t file_id, double * hit_rate_ptr);

< H5_DLL herr_t H5Fget_mdc_size(hid_t file_id,

< size_t * max_size_ptr,

< size_t * min_clean_size_ptr,

< size_t * cur_size_ptr,

< int * cur_num_entries_ptr);

> H5_DLL herr_t H5Fget_eoa(hid_t file_id, haddr_t *eoa);

> H5_DLL herr_t H5Fincrement_filesize(hid_t file_id, hsize_t increment);

> H5_DLL ssize_t H5Fget_file_image(hid_t file_id, void *buf_ptr, size_t buf_len);

```

> H5_DLL herr_t H5Fget_mdc_config(hid_t file_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Fset_mdc_config(hid_t file_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Fget_mdc_hit_rate(hid_t file_id, double *hit_rate_ptr);
> H5_DLL herr_t H5Fget_mdc_size(hid_t file_id, size_t *max_size_ptr, size_t *min_clean_size_ptr,
>                               size_t *cur_size_ptr, int *cur_num_entries_ptr);
33c30,34
< H5_DLL herr_t H5Fget_info(hid_t obj_id, H5F_info_t *bh_info);
----
> H5_DLL herr_t H5Fget_info2(hid_t obj_id, H5F_info2_t *file_info);
> H5_DLL herr_t H5Fget_metadata_read_retry_info(hid_t file_id, H5F_retry_info_t *info);
> H5_DLL herr_t H5Fstart_swmr_write(hid_t file_id);
> H5_DLL ssize_t H5Fget_free_sections(hid_t file_id, H5F_mem_t type, size_t nsects,
>                                     H5F_sect_info_t *sect_info );
34a36,46
> H5_DLL herr_t H5Fset_libver_bounds(hid_t file_id, H5F_libver_t low, H5F_libver_t high);
> H5_DLL herr_t H5Fstart_mdc_logging(hid_t file_id);
> H5_DLL herr_t H5Fstop_mdc_logging(hid_t file_id);
> H5_DLL herr_t H5Fget_mdc_logging_status(hid_t file_id, hbool_t *is_enabled, hbool_t *is_currently_logging);
> H5_DLL herr_t H5Fformat_convert(hid_t fid);
> H5_DLL herr_t H5Freset_page_buffering_stats(hid_t file_id);
> H5_DLL herr_t H5Fget_page_buffering_stats(hid_t file_id, unsigned accesses[2], unsigned hits[2],
>                                           unsigned misses[2], unsigned evictions[2], unsigned
bypasses[2]);
> H5_DLL herr_t H5Fget_mdc_image_info(hid_t file_id, haddr_t *image_addr, hsize_t *image_size);
> H5_DLL herr_t H5Fget_dset_no_attrs_hint(hid_t file_id, hbool_t *minimize);
> H5_DLL herr_t H5Fset_dset_no_attrs_hint(hid_t file_id, hbool_t minimize);
36a49,51
> H5_DLL herr_t H5Fget_info1(hid_t obj_id, H5F_info1_t *file_info);
> H5_DLL herr_t H5Fset_latest_format(hid_t file_id, hbool_t latest_format);
> H5_DLL htri_t H5Fis_hdf5(const char *file_name);
37a53
>
改变 H5Gpublic.h
2,3c2
< H5_DLL hid_t H5Gcreate2(hid_t loc_id, const char *name, hid_t lcpl_id,
<   hid_t gcpl_id, hid_t gapl_id);
----
> H5_DLL hid_t H5Gcreate2(hid_t loc_id, const char *name, hid_t lcpl_id, hid_t gcpl_id, hid_t gapl_id);
8,12c7,11
< H5_DLL herr_t H5Gget_info_by_name(hid_t loc_id, const char *name, H5G_info_t *ginfo,
<   hid_t lapl_id);
< H5_DLL herr_t H5Gget_info_by_idx(hid_t loc_id, const char *group_name,
<   H5_index_t idx_type, H5_iter_order_t order, hsize_t n, H5G_info_t *ginfo,
<   hid_t lapl_id);

```



```

-----
> H5_DLL herr_t H5Gget_info_by_name(hid_t loc_id, const char *name, H5G_info_t *ginfo, hid_t lapl_id);
> H5_DLL herr_t H5Gget_info_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, H5G_info_t *ginfo, hid_t lapl_id);
> H5_DLL herr_t H5Gflush(hid_t group_id);
> H5_DLL herr_t H5Grefresh(hid_t group_id);
16,23c15,19
< H5_DLL herr_t H5Glink(hid_t cur_loc_id, H5G_link_t type, const char *cur_name,
<
<     const char *new_name);
< H5_DLL herr_t H5Glink2(hid_t cur_loc_id, const char *cur_name, H5G_link_t type,
<
<     hid_t new_loc_id, const char *new_name);
< H5_DLL herr_t H5Gmove(hid_t src_loc_id, const char *src_name,
<
<     const char *dst_name);
< H5_DLL herr_t H5Gmove2(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id,
<
<     const char *dst_name);
-----
> H5_DLL herr_t H5Glink(hid_t cur_loc_id, H5G_link_t type, const char *cur_name, const char *new_name);
> H5_DLL herr_t H5Glink2(hid_t cur_loc_id, const char *cur_name, H5G_link_t type, hid_t new_loc_id,
>
>                                     const char *new_name);
> H5_DLL herr_t H5Gmove(hid_t src_loc_id, const char *src_name, const char *dst_name);
> H5_DLL herr_t H5Gmove2(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id, const char *dst_name);
25,26c21
< H5_DLL herr_t H5Gget_linkval(hid_t loc_id, const char *name, size_t size,
<
<     char *buf);
-----
> H5_DLL herr_t H5Gget_linkval(hid_t loc_id, const char *name, size_t size, char *buf);
28,31c23,24
< H5_DLL int H5Gget_comment(hid_t loc_id, const char *name, size_t bufsize,
<
<     char *buf);
< H5_DLL herr_t H5Giterate(hid_t loc_id, const char *name, int *idx,
<
<     H5G_iterate_t op, void *op_data);
-----
> H5_DLL int H5Gget_comment(hid_t loc_id, const char *name, size_t bufsize, char *buf);
> H5_DLL herr_t H5Giterate(hid_t loc_id, const char *name, int *idx, H5G_iterate_t op, void *op_data);
33,36c26,28
< H5_DLL herr_t H5Gget_objinfo(hid_t loc_id, const char *name,
<
<     hbool_t follow_link, H5G_stat_t *statbuf);
< H5_DLL ssize_t H5Gget_objname_by_idx(hid_t loc_id, hsize_t idx, char* name,
<
<     size_t size);
-----
> H5_DLL herr_t H5Gget_objinfo(hid_t loc_id, const char *name, hbool_t follow_link,
>
>                                     H5G_stat_t *statbuf);
> H5_DLL ssize_t H5Gget_objname_by_idx(hid_t loc_id, hsize_t idx, char *name, size_t size);
38a31

```

>

改变 H5IMpublic.h

2,47c2,18

```
< H5_HLDLL herr_t H5IMmake_image_8bit( hid_t loc_id,
<                                     const char *dset_name,
<                                     hsize_t width,
<                                     hsize_t height,
<                                     const unsigned char *buffer );
< H5_HLDLL herr_t H5IMmake_image_24bit( hid_t loc_id,
<                                     const char *dset_name,
<                                     hsize_t width,
<                                     hsize_t height,
<                                     const char *interlace,
<                                     const unsigned char *buffer );
< H5_HLDLL herr_t H5IMget_image_info( hid_t loc_id,
<                                     const char *dset_name,
<                                     hsize_t *width,
<                                     hsize_t *height,
<                                     hsize_t *planes,
<                                     char *interlace,
<                                     hsize_t *npals );
< H5_HLDLL herr_t H5IMread_image( hid_t loc_id,
<                                     const char *dset_name,
<                                     unsigned char *buffer );
< H5_HLDLL herr_t H5IMmake_palette( hid_t loc_id,
<                                     const char *pal_name,
<                                     const hsize_t *pal_dims,
<                                     const unsigned char *pal_data );
< H5_HLDLL herr_t H5IMlink_palette( hid_t loc_id,
<                                     const char *image_name,
<                                     const char *pal_name );
< H5_HLDLL herr_t H5IMunlink_palette( hid_t loc_id,
<                                     const char *image_name,
<                                     const char *pal_name );
< H5_HLDLL herr_t H5IMget_npalettes( hid_t loc_id,
<                                     const char *image_name,
<                                     hsize_t *npals );
< H5_HLDLL herr_t H5IMget_palette_info( hid_t loc_id,
<                                     const char *image_name,
<                                     int pal_number,
<                                     hsize_t *pal_dims );
< H5_HLDLL herr_t H5IMget_palette( hid_t loc_id,
<                                     const char *image_name,
<                                     int pal_number,
```

```

< unsigned char *pal_data );
< H5_HLDLL herr_t H5IMis_image( hid_t loc_id,
< const char *dset_name );
< H5_HLDLL herr_t H5IMis_palette( hid_t loc_id,
< const char *dset_name );
——
> H5_HLDLL herr_t H5IMmake_image_8bit(hid_t loc_id, const char *dset_name, hsize_t width, hsize_t height,
> const unsigned char *buffer);
> H5_HLDLL herr_t H5IMmake_image_24bit(hid_t loc_id, const char *dset_name, hsize_t width, hsize_t height,
> const char *interlace, const unsigned char *buffer);
> H5_HLDLL herr_t H5IMget_image_info(hid_t loc_id, const char *dset_name, hsize_t *width, hsize_t *height,
> hsize_t *planes, char *interlace, hsize_t *npals);
> H5_HLDLL herr_t H5IMread_image(hid_t loc_id, const char *dset_name, unsigned char *buffer);
> H5_HLDLL herr_t H5IMmake_palette(hid_t loc_id, const char *pal_name, const hsize_t *pal_dims,
> const unsigned char *pal_data);
> H5_HLDLL herr_t H5IMlink_palette(hid_t loc_id, const char *image_name, const char *pal_name);
> H5_HLDLL herr_t H5IMunlink_palette(hid_t loc_id, const char *image_name, const char *pal_name);
> H5_HLDLL herr_t H5IMget_npalettes(hid_t loc_id, const char *image_name, hsize_t *npals);
> H5_HLDLL herr_t H5IMget_palette_info(hid_t loc_id, const char *image_name, int pal_number, hsize_t
*pal_dims);
> H5_HLDLL herr_t H5IMget_palette(hid_t loc_id, const char *image_name, int pal_number,
> unsigned char *pal_data);
> H5_HLDLL herr_t H5IMis_image(hid_t loc_id, const char *dset_name);
> H5_HLDLL herr_t H5IMis_palette(hid_t loc_id, const char *dset_name);

```

改变 H5Ipublic.h

3,4c3,4

```

< H5_DLL void *H5Iobject_verify(hid_t id, H5I_type_t id_type);
< H5_DLL void *H5Iremove_verify(hid_t id, H5I_type_t id_type);
——

```

```

> H5_DLL void *H5Iobject_verify(hid_t id, H5I_type_t type);
> H5_DLL void *H5Iremove_verify(hid_t id, H5I_type_t type);

```

7c7

```

< H5_DLL ssize_t H5Iget_name(hid_t id, char *name, size_t size);
——

```

```

> H5_DLL ssize_t H5Iget_name(hid_t id, char *name , size_t size);

```

17a18

```

> H5_DLL herr_t H5Iiterate(H5I_type_t type, H5I_iterate_func_t op, void *op_data);

```

21a23

>

增加头文件： H5LDpublic.h

改变 H5Lpublic.h

2,9c2,9

```

< H5_DLL herr_t H5Lmove(hid_t src_loc, const char *src_name, hid_t dst_loc,
< const char *dst_name, hid_t lcpl_id, hid_t lapl_id);

```

```

< H5_DLL herr_t H5Lcopy(hid_t src_loc, const char *src_name, hid_t dst_loc,
<     const char *dst_name, hid_t lcpl_id, hid_t lapl_id);
< H5_DLL herr_t H5Lcreate_hard(hid_t cur_loc, const char *cur_name,
<     hid_t dst_loc, const char *dst_name, hid_t lcpl_id, hid_t lapl_id);
< H5_DLL herr_t H5Lcreate_soft(const char *link_target, hid_t link_loc_id,
<     const char *link_name, hid_t lcpl_id, hid_t lapl_id);
____
> H5_DLL herr_t H5Lmove(hid_t src_loc, const char *src_name, hid_t dst_loc, const char *dst_name, hid_t lcpl_id,
>     hid_t lapl_id);
> H5_DLL herr_t H5Lcopy(hid_t src_loc, const char *src_name, hid_t dst_loc, const char *dst_name, hid_t lcpl_id,
>     hid_t lapl_id);
> H5_DLL herr_t H5Lcreate_hard(hid_t cur_loc, const char *cur_name, hid_t dst_loc, const char *dst_name,
>     hid_t lcpl_id, hid_t lapl_id);
> H5_DLL herr_t H5Lcreate_soft(const char *link_target, hid_t link_loc_id, const char *link_name, hid_t lcpl_id,
>     hid_t lapl_id);
11,17c11,16
< H5_DLL herr_t H5Ldelete_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t lapl_id);
< H5_DLL herr_t H5Lget_val(hid_t loc_id, const char *name, void *buf,
<     size_t size, hid_t lapl_id);
< H5_DLL herr_t H5Lget_val_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<     void *buf, size_t size, hid_t lapl_id);
____
> H5_DLL herr_t H5Ldelete_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, hid_t lapl_id);
> H5_DLL herr_t H5Lget_val(hid_t loc_id, const char *name, void *buf, size_t size, hid_t lapl_id);
> H5_DLL herr_t H5Lget_val_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, void *buf, size_t size,
>     hid_t lapl_id);
19,39c18,34
< H5_DLL herr_t H5Lget_info(hid_t loc_id, const char *name,
<     H5L_info_t *linfo, hid_t lapl_id);
< H5_DLL herr_t H5Lget_info_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<     H5L_info_t *linfo, hid_t lapl_id);
< H5_DLL ssize_t H5Lget_name_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n,
<     char *name, size_t size, hid_t lapl_id);
< H5_DLL herr_t H5Literate(hid_t grp_id, H5_index_t idx_type,
<     H5_iter_order_t order, hsize_t *idx, H5L_iterate_t op, void *op_data);
< H5_DLL herr_t H5Literate_by_name(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
<     H5L_iterate_t op, void *op_data, hid_t lapl_id);

```

```

< H5_DLL herr_t H5Lvisit(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order,
<     H5L_iterate_t op, void *op_data);
< H5_DLL herr_t H5Lvisit_by_name(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, H5L_iterate_t op,
<     void *op_data, hid_t lcpl_id);
< H5_DLL herr_t H5Lcreate_ud(hid_t link_loc_id, const char *link_name,
<     H5L_type_t link_type, const void *udata, size_t udata_size, hid_t lcpl_id,
<     hid_t lapl_id);
____
> H5_DLL herr_t H5Lget_info2(hid_t loc_id, const char *name, H5L_info2_t *linfo, hid_t lapl_id);
> H5_DLL herr_t H5Lget_info_by_idx2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, H5L_info2_t *linfo, hid_t lapl_id);
> H5_DLL ssize_t H5Lget_name_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, char *name, size_t size,
>     hid_t lapl_id);
> H5_DLL herr_t H5Literate2(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
>     H5L_iterate2_t op, void *op_data);
> H5_DLL herr_t H5Literate_by_name2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t *idx, H5L_iterate2_t op, void *op_data,
>     hid_t lapl_id);
> H5_DLL herr_t H5Lvisit2(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order, H5L_iterate2_t op,
>     void *op_data);
> H5_DLL herr_t H5Lvisit_by_name2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, H5L_iterate2_t op, void *op_data, hid_t lapl_id);
> H5_DLL herr_t H5Lcreate_ud(hid_t link_loc_id, const char *link_name, H5L_type_t link_type, const void *udata,
>     size_t udata_size, hid_t lcpl_id, hid_t lapl_id);
43,46c38,54
< H5_DLL herr_t H5Lunpack_elink_val(const void *ext_linkval, size_t link_size,
<     unsigned *flags, const char **filename, const char **obj_path );
< H5_DLL herr_t H5Lcreate_external(const char *file_name, const char *obj_name,
<     hid_t link_loc_id, const char *link_name, hid_t lcpl_id, hid_t lapl_id);
____
> H5_DLL herr_t H5Lunpack_elink_val(const void *ext_linkval, size_t link_size, unsigned *flags,
>     const char **filename, const char **obj_path );
> H5_DLL herr_t H5Lcreate_external(const char *file_name, const char *obj_name, hid_t link_loc_id,
>     const char *link_name, hid_t lcpl_id, hid_t lapl_id);
> H5_DLL herr_t H5Lget_info1(hid_t loc_id, const char *name, H5L_info1_t *linfo, hid_t lapl_id);
> H5_DLL herr_t H5Lget_info_by_idx1(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, H5L_info1_t *linfo,
>     hid_t lapl_id);
> H5_DLL herr_t H5Literate1(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order, hsize_t *idx,
>     H5L_iterate1_t op, void *op_data);
> H5_DLL herr_t H5Literate_by_name1(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t *idx, H5L_iterate1_t op, void *op_data,

```

```

>                                hid_t lapl_id);
> H5_DLL herr_t H5Lvisit1(hid_t grp_id, H5_index_t idx_type, H5_iter_order_t order, H5L_iterate1_t op,
>                                void *op_data);
> H5_DLL herr_t H5Lvisit_by_name1(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>                                H5_iter_order_t order, H5L_iterate1_t op, void *op_data, hid_t lapl_id);

```

47a56

>

改变 H5LTpublic.h

2,197c2,79

```

< H5_HLDLL herr_t  H5LTmake_dataset( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,
<                                hid_t type_id,
<                                const void *buffer );
< H5_HLDLL herr_t  H5LTmake_dataset_char( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,
<                                const char *buffer );
< H5_HLDLL herr_t  H5LTmake_dataset_short( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,
<                                const short *buffer );
< H5_HLDLL herr_t  H5LTmake_dataset_int( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,
<                                const int *buffer );
< H5_HLDLL herr_t  H5LTmake_dataset_long( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,
<                                const long *buffer );
< H5_HLDLL herr_t  H5LTmake_dataset_float( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,
<                                const float *buffer );
< H5_HLDLL herr_t  H5LTmake_dataset_double( hid_t loc_id,
<                                const char *dset_name,
<                                int rank,
<                                const hsize_t *dims,

```



```

<                                const char *attr_name,
<                                const char *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_uchar( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const unsigned char *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_short( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const short *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_ushort( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const unsigned short *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_int( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const int *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_uint( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const unsigned int *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_long( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const long *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_long_long( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const long long *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_ulong( hid_t loc_id,
<                                const char *obj_name,
<                                const char *attr_name,
<                                const unsigned long *buffer,
<                                size_t size );
< H5_HLDLL herr_t  H5LTset_attribute_float( hid_t loc_id,

```



```

> H5_HLDLL herr_t H5Tmake_dataset_string(hid_t loc_id, const char *dset_name, const char *buf);
> H5_HLDLL herr_t H5Tread_dataset(hid_t loc_id, const char *dset_name, hid_t type_id, void *buffer);
> H5_HLDLL herr_t H5Tread_dataset_char(hid_t loc_id, const char *dset_name, char *buffer);
> H5_HLDLL herr_t H5Tread_dataset_short(hid_t loc_id, const char *dset_name, short *buffer);
> H5_HLDLL herr_t H5Tread_dataset_int(hid_t loc_id, const char *dset_name, int *buffer);
> H5_HLDLL herr_t H5Tread_dataset_long(hid_t loc_id, const char *dset_name, long *buffer);
> H5_HLDLL herr_t H5Tread_dataset_float(hid_t loc_id, const char *dset_name, float *buffer);
> H5_HLDLL herr_t H5Tread_dataset_double(hid_t loc_id, const char *dset_name, double *buffer);
> H5_HLDLL herr_t H5Tread_dataset_string(hid_t loc_id, const char *dset_name, char *buf);
> H5_HLDLL herr_t H5Tget_dataset_ndims(hid_t loc_id, const char *dset_name, int *rank);
> H5_HLDLL herr_t H5Tget_dataset_info(hid_t loc_id, const char *dset_name, hsize_t *dims,
>
>                                     H5T_class_t *type_class, size_t *type_size);
> H5_HLDLL herr_t H5Tfind_dataset(hid_t loc_id, const char *name);
> H5_HLDLL herr_t H5Tset_attribute_string(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const char *attr_data);
> H5_HLDLL herr_t H5Tset_attribute_char(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const char *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_uchar(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned char *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_short(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const short *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_ushort(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned short *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_int(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const int *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_uint(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned int *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_long(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const long *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_long_long(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const long long *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_ulong(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const unsigned long *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_float(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const float *buffer, size_t size);
> H5_HLDLL herr_t H5Tset_attribute_double(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     const double *buffer, size_t size);
> H5_HLDLL herr_t H5Tget_attribute(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     hid_t mem_type_id, void *data);
> H5_HLDLL herr_t H5Tget_attribute_string(hid_t loc_id, const char *obj_name, const char *attr_name,
>
>                                     char *data);
> H5_HLDLL herr_t H5Tget_attribute_char(hid_t loc_id, const char *obj_name, const char *attr_name, char
*data);
> H5_HLDLL herr_t H5Tget_attribute_uchar(hid_t loc_id, const char *obj_name, const char *attr_name,

```

```

> unsigned char *data);
> H5_HLDLL herr_t H5LTget_attribute_short(hid_t loc_id, const char *obj_name, const char *attr_name,
> short *data);
> H5_HLDLL herr_t H5LTget_attribute_ushort(hid_t loc_id, const char *obj_name, const char *attr_name,
> unsigned short *data);
> H5_HLDLL herr_t H5LTget_attribute_int(hid_t loc_id, const char *obj_name, const char *attr_name, int *data);
> H5_HLDLL herr_t H5LTget_attribute_uint(hid_t loc_id, const char *obj_name, const char *attr_name,
> unsigned int *data);
> H5_HLDLL herr_t H5LTget_attribute_long(hid_t loc_id, const char *obj_name, const char *attr_name, long
*data);
> H5_HLDLL herr_t H5LTget_attribute_long_long(hid_t loc_id, const char *obj_name, const char *attr_name,
> long long *data);
> H5_HLDLL herr_t H5LTget_attribute_ulong(hid_t loc_id, const char *obj_name, const char *attr_name,
> unsigned long *data);
> H5_HLDLL herr_t H5LTget_attribute_float(hid_t loc_id, const char *obj_name, const char *attr_name,
> float *data);
> H5_HLDLL herr_t H5LTget_attribute_double(hid_t loc_id, const char *obj_name, const char *attr_name,
> double *data);
> H5_HLDLL herr_t H5LTget_attribute_ndims(hid_t loc_id, const char *obj_name, const char *attr_name, int
*rank);
> H5_HLDLL herr_t H5LTget_attribute_info(hid_t loc_id, const char *obj_name, const char *attr_name,
> hsize_t *dims, H5T_class_t *type_class, size_t *type_size);
> H5_HLDLL hid_t H5LTtext_to_dtype(const char *text, H5LT_lang_t lang_type);
199c81
< H5_HLDLL herr_t H5LTfind_attribute( hid_t loc_id, const char *name );
----
> H5_HLDLL herr_t H5LTfind_attribute(hid_t loc_id, const char *name);
改变 H5MMpublic.h
2a3
>
增加头文件： H5Mpublic.h
改变 H5Opublic.h
3,5c3,5
< H5_DLL hid_t H5Oopen_by_addr(hid_t loc_id, haddr_t addr);
< H5_DLL hid_t H5Oopen_by_idx(hid_t loc_id, const char *group_name,
< H5_index_t idx_type, H5_iter_order_t order, hsize_t n, hid_t lapl_id);
----
> H5_DLL hid_t H5Oopen_by_token(hid_t loc_id, H5O_token_t token);
> H5_DLL hid_t H5Oopen_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type, H5_iter_order_t
order,
> hsize_t n, hid_t lapl_id);
7,14c7,19
< H5_DLL herr_t H5Oget_info(hid_t loc_id, H5O_info_t *oinfo);
< H5_DLL herr_t H5Oget_info_by_name(hid_t loc_id, const char *name, H5O_info_t *oinfo,

```

```

<     hid_t lapl_id);
< H5_DLL herr_t H5Oget_info_by_idx(hid_t loc_id, const char *group_name,
<     H5_index_t idx_type, H5_iter_order_t order, hsize_t n, H5O_info_t *oinfo,
<     hid_t lapl_id);
< H5_DLL herr_t H5Olink(hid_t obj_id, hid_t new_loc_id, const char *new_name,
<     hid_t lcpl_id, hid_t lapl_id);
____
> H5_DLL herr_t H5Oget_info3(hid_t loc_id, H5O_info2_t *oinfo, unsigned fields);
> H5_DLL herr_t H5Oget_info_by_name3(hid_t loc_id, const char *name, H5O_info2_t *oinfo, unsigned fields,
>     hid_t lapl_id);
> H5_DLL herr_t H5Oget_info_by_idx3(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, H5O_info2_t *oinfo, unsigned fields,
>     hid_t lapl_id);
> H5_DLL herr_t H5Oget_native_info(hid_t loc_id, H5O_native_info_t *oinfo, unsigned fields);
> H5_DLL herr_t H5Oget_native_info_by_name(hid_t loc_id, const char *name, H5O_native_info_t *oinfo,
>     unsigned fields, hid_t lapl_id);
> H5_DLL herr_t H5Oget_native_info_by_idx(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>     H5_iter_order_t order, hsize_t n, H5O_native_info_t *oinfo,
>     unsigned fields, hid_t lapl_id);
> H5_DLL herr_t H5Olink(hid_t obj_id, hid_t new_loc_id, const char *new_name, hid_t lcpl_id, hid_t lapl_id);
17,18c22,23
< H5_DLL herr_t H5Ocopy(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id,
<     const char *dst_name, hid_t ocpypl_id, hid_t lcpl_id);
____
> H5_DLL herr_t H5Ocopy(hid_t src_loc_id, const char *src_name, hid_t dst_loc_id, const char *dst_name,
>     hid_t ocpypl_id, hid_t lcpl_id);
20,21c25
< H5_DLL herr_t H5Oset_comment_by_name(hid_t loc_id, const char *name,
<     const char *comment, hid_t lapl_id);
____
> H5_DLL herr_t H5Oset_comment_by_name(hid_t loc_id, const char *name, const char *comment, hid_t lapl_id);
23,29c27,33
< H5_DLL ssize_t H5Oget_comment_by_name(hid_t loc_id, const char *name,
<     char *comment, size_t bufsize, hid_t lapl_id);
< H5_DLL herr_t H5Ovisit(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order,
<     H5O_iterate_t op, void *op_data);
< H5_DLL herr_t H5Ovisit_by_name(hid_t loc_id, const char *obj_name,
<     H5_index_t idx_type, H5_iter_order_t order, H5O_iterate_t op,
<     void *op_data, hid_t lapl_id);
____
> H5_DLL ssize_t H5Oget_comment_by_name(hid_t loc_id, const char *name, char *comment, size_t bufsize,
>     hid_t lapl_id);
> H5_DLL herr_t H5Ovisit3(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order, H5O_iterate2_t op,
>     void *op_data, unsigned fields);

```

```

> H5_DLL herr_t H5Ovisit_by_name3(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, H5O_iterate2_t op, void *op_data, unsigned
fields,
>
>                                     hid_t lapl_id);
30a35,64
> H5_DLL herr_t H5Oflush(hid_t obj_id);
> H5_DLL herr_t H5Orefresh(hid_t oid);
> H5_DLL herr_t H5Odisable_mdc_flushes(hid_t object_id);
> H5_DLL herr_t H5Oenable_mdc_flushes(hid_t object_id);
> H5_DLL herr_t H5Oare_mdc_flushes_disabled(hid_t object_id, hbool_t *are_disabled);
> H5_DLL herr_t H5Otoken_cmp(hid_t loc_id, const H5O_token_t *token1, const H5O_token_t *token2,
>
>                                     int *cmp_value);
> H5_DLL herr_t H5Otoken_to_str(hid_t loc_id, const H5O_token_t *token, char **token_str);
> H5_DLL herr_t H5Otoken_from_str(hid_t loc_id, const char *token_str, H5O_token_t *token);
> H5_DLLVAR const H5O_token_t H5O_TOKEN_UNDEF_g;
> H5_DLL hid_t H5Oopen_by_addr(hid_t loc_id, haddr_t addr);
> H5_DLL herr_t H5Oget_info1(hid_t loc_id, H5O_info1_t *oinfo);
> H5_DLL herr_t H5Oget_info_by_name1(hid_t loc_id, const char *name, H5O_info1_t *oinfo, hid_t lapl_id);
> H5_DLL herr_t H5Oget_info_by_idx1(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, H5O_info1_t *oinfo, hid_t lapl_id);
> H5_DLL herr_t H5Oget_info2(hid_t loc_id, H5O_info1_t *oinfo, unsigned fields);
> H5_DLL herr_t H5Oget_info_by_name2(hid_t loc_id, const char *name, H5O_info1_t *oinfo, unsigned fields,
>
>                                     hid_t lapl_id);
> H5_DLL herr_t H5Oget_info_by_idx2(hid_t loc_id, const char *group_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, hsize_t n, H5O_info1_t *oinfo, unsigned fields,
>
>                                     hid_t lapl_id);
> H5_DLL herr_t H5Ovisit1(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order, H5O_iterate1_t op,
>
>                                     void *op_data);
> H5_DLL herr_t H5Ovisit_by_name1(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, H5O_iterate1_t op, void *op_data, hid_t lapl_id);
> H5_DLL herr_t H5Ovisit2(hid_t obj_id, H5_index_t idx_type, H5_iter_order_t order, H5O_iterate1_t op,
>
>                                     void *op_data, unsigned fields);
> H5_DLL herr_t H5Ovisit_by_name2(hid_t loc_id, const char *obj_name, H5_index_t idx_type,
>
>                                     H5_iter_order_t order, H5O_iterate1_t op, void *op_data, unsigned
fields,
>
>                                     hid_t lapl_id);
31a66
>
>
改变 H5PLextern.h
4a5
>
>
改变 H5PLpublic.h
2,7c2,7
< H5_DLL herr_t H5PLset_loading_state(unsigned int plugin_type);

```

```

< H5_DLL herr_t H5PLget_loading_state(unsigned int *plugin_type);
< H5_DLL herr_t H5PLappend(const char *plugin_path);
< H5_DLL herr_t H5PLprepend(const char *plugin_path);
< H5_DLL herr_t H5PLreplace(const char *plugin_path, unsigned int index);
< H5_DLL herr_t H5PLinsert(const char *plugin_path, unsigned int index);
____

> H5_DLL herr_t H5PLset_loading_state(unsigned int plugin_control_mask);
> H5_DLL herr_t H5PLget_loading_state(unsigned int *plugin_control_mask );
> H5_DLL herr_t H5PLappend(const char *search_path);
> H5_DLL herr_t H5PLprepend(const char *search_path);
> H5_DLL herr_t H5PLreplace(const char *search_path, unsigned int index);
> H5_DLL herr_t H5PLinsert(const char *search_path, unsigned int index);
9,10c9,10
< H5_DLL ssize_t H5PLget(unsigned int index, char *pathname, size_t size);
< H5_DLL herr_t H5PLsize(unsigned int *listsize);
____

> H5_DLL ssize_t H5PLget(unsigned int index, char *path_buf , size_t buf_size);
> H5_DLL herr_t H5PLsize(unsigned int *num_paths );
11a12
>
改变 H5Ppublic.h
13a14,15
> H5_DLLVAR hid_t H5P_CLS_MAP_CREATE_ID_g;
> H5_DLLVAR hid_t H5P_CLS_MAP_ACCESS_ID_g;
15a18
> H5_DLLVAR hid_t H5P_CLS_ATTRIBUTE_ACCESS_ID_g;
18a22,23
> H5_DLLVAR hid_t H5P_CLS_VOL_INITIALIZE_ID_g;
> H5_DLLVAR hid_t H5P_CLS_REFERENCE_ACCESS_ID_g;
28a34,35
> H5_DLLVAR hid_t H5P_LST_MAP_CREATE_ID_g;
> H5_DLLVAR hid_t H5P_LST_MAP_ACCESS_ID_g;
29a37
> H5_DLLVAR hid_t H5P_LST_ATTRIBUTE_ACCESS_ID_g;
33,37c41,46
< H5_DLL hid_t H5Pcreate_class(hid_t parent, const char *name,
<     H5P_cls_create_func_t cls_create, void *create_data,
<     H5P_cls_copy_func_t cls_copy, void *copy_data,
<     H5P_cls_close_func_t cls_close, void *close_data);
< H5_DLL char *H5Pget_class_name(hid_t pclass_id);
____

> H5_DLLVAR hid_t H5P_LST_VOL_INITIALIZE_ID_g;
> H5_DLLVAR hid_t H5P_LST_REFERENCE_ACCESS_ID_g;
> H5_DLL herr_t H5Pclose(hid_t plist_id);

```

```

> H5_DLL herr_t H5Pclose_class(hid_t plist_id);
> H5_DLL hid_t H5Pcopy(hid_t plist_id);
> H5_DLL herr_t H5Pcopy_prop(hid_t dst_id, hid_t src_id, const char *name);
39,48c48,53
< H5_DLL herr_t H5Pregister2(hid_t cls_id, const char *name, size_t size,
<     void *def_value, H5P_prp_create_func_t prp_create,
<     H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_compare_func_t prp_cmp, H5P_prp_close_func_t prp_close);
< H5_DLL herr_t H5Pinsert2(hid_t plist_id, const char *name, size_t size,
<     void *value, H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_delete, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_compare_func_t prp_cmp, H5P_prp_close_func_t prp_close);
< H5_DLL herr_t H5Pset(hid_t plist_id, const char *name, void *value);
____
> H5_DLL hid_t H5Pcreate_class(hid_t parent, const char *name, H5P_cls_create_func_t create, void *create_data,
>
>     H5P_cls_copy_func_t copy, void *copy_data, H5P_cls_close_func_t close,
>
>     void *close_data);
> H5_DLL hid_t H5Pdecode(const void *buf);
> H5_DLL herr_t H5Pencode2(hid_t plist_id, void *buf, size_t *nalloc, hid_t fapl_id);
> H5_DLL htri_t H5Pequal(hid_t id1, hid_t id2);
50,51c55
< H5_DLL herr_t H5Pget_size(hid_t id, const char *name, size_t *size);
< H5_DLL herr_t H5Pget_nprops(hid_t id, size_t *nprops);
____
> H5_DLL herr_t H5Pget(hid_t plist_id, const char *name, void *value);
52a57
> H5_DLL char *H5Pget_class_name(hid_t pclass_id);
54,55c59,63
< H5_DLL herr_t H5Pget(hid_t plist_id, const char *name, void *value);
< H5_DLL htri_t H5Pequal(hid_t id1, hid_t id2);
____
> H5_DLL herr_t H5Pget_nprops(hid_t id, size_t *nprops);
> H5_DLL herr_t H5Pget_size(hid_t id, const char *name, size_t *size);
> H5_DLL herr_t H5Pinsert2(hid_t plist_id, const char *name, size_t size, void *value, H5P_prp_set_func_t set,
>
>     H5P_prp_get_func_t get, H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t
copy,
>
>     H5P_prp_compare_func_t compare, H5P_prp_close_func_t close);
57,59c65,69
< H5_DLL int H5Piterate(hid_t id, int *idx, H5P_iterate_t iter_func,
<
<     void *iter_data);
< H5_DLL herr_t H5Pcopy_prop(hid_t dst_id, hid_t src_id, const char *name);
____
> H5_DLL int H5Piterate(hid_t id, int *idx, H5P_iterate_t iter_func, void *iter_data);

```



```

> H5_DLL herr_t H5Pregister2(hid_t cls_id, const char *name, size_t size, void *def_value,
>
>                                     H5P_prp_create_func_t create, H5P_prp_set_func_t set, H5P_prp_get_func_t
get,
>
>                                     H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t copy,
>
>                                     H5P_prp_compare_func_t compare, H5P_prp_close_func_t close);
60a71
> H5_DLL herr_t H5Pset(hid_t plist_id, const char *name, const void *value);
62,67c73
< H5_DLL herr_t H5Pclose_class(hid_t plist_id);
< H5_DLL herr_t H5Pclose(hid_t plist_id);
< H5_DLL hid_t H5Pcopy(hid_t plist_id);
< H5_DLL herr_t H5Pset_attr_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
< H5_DLL herr_t H5Pget_attr_phase_change(hid_t plist_id, unsigned *max_compact, unsigned *min_dense);
< H5_DLL herr_t H5Pset_attr_creation_order(hid_t plist_id, unsigned crt_order_flags);
____
> H5_DLL htri_t H5Pall_filters_avail(hid_t plist_id);
69,76c75,81
< H5_DLL herr_t H5Pset_obj_track_times(hid_t plist_id, hbool_t track_times);
< H5_DLL herr_t H5Pget_obj_track_times(hid_t plist_id, hbool_t *track_times);
< H5_DLL herr_t H5Pmodify_filter(hid_t plist_id, H5Z_filter_t filter,
<
<         unsigned int flags, size_t cd_nelmts,
<
<         const unsigned int cd_values[]);
< H5_DLL herr_t H5Pset_filter(hid_t plist_id, H5Z_filter_t filter,
<
<         unsigned int flags, size_t cd_nelmts,
<
<         const unsigned int c_values[]);
____
> H5_DLL herr_t H5Pget_attr_phase_change(hid_t plist_id, unsigned *max_compact, unsigned *min_dense);
> H5_DLL H5Z_filter_t H5Pget_filter2(hid_t plist_id, unsigned idx, unsigned int *flags ,
>
>                                     size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
>
>                                     char name[], unsigned *filter_config );
> H5_DLL herr_t H5Pget_filter_by_id2(hid_t plist_id, H5Z_filter_t filter_id, unsigned int *flags ,
>
>                                     size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
>
>                                     char name[] , unsigned *filter_config );
78,88c83,85
< H5_DLL H5Z_filter_t H5Pget_filter2(hid_t plist_id, unsigned filter,
<
<         unsigned int *flags,
<
<         size_t *cd_nelmts,
<
<         unsigned cd_values[],
<
<         size_t namelen, char name[],
<
<         unsigned *filter_config );
< H5_DLL herr_t H5Pget_filter_by_id2(hid_t plist_id, H5Z_filter_t id,
<
<         unsigned int *flags, size_t *cd_nelmts,
<
<         unsigned cd_values[], size_t namelen, char name[],
<
<         unsigned *filter_config);

```

```

< H5_DLL htri_t H5Pall_filters_avail(hid_t plist_id);
----
> H5_DLL herr_t H5Pget_obj_track_times(hid_t plist_id, hbool_t *track_times);
> H5_DLL herr_t H5Pmodify_filter(hid_t plist_id, H5Z_filter_t filter, unsigned int flags, size_t cd_nelmts,
>
const unsigned int cd_values[]);
90c87,91
< H5_DLL herr_t H5Pset_deflate(hid_t plist_id, unsigned aggression);
----
> H5_DLL herr_t H5Pset_attr_creation_order(hid_t plist_id, unsigned crt_order_flags);
> H5_DLL herr_t H5Pset_attr_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
> H5_DLL herr_t H5Pset_deflate(hid_t plist_id, unsigned level);
> H5_DLL herr_t H5Pset_filter(hid_t plist_id, H5Z_filter_t filter, unsigned int flags, size_t cd_nelmts,
>
const unsigned int c_values[]);
92,95c93,103
< H5_DLL herr_t H5Pget_version(hid_t plist_id, unsigned *boot,
<
unsigned *freelist, unsigned *stab,
<
unsigned *shhdr);
< H5_DLL herr_t H5Pset_userblock(hid_t plist_id, hsize_t size);
----
> H5_DLL herr_t H5Pset_obj_track_times(hid_t plist_id, hbool_t track_times);
> H5_DLL herr_t H5Pget_file_space_page_size(hid_t plist_id, hsize_t *fsp_size);
> H5_DLL herr_t H5Pget_file_space_strategy(hid_t plist_id, H5F_fspace_strategy_t *strategy, hbool_t *persist,
>
hsize_t *threshold);
> H5_DLL herr_t H5Pget_istore_k(hid_t plist_id, unsigned *ik );
> H5_DLL herr_t H5Pget_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned *mesg_type_flags,
>
unsigned *min_mesg_size);
> H5_DLL herr_t H5Pget_shared_mesg_nindexes(hid_t plist_id, unsigned *nindexes);
> H5_DLL herr_t H5Pget_shared_mesg_phase_change(hid_t plist_id, unsigned *max_list, unsigned *min_btree);
> H5_DLL herr_t H5Pget_sizes(hid_t plist_id, size_t *sizeof_addr , size_t *sizeof_size );
> H5_DLL herr_t H5Pget_sym_k(hid_t plist_id, unsigned *ik , unsigned *lk );
97,102c105,107
< H5_DLL herr_t H5Pset_sizes(hid_t plist_id, size_t sizeof_addr,
<
size_t sizeof_size);
< H5_DLL herr_t H5Pget_sizes(hid_t plist_id, size_t *sizeof_addr,
<
size_t *sizeof_size);
< H5_DLL herr_t H5Pset_sym_k(hid_t plist_id, unsigned ik, unsigned lk);
< H5_DLL herr_t H5Pget_sym_k(hid_t plist_id, unsigned *ik, unsigned *lk);
----
> H5_DLL herr_t H5Pset_file_space_page_size(hid_t plist_id, hsize_t fsp_size);
> H5_DLL herr_t H5Pset_file_space_strategy(hid_t plist_id, H5F_fspace_strategy_t strategy, hbool_t persist,
>
hsize_t threshold);
104c109,110
< H5_DLL herr_t H5Pget_istore_k(hid_t plist_id, unsigned *ik);
----

```

```

> H5_DLL herr_t H5Pset_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned mesg_type_flags,
>
>                                     unsigned min_mesg_size);
106,108d111
< H5_DLL herr_t H5Pget_shared_mesg_nindexes(hid_t plist_id, unsigned *nindexes);
< H5_DLL herr_t H5Pset_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned mesg_type_flags,
unsigned min_mesg_size);
< H5_DLL herr_t H5Pget_shared_mesg_index(hid_t plist_id, unsigned index_num, unsigned *mesg_type_flags,
unsigned *min_mesg_size);
110,116c113,119
< H5_DLL herr_t H5Pget_shared_mesg_phase_change(hid_t plist_id, unsigned *max_list, unsigned *min_btree);
< H5_DLL herr_t H5Pset_alignment(hid_t fapl_id, hsize_t threshold,
<
<     hsize_t alignment);
< H5_DLL herr_t H5Pget_alignment(hid_t fapl_id, hsize_t *threshold,
<
<     hsize_t *alignment);
< H5_DLL herr_t H5Pset_driver(hid_t plist_id, hid_t driver_id,
<
<     const void *driver_info);
-----
> H5_DLL herr_t H5Pset_sizes(hid_t plist_id, size_t sizeof_addr, size_t sizeof_size);
> H5_DLL herr_t H5Pset_sym_k(hid_t plist_id, unsigned ik, unsigned lk);
> H5_DLL herr_t H5Pset_userblock(hid_t plist_id, hsize_t size);
> H5_DLL herr_t H5Pget_alignment(hid_t fapl_id, hsize_t *threshold, hsize_t *alignment);
> H5_DLL herr_t H5Pget_cache(hid_t plist_id, int *mdc_nelmts,
>
>     size_t *rdcc_nslots, size_t *rdcc_nbytes, double *rdcc_w0);
> H5_DLL herr_t H5Pget_core_write_tracking(hid_t fapl_id, hbool_t *is_enabled, size_t *page_size);
118,119c121,123
< H5_DLL void *H5Pget_driver_info(hid_t plist_id);
< H5_DLL herr_t H5Pset_family_offset(hid_t fapl_id, hsize_t offset);
-----
> H5_DLL const void *H5Pget_driver_info(hid_t plist_id);
> H5_DLL herr_t H5Pget_elink_file_cache_size(hid_t plist_id, unsigned *efc_size);
> H5_DLL herr_t H5Pget_evict_on_close(hid_t fapl_id, hbool_t *evict_on_close);
121c125,136
< H5_DLL herr_t H5Pset_multi_type(hid_t fapl_id, H5FD_mem_t type);
-----
> H5_DLL herr_t H5Pget_fcclose_degree(hid_t fapl_id, H5F_close_degree_t *degree);
> H5_DLL herr_t H5Pget_file_image(hid_t fapl_id, void **buf_ptr_ptr, size_t *buf_len_ptr);
> H5_DLL herr_t H5Pget_file_image_callbacks(hid_t fapl_id, H5FD_file_image_callbacks_t *callbacks_ptr);
> H5_DLL herr_t H5Pget_file_locking(hid_t fapl_id, hbool_t *use_file_locking, hbool_t *ignore_when_disabled);
> H5_DLL herr_t H5Pget_gc_references(hid_t fapl_id, unsigned *gc_ref);
> H5_DLL herr_t H5Pget_libver_bounds(hid_t plist_id, H5F_libver_t *low, H5F_libver_t *high);
> H5_DLL herr_t H5Pget_mdc_config(hid_t plist_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Pget_mdc_image_config(hid_t plist_id, H5AC_cache_image_config_t *config_ptr);
> H5_DLL herr_t H5Pget_mdc_log_options(hid_t plist_id, hbool_t *is_enabled, char *location,
>
>     size_t *location_size, hbool_t *start_on_access);

```

```

> H5_DLL herr_t H5Pget_meta_block_size(hid_t fapl_id, hsize_t *size);
> H5_DLL herr_t H5Pget_metadata_read_attempts(hid_t plist_id, unsigned *attempts);
123,135c138,152
< H5_DLL herr_t H5Pset_cache(hid_t plist_id, int mdc_nelmts,
<         size_t rdcc_nslots, size_t rdcc_nbytes,
<         double rdcc_w0);
< H5_DLL herr_t H5Pget_cache(hid_t plist_id,
<         int *mdc_nelmts,
<         size_t *rdcc_nslots,
<         size_t *rdcc_nbytes, double *rdcc_w0);
< H5_DLL herr_t H5Pset_mdc_config(hid_t      plist_id,
<         H5AC_cache_config_t * config_ptr);
< H5_DLL herr_t H5Pget_mdc_config(hid_t      plist_id,
<         H5AC_cache_config_t * config_ptr);
< H5_DLL herr_t H5Pset_gc_references(hid_t fapl_id, unsigned gc_ref);
< H5_DLL herr_t H5Pget_gc_references(hid_t fapl_id, unsigned *gc_ref);
-----
> H5_DLL herr_t H5Pget_object_flush_cb(hid_t plist_id, H5F_flush_cb_t *func, void **udata);
> H5_DLL herr_t H5Pget_page_buffer_size(hid_t plist_id, size_t *buf_size, unsigned *min_meta_perc,
>         unsigned *min_raw_perc);
> H5_DLL herr_t H5Pget_sieve_buf_size(hid_t fapl_id, size_t *size );
> H5_DLL herr_t H5Pget_small_data_block_size(hid_t fapl_id, hsize_t *size );
> H5_DLL herr_t H5Pget_vol_id(hid_t plist_id, hid_t *vol_id);
> H5_DLL herr_t H5Pget_vol_info(hid_t plist_id, void **vol_info);
> H5_DLL herr_t H5Pset_alignment(hid_t fapl_id, hsize_t threshold, hsize_t alignment);
> H5_DLL herr_t H5Pset_cache(hid_t plist_id, int mdc_nelmts, size_t rdcc_nslots, size_t rdcc_nbytes,
>         double rdcc_w0);
> H5_DLL herr_t H5Pset_core_write_tracking(hid_t fapl_id, hbool_t is_enabled, size_t page_size);
> H5_DLL herr_t H5Pset_driver(hid_t plist_id, hid_t driver_id, const void *driver_info);
> H5_DLL herr_t H5Pset_elink_file_cache_size(hid_t plist_id, unsigned efc_size);
> H5_DLL herr_t H5Pset_evict_on_close(hid_t fapl_id, hbool_t evict_on_close);
> H5_DLL herr_t H5Pset_family_offset(hid_t fapl_id, hsize_t offset);
137c154,161
< H5_DLL herr_t H5Pget_fcclose_degree(hid_t fapl_id, H5F_close_degree_t *degree);
-----
> H5_DLL herr_t H5Pset_file_image(hid_t fapl_id, void *buf_ptr, size_t buf_len);
> H5_DLL herr_t H5Pset_file_image_callbacks(hid_t fapl_id, H5FD_file_image_callbacks_t *callbacks_ptr);
> H5_DLL herr_t H5Pset_file_locking(hid_t fapl_id, hbool_t use_file_locking, hbool_t ignore_when_disabled);
> H5_DLL herr_t H5Pset_gc_references(hid_t fapl_id, unsigned gc_ref);
> H5_DLL herr_t H5Pset_libver_bounds(hid_t plist_id, H5F_libver_t low, H5F_libver_t high);
> H5_DLL herr_t H5Pset_mdc_config(hid_t plist_id, H5AC_cache_config_t *config_ptr);
> H5_DLL herr_t H5Pset_mdc_log_options(hid_t plist_id, hbool_t is_enabled, const char *location,
>         hbool_t start_on_access);
139c163,165

```

```

< H5_DLL herr_t H5Pget_meta_block_size(hid_t fapl_id, hsize_t *size);
____

> H5_DLL herr_t H5Pset_metadata_read_attempts(hid_t plist_id, unsigned attempts);
> H5_DLL herr_t H5Pset_multi_type(hid_t fapl_id, H5FD_mem_t type);
> H5_DLL herr_t H5Pset_object_flush_cb(hid_t plist_id, H5F_flush_cb_t func, void *udata);
141d166
< H5_DLL herr_t H5Pget_sieve_buf_size(hid_t fapl_id, size_t *size);
143,158c168,187
< H5_DLL herr_t H5Pget_small_data_block_size(hid_t fapl_id, hsize_t *size);
< H5_DLL herr_t H5Pset_libver_bounds(hid_t plist_id, H5F_libver_t low,
<     H5F_libver_t high);
< H5_DLL herr_t H5Pget_libver_bounds(hid_t plist_id, H5F_libver_t *low,
<     H5F_libver_t *high);
< H5_DLL herr_t H5Pset_elink_file_cache_size(hid_t plist_id, unsigned efc_size);
< H5_DLL herr_t H5Pget_elink_file_cache_size(hid_t plist_id, unsigned *efc_size);
< H5_DLL herr_t H5Pset_file_image(hid_t fapl_id, void *buf_ptr, size_t buf_len);
< H5_DLL herr_t H5Pget_file_image(hid_t fapl_id, void **buf_ptr_ptr, size_t *buf_len_ptr);
< H5_DLL herr_t H5Pset_file_image_callbacks(hid_t fapl_id,
<     H5FD_file_image_callbacks_t *callbacks_ptr);
< H5_DLL herr_t H5Pget_file_image_callbacks(hid_t fapl_id,
<     H5FD_file_image_callbacks_t *callbacks_ptr);
< H5_DLL herr_t H5Pset_core_write_tracking(hid_t fapl_id, hbool_t is_enabled, size_t page_size);
< H5_DLL herr_t H5Pget_core_write_tracking(hid_t fapl_id, hbool_t *is_enabled, size_t *page_size);
< H5_DLL herr_t H5Pset_layout(hid_t plist_id, H5D_layout_t layout);
____

> H5_DLL herr_t H5Pset_vol(hid_t plist_id, hid_t new_vol_id, const void *new_vol_info);
> H5_DLL herr_t H5Pset_all_coll_metadata_ops(hid_t plist_id, hbool_t is_collective);
> H5_DLL herr_t H5Pget_all_coll_metadata_ops(hid_t plist_id, hbool_t *is_collective);
> H5_DLL herr_t H5Pset_coll_metadata_write(hid_t plist_id, hbool_t is_collective);
> H5_DLL herr_t H5Pget_coll_metadata_write(hid_t plist_id, hbool_t *is_collective);
> H5_DLL herr_t H5Pget_mpi_params(hid_t fapl_id, MPI_Comm *comm, MPI_Info *info);
> H5_DLL herr_t H5Pset_mpi_params(hid_t fapl_id, MPI_Comm comm, MPI_Info info);
> H5_DLL herr_t H5Pset_mdc_image_config(hid_t plist_id, H5AC_cache_image_config_t *config_ptr);
> H5_DLL herr_t H5Pset_page_buffer_size(hid_t plist_id, size_t buf_size, unsigned min_meta_per,
>     unsigned min_raw_per);
> H5_DLL herr_t H5Pfill_value_defined(hid_t plist, H5D_fill_value_t *status);
> H5_DLL herr_t H5Pget_alloc_time(hid_t plist_id, H5D_alloc_time_t *alloc_time );
> H5_DLL int H5Pget_chunk(hid_t plist_id, int max_ndims, hsize_t dim[] );
> H5_DLL herr_t H5Pget_chunk_opts(hid_t plist_id, unsigned *opts);
> H5_DLL herr_t H5Pget_dset_no_attrs_hint(hid_t dcpl_id, hbool_t *minimize);
> H5_DLL herr_t H5Pget_external(hid_t plist_id, unsigned idx, size_t name_size, char *name ,
>     off_t *offset , hsize_t *size );
> H5_DLL int H5Pget_external_count(hid_t plist_id);
> H5_DLL herr_t H5Pget_fill_time(hid_t plist_id, H5D_fill_time_t *fill_time );

```

```

> H5_DLL herr_t H5Pget_fill_value(hid_t plist_id, hid_t type_id, void *value );
159a189,194

> H5_DLL herr_t H5Pget_virtual_count(hid_t dcpl_id, size_t *count );

> H5_DLL ssize_t H5Pget_virtual_dsetname(hid_t dcpl_id, size_t index, char *name , size_t size);

> H5_DLL ssize_t H5Pget_virtual_filename(hid_t dcpl_id, size_t index, char *name , size_t size);

> H5_DLL hid_t H5Pget_virtual_srcspace(hid_t dcpl_id, size_t index);

> H5_DLL hid_t H5Pget_virtual_vspace(hid_t dcpl_id, size_t index);

> H5_DLL herr_t H5Pset_alloc_time(hid_t plist_id, H5D_alloc_time_t alloc_time);
161,168c196,200

< H5_DLL int H5Pget_chunk(hid_t plist_id, int max_ndims, hsize_t dim[]);

< H5_DLL herr_t H5Pset_external(hid_t plist_id, const char *name, off_t offset,
<
    hsize_t size);

< H5_DLL int H5Pget_external_count(hid_t plist_id);

< H5_DLL herr_t H5Pget_external(hid_t plist_id, unsigned idx, size_t name_size,
<
    char *name, off_t *offset,
<
    hsize_t *size);

< H5_DLL herr_t H5Pset_szip(hid_t plist_id, unsigned options_mask, unsigned pixels_per_block);
-----

> H5_DLL herr_t H5Pset_chunk_opts(hid_t plist_id, unsigned opts);

> H5_DLL herr_t H5Pset_dset_no_attrs_hint(hid_t dcpl_id, hbool_t minimize);

> H5_DLL herr_t H5Pset_external(hid_t plist_id, const char *name, off_t offset, hsize_t size);

> H5_DLL herr_t H5Pset_fill_time(hid_t plist_id, H5D_fill_time_t fill_time);

> H5_DLL herr_t H5Pset_fill_value(hid_t plist_id, hid_t type_id, const void *value);
169a202

> H5_DLL herr_t H5Pset_layout(hid_t plist_id, H5D_layout_t layout);
172,198c205,228

< H5_DLL herr_t H5Pset_fill_value(hid_t plist_id, hid_t type_id,
<
    const void *value);

< H5_DLL herr_t H5Pget_fill_value(hid_t plist_id, hid_t type_id,
<
    void *value);

< H5_DLL herr_t H5Pfill_value_defined(hid_t plist, H5D_fill_value_t *status);

< H5_DLL herr_t H5Pset_alloc_time(hid_t plist_id, H5D_alloc_time_t
<
    alloc_time);

< H5_DLL herr_t H5Pget_alloc_time(hid_t plist_id, H5D_alloc_time_t
<
    *alloc_time);

< H5_DLL herr_t H5Pset_fill_time(hid_t plist_id, H5D_fill_time_t fill_time);

< H5_DLL herr_t H5Pget_fill_time(hid_t plist_id, H5D_fill_time_t
<
    *fill_time);

< H5_DLL herr_t H5Pset_chunk_cache(hid_t dcpl_id, size_t rdcc_nslots,
<
    size_t rdcc_nbytes, double rdcc_w0);

< H5_DLL herr_t H5Pget_chunk_cache(hid_t dcpl_id,
<
    size_t *rdcc_nslots,
<
    size_t *rdcc_nbytes,
<
    double *rdcc_w0);

```

```

< H5_DLL herr_t H5Pset_efile_prefix(hid_t dapl_id, const char* prefix);
< H5_DLL ssize_t H5Pget_efile_prefix(hid_t dapl_id, char* prefix, size_t size);
< H5_DLL herr_t H5Pset_data_transform(hid_t plist_id, const char* expression);
< H5_DLL ssize_t H5Pget_data_transform(hid_t plist_id, char* expression, size_t size);
< H5_DLL herr_t H5Pset_buffer(hid_t plist_id, size_t size, void *tconv,
<         void *bkg);
< H5_DLL size_t H5Pget_buffer(hid_t plist_id, void **tconv,
<         void **bkg);
< H5_DLL herr_t H5Pset_preserve(hid_t plist_id, hbool_t status);
----
> H5_DLL herr_t H5Pset_szip(hid_t plist_id, unsigned options_mask, unsigned pixels_per_block);
> H5_DLL herr_t H5Pset_virtual(hid_t dcpl_id, hid_t vspace_id, const char *src_file_name,
>         const char *src_dset_name, hid_t src_space_id);
> H5_DLL herr_t H5Pget_append_flush(hid_t dapl_id, unsigned dims, hsize_t boundary[], H5D_append_cb_t *func,
>         void **udata);
> H5_DLL herr_t H5Pget_chunk_cache(hid_t dapl_id, size_t *rdcc_nslots, size_t *rdcc_nbytes,
>         double *rdcc_w0);
> H5_DLL ssize_t H5Pget_efile_prefix(hid_t dapl_id, char *prefix, size_t size);
> H5_DLL ssize_t H5Pget_virtual_prefix(hid_t dapl_id, char *prefix, size_t size);
> H5_DLL herr_t H5Pget_virtual_printf_gap(hid_t dapl_id, hsize_t *gap_size);
> H5_DLL herr_t H5Pget_virtual_view(hid_t dapl_id, H5D_vds_view_t *view);
> H5_DLL herr_t H5Pset_append_flush(hid_t dapl_id, unsigned ndims, const hsize_t boundary[],
>         H5D_append_cb_t func, void *udata);
> H5_DLL herr_t H5Pset_chunk_cache(hid_t dapl_id, size_t rdcc_nslots, size_t rdcc_nbytes, double rdcc_w0);
> H5_DLL herr_t H5Pset_efile_prefix(hid_t dapl_id, const char *prefix);
> H5_DLL herr_t H5Pset_virtual_prefix(hid_t dapl_id, const char *prefix);
> H5_DLL herr_t H5Pset_virtual_printf_gap(hid_t dapl_id, hsize_t gap_size);
> H5_DLL herr_t H5Pset_virtual_view(hid_t dapl_id, H5D_vds_view_t view);
> H5_DLL herr_t H5Pget_btree_ratios(hid_t plist_id, double *left, double *middle,
>         double *right);
> H5_DLL size_t H5Pget_buffer(hid_t plist_id, void **tconv, void **bkg);
> H5_DLL ssize_t H5Pget_data_transform(hid_t plist_id, char *expression, size_t size);
> H5_DLL H5Z_EDC_t H5Pget_edc_check(hid_t plist_id);
> H5_DLL herr_t H5Pget_hyper_vector_size(hid_t fapl_id, size_t *size);
199a230,235
> H5_DLL herr_t H5Pget_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t *op, void **operate_data);
> H5_DLL herr_t H5Pget_vlen_mem_manager(hid_t plist_id, H5MM_allocate_t *alloc_func, void **alloc_info,
>         H5MM_free_t *free_func, void **free_info);
> H5_DLL herr_t H5Pset_btree_ratios(hid_t plist_id, double left, double middle, double right);
> H5_DLL herr_t H5Pset_buffer(hid_t plist_id, size_t size, void *tconv, void *bkg);
> H5_DLL herr_t H5Pset_data_transform(hid_t plist_id, const char *expression);
201,222c237,244
< H5_DLL H5Z_EDC_t H5Pget_edc_check(hid_t plist_id);
< H5_DLL herr_t H5Pset_filter_callback(hid_t plist_id, H5Z_filter_func_t func,

```

```

void* op_data);
< H5_DLL herr_t H5Pset_btree_ratios(hid_t plist_id, double left, double middle,
<         double right);
< H5_DLL herr_t H5Pget_btree_ratios(hid_t plist_id, double *left,
<         double *middle,
<         double *right);
< H5_DLL herr_t H5Pset_vlen_mem_manager(hid_t plist_id,
<         H5MM_allocate_t alloc_func,
<         void *alloc_info, H5MM_free_t free_func,
<         void *free_info);
< H5_DLL herr_t H5Pget_vlen_mem_manager(hid_t plist_id,
<         H5MM_allocate_t *alloc_func,
<         void **alloc_info,
<         H5MM_free_t *free_func,
<         void **free_info);
< H5_DLL herr_t H5Pset_hyper_vector_size(hid_t fapl_id, size_t size);
< H5_DLL herr_t H5Pget_hyper_vector_size(hid_t fapl_id, size_t *size);
< H5_DLL herr_t H5Pset_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t op, void* operate_data);
< H5_DLL herr_t H5Pget_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t *op, void** operate_data);
< H5_DLL herr_t H5Pget_mpio_actual_chunk_opt_mode(hid_t plist_id, H5D_mpio_actual_chunk_opt_mode_t
*actual_chunk_opt_mode);
----
> H5_DLL herr_t H5Pset_filter_callback(hid_t plist_id, H5Z_filter_func_t func, void *op_data);
> H5_DLL herr_t H5Pset_hyper_vector_size(hid_t plist_id, size_t size);
> H5_DLL herr_t H5Pset_preserve(hid_t plist_id, hbool_t status);
> H5_DLL herr_t H5Pset_type_conv_cb(hid_t dxpl_id, H5T_conv_except_func_t op, void *operate_data);
> H5_DLL herr_t H5Pset_vlen_mem_manager(hid_t plist_id, H5MM_allocate_t alloc_func, void *alloc_info,
>         H5MM_free_t free_func, void *free_info);
> H5_DLL herr_t H5Pget_mpio_actual_chunk_opt_mode(hid_t plist_id,
>         H5D_mpio_actual_chunk_opt_mode_t
*actual_chunk_opt_mode);
224,225c246,247
< H5_DLL herr_t H5Pget_mpio_no_collective_cause(hid_t plist_id, uint32_t *local_no_collective_cause, uint32_t
*global_no_collective_cause);
< H5_DLL herr_t H5Pset_create_intermediate_group(hid_t plist_id, unsigned crt_intmd);
----
> H5_DLL herr_t H5Pget_mpio_no_collective_cause(hid_t plist_id, uint32_t *local_no_collective_cause,
>         uint32_t *global_no_collective_cause);
227c249,254
< H5_DLL herr_t H5Pset_local_heap_size_hint(hid_t plist_id, size_t size_hint);
----
> H5_DLL herr_t H5Pset_create_intermediate_group(hid_t plist_id, unsigned crt_intmd);
> H5_DLL herr_t H5Pget_est_link_info(hid_t plist_id, unsigned *est_num_entries ,
>         unsigned *est_name_len );

```



```

> H5_DLL herr_t H5Pget_link_creation_order(hid_t plist_id, unsigned *crt_order_flags );
> H5_DLL herr_t H5Pget_link_phase_change(hid_t plist_id, unsigned *max_compact ,
>
>                                     unsigned *min_dense );
229,230d255
< H5_DLL herr_t H5Pset_link_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
< H5_DLL herr_t H5Pget_link_phase_change(hid_t plist_id, unsigned *max_compact , unsigned *min_dense );
232d256
< H5_DLL herr_t H5Pget_est_link_info(hid_t plist_id, unsigned *est_num_entries , unsigned *est_name_len );
234,235c258,262
< H5_DLL herr_t H5Pget_link_creation_order(hid_t plist_id, unsigned *crt_order_flags );
< H5_DLL herr_t H5Pset_char_encoding(hid_t plist_id, H5T_cset_t encoding);
-----
> H5_DLL herr_t H5Pset_link_phase_change(hid_t plist_id, unsigned max_compact, unsigned min_dense);
> H5_DLL herr_t H5Pset_local_heap_size_hint(hid_t plist_id, size_t size_hint);
> H5_DLL herr_t H5Pset_map_iterate_hints(hid_t mapl_id, size_t key_prefetch_size, size_t key_alloc_size);
> H5_DLL herr_t H5Pget_map_iterate_hints(hid_t mapl_id, size_t *key_prefetch_size ,
>
>                                     size_t *key_alloc_size );
237,240c264,266
< H5_DLL herr_t H5Pset_nlinks(hid_t plist_id, size_t nlinks);
< H5_DLL herr_t H5Pget_nlinks(hid_t plist_id, size_t *nlinks);
< H5_DLL herr_t H5Pset_elink_prefix(hid_t plist_id, const char *prefix);
< H5_DLL ssize_t H5Pget_elink_prefix(hid_t plist_id, char *prefix, size_t size);
-----
> H5_DLL herr_t H5Pset_char_encoding(hid_t plist_id, H5T_cset_t encoding);
> H5_DLL herr_t H5Pget_elink_acc_flags(hid_t lapl_id, unsigned *flags);
> H5_DLL herr_t H5Pget_elink_cb(hid_t lapl_id, H5L_elink_traverse_t *func, void **op_data);
242c268,269
< H5_DLL herr_t H5Pset_elink_fapl(hid_t lapl_id, hid_t fapl_id);
-----
> H5_DLL ssize_t H5Pget_elink_prefix(hid_t plist_id, char *prefix, size_t size);
> H5_DLL herr_t H5Pget_nlinks(hid_t plist_id, size_t *nlinks);
244d270
< H5_DLL herr_t H5Pget_elink_acc_flags(hid_t lapl_id, unsigned *flags);
246,248c272,274
< H5_DLL herr_t H5Pget_elink_cb(hid_t lapl_id, H5L_elink_traverse_t *func, void **op_data);
< H5_DLL herr_t H5Pset_copy_object(hid_t plist_id, unsigned crt_intmd);
< H5_DLL herr_t H5Pget_copy_object(hid_t plist_id, unsigned *crt_intmd );
-----
> H5_DLL herr_t H5Pset_elink_fapl(hid_t lapl_id, hid_t fapl_id);
> H5_DLL herr_t H5Pset_elink_prefix(hid_t plist_id, const char *prefix);
> H5_DLL herr_t H5Pset_nlinks(hid_t plist_id, size_t nlinks);
251c277
< H5_DLL herr_t H5Pset_mcdt_search_cb(hid_t plist_id, H5O_mcdt_search_cb_t func, void *op_data);
-----

```

```

> H5_DLL herr_t H5Pget_copy_object(hid_t plist_id, unsigned *copy_options );
253,267c279,299
< H5_DLL herr_t H5Pregister1(hid_t cls_id, const char *name, size_t size,
<     void *def_value, H5P_prp_create_func_t prp_create,
<     H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_del, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_close_func_t prp_close);
< H5_DLL herr_t H5Pinsert1(hid_t plist_id, const char *name, size_t size,
<     void *value, H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
<     H5P_prp_delete_func_t prp_delete, H5P_prp_copy_func_t prp_copy,
<     H5P_prp_close_func_t prp_close);
< H5_DLL H5Z_filter_t H5Pget_filter1(hid_t plist_id, unsigned filter,
<     unsigned int *flags, size_t *cd_nelmts,
<     unsigned cd_values[], size_t namelen, char name[]);
< H5_DLL herr_t H5Pget_filter_by_id1(hid_t plist_id, H5Z_filter_t id,
<     unsigned int *flags, size_t *cd_nelmts,
<     unsigned cd_values[], size_t namelen, char name[]);
-----
> H5_DLL herr_t H5Pset_copy_object(hid_t plist_id, unsigned copy_options);
> H5_DLL herr_t H5Pset_mcdt_search_cb(hid_t plist_id, H5O_mcdt_search_cb_t func, void *op_data);
> H5_DLL herr_t H5Pregister1(hid_t cls_id, const char *name, size_t size, void *def_value,
>
>         H5P_prp_create_func_t prp_create, H5P_prp_set_func_t prp_set,
>         H5P_prp_get_func_t prp_get, H5P_prp_delete_func_t prp_del,
>         H5P_prp_copy_func_t prp_copy, H5P_prp_close_func_t prp_close);
> H5_DLL herr_t H5Pinsert1(hid_t plist_id, const char *name, size_t size, void *value,
>
>         H5P_prp_set_func_t prp_set, H5P_prp_get_func_t prp_get,
>         H5P_prp_delete_func_t prp_delete, H5P_prp_copy_func_t prp_copy,
>         H5P_prp_close_func_t prp_close);
> H5_DLL herr_t H5Pencode1(hid_t plist_id, void *buf, size_t *nalloc);
> H5_DLL H5Z_filter_t H5Pget_filter1(hid_t plist_id, unsigned filter, unsigned int *flags ,
>
>         size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
>         char name[]);
> H5_DLL herr_t H5Pget_filter_by_id1(hid_t plist_id, H5Z_filter_t id, unsigned int *flags ,
>
>         size_t *cd_nelmts , unsigned cd_values[] , size_t namelen,
>         char name[] );
> H5_DLL herr_t H5Pget_version(hid_t plist_id, unsigned *boot , unsigned *freelist ,
>
>         unsigned *stab , unsigned *shhdr );
> H5_DLL herr_t H5Pset_file_space(hid_t plist_id, H5F_file_space_type_t strategy, hsize_t threshold);
> H5_DLL herr_t H5Pget_file_space(hid_t plist_id, H5F_file_space_type_t *strategy, hsize_t *threshold);
268a301
>
改变 H5PTpublic.h
2,3c2,3
< H5_HLDLL hid_t H5PTcreate(hid_t loc_id, const char *dset_name,

```

```

<          hid_t dtype_id, hsize_t chunk_size, hid_t plist_id);
-----
> H5_HLDLL hid_t H5PTcreate(hid_t loc_id, const char *dset_name, hid_t dtype_id, hsize_t chunk_size,
>          hid_t plist_id);
6,7c6,7
< H5_HLDLL hid_t H5PTcreate_fl(hid_t loc_id, const char *dset_name,
<          hid_t dtype_id, hsize_t chunk_size, int compression);
-----
> H5_HLDLL hid_t H5PTcreate_fl(hid_t loc_id, const char *dset_name, hid_t dtype_id, hsize_t chunk_size,
>          int compression);
9,11c9,10
< H5_HLDLL herr_t H5PTget_next(hid_t table_id, size_t nrecords, void * data);
< H5_HLDLL herr_t H5PTread_packets(hid_t table_id, hsize_t start,
<          size_t nrecords, void *data);
-----
> H5_HLDLL herr_t H5PTget_next(hid_t table_id, size_t nrecords, void *data);
> H5_HLDLL herr_t H5PTread_packets(hid_t table_id, hsize_t start, size_t nrecords, void *data);
17,24c16,19
< H5_HLDLL herr_t  H5PTcreate_index( hid_t table_id );
< H5_HLDLL herr_t  H5PTset_index( hid_t table_id,
<          hsize_t pt_index );
< H5_HLDLL herr_t  H5PTget_index( hid_t table_id,
<          hsize_t *pt_index );
< H5_HLDLL herr_t  H5PTfree_vlen_buff( hid_t table_id,
<          size_t buflen,
<          void * buff );
-----
> H5_HLDLL herr_t H5PTcreate_index(hid_t table_id);
> H5_HLDLL herr_t H5PTset_index(hid_t table_id, hsize_t pt_index);
> H5_HLDLL herr_t H5PTget_index(hid_t table_id, hsize_t *pt_index);
> H5_HLDLL herr_t H5PTfree_vlen_buff(hid_t table_id, size_t buflen, void *buff);
改变 H5public.h
2,4c2,4
<          H5_GCC_DIAG_OFF(long-long)
< H5_GCC_DIAG_ON(long-long)
<
H5_DLL herr_t H5open(void);
-----
>  H5_GCC_DIAG_OFF("long-long")
> H5_GCC_DIAG_ON("long-long")
>  H5_DLL herr_t H5open(void);
8,14c8,13
< H5_DLL herr_t H5set_free_list_limits (int reg_global_lim, int reg_list_lim,
<          int arr_global_lim, int arr_list_lim, int blk_global_lim,

```

```

<                int blk_list_lim);
< H5_DLL herr_t H5get_libversion(unsigned *majnum, unsigned *minnum,
<                unsigned *relnum);
< H5_DLL herr_t H5check_version(unsigned majnum, unsigned minnum,
<                unsigned relnum);
____
> H5_DLL herr_t H5set_free_list_limits(int reg_global_lim, int reg_list_lim, int arr_global_lim,
>                int arr_list_lim, int blk_global_lim, int blk_list_lim);
> H5_DLL herr_t H5get_free_list_sizes(size_t *reg_size, size_t *arr_size, size_t *blk_size, size_t *fac_size);
> H5_DLL herr_t H5get_alloc_stats(H5_alloc_stats_t *stats);
> H5_DLL herr_t H5get_libversion(unsigned *majnum, unsigned *minnum, unsigned *relnum);
> H5_DLL herr_t H5check_version(unsigned majnum, unsigned minnum, unsigned relnum);
19a19
>
改变 H5Rpublic.h
1d0
< typedef haddr_t hobj_ref_t;
3,5c2,22
< H5_DLL herr_t H5Rcreate(void *ref, hid_t loc_id, const char *name,
<                H5R_type_t ref_type, hid_t space_id);
< H5_DLL hid_t H5Rdereference(hid_t dataset, H5R_type_t ref_type, const void *ref);
____
> H5_DLL herr_t H5Rcreate_object(hid_t loc_id, const char *name, hid_t oapl_id, H5R_ref_t *ref_ptr);
> H5_DLL herr_t H5Rcreate_region(hid_t loc_id, const char *name, hid_t space_id, hid_t oapl_id,
>                H5R_ref_t *ref_ptr);
> H5_DLL herr_t H5Rcreate_attr(hid_t loc_id, const char *name, const char *attr_name, hid_t oapl_id,
>                H5R_ref_t *ref_ptr);
> H5_DLL herr_t H5Rdestroy(H5R_ref_t *ref_ptr);
> H5_DLL H5R_type_t H5Rget_type(const H5R_ref_t *ref_ptr);
> H5_DLL htri_t H5Requal(const H5R_ref_t *ref1_ptr, const H5R_ref_t *ref2_ptr);
> H5_DLL herr_t H5Rcopy(const H5R_ref_t *src_ref_ptr, H5R_ref_t *dst_ref_ptr);
> H5_DLL hid_t H5Ropen_object(H5R_ref_t *ref_ptr, hid_t rapl_id, hid_t oapl_id);
> H5_DLL hid_t H5Ropen_region(H5R_ref_t *ref_ptr, hid_t rapl_id, hid_t oapl_id);
> H5_DLL hid_t H5Ropen_attr(H5R_ref_t *ref_ptr, hid_t rapl_id, hid_t aapl_id);
> H5_DLL herr_t H5Rget_obj_type3(H5R_ref_t *ref_ptr, hid_t rapl_id, H5O_type_t *obj_type);
> H5_DLL ssize_t H5Rget_file_name(const H5R_ref_t *ref_ptr, char *name, size_t size);
> H5_DLL ssize_t H5Rget_obj_name(H5R_ref_t *ref_ptr, hid_t rapl_id, char *name, size_t size);
> H5_DLL ssize_t H5Rget_attr_name(const H5R_ref_t *ref_ptr, char *name, size_t size);
> H5_DLL H5G_obj_t H5Rget_obj_type1(hid_t id, H5R_type_t ref_type, const void *ref);
> H5_DLL hid_t H5Rdereference1(hid_t obj_id, H5R_type_t ref_type, const void *ref);
> H5_DLL herr_t H5Rcreate(void *ref, hid_t loc_id, const char *name, H5R_type_t ref_type, hid_t space_id);
> H5_DLL herr_t H5Rget_obj_type2(hid_t id, H5R_type_t ref_type, const void *ref, H5O_type_t *obj_type);
> H5_DLL hid_t H5Rdereference2(hid_t obj_id, hid_t oapl_id, H5R_type_t ref_type, const void *ref);
7,11c24

```

```

< H5_DLL herr_t H5Rget_obj_type2(hid_t id, H5R_type_t ref_type, const void *_ref,
<     H5O_type_t *obj_type);
< H5_DLL ssize_t H5Rget_name(hid_t loc_id, H5R_type_t ref_type, const void *ref,
<     char *name, size_t size);
< H5_DLL H5G_obj_t H5Rget_obj_type1(hid_t id, H5R_type_t ref_type, const void *_ref);
----
> H5_DLL ssize_t H5Rget_name(hid_t loc_id, H5R_type_t ref_type, const void *ref, char *name, size_t size);
12a26
>
改变 H5Spublic.h
2,8d1
< H5_DLL hid_t H5Screate(H5S_class_t type);
< H5_DLL hid_t H5Screate_simple(int rank, const hsize_t dims[],
<     const hsize_t maxdims[]);
< H5_DLL herr_t H5Sset_extent_simple(hid_t space_id, int rank,
<     const hsize_t dims[],
<     const hsize_t max[]);
< H5_DLL hid_t H5Scopy(hid_t space_id);
10c3,8
< H5_DLL herr_t H5Sencode(hid_t obj_id, void *buf, size_t *nalloc);
----
> H5_DLL hid_t H5Scombine_hyperslab(hid_t space_id, H5S_seloper_t op, const hsize_t start[],
>     const hsize_t stride[], const hsize_t count[], const hsize_t block[]);
> H5_DLL hid_t H5Scombine_select(hid_t space1_id, H5S_seloper_t op, hid_t space2_id);
> H5_DLL hid_t H5Scopy(hid_t space_id);
> H5_DLL hid_t H5Screate(H5S_class_t type);
> H5_DLL hid_t H5Screate_simple(int rank, const hsize_t dims[], const hsize_t maxdims[]);
12,16c10,21
< H5_DLL hssize_t H5Sget_simple_extent_npoints(hid_t space_id);
< H5_DLL int H5Sget_simple_extent_ndims(hid_t space_id);
< H5_DLL int H5Sget_simple_extent_dims(hid_t space_id, hsize_t dims[],
<     hsize_t maxdims[]);
< H5_DLL htri_t H5Sis_simple(hid_t space_id);
----
> H5_DLL herr_t H5Sencode2(hid_t obj_id, void *buf, size_t *nalloc, hid_t fapl);
> H5_DLL herr_t H5Sextent_copy(hid_t dst_id, hid_t src_id);
> H5_DLL htri_t H5Sextent_equal(hid_t space1_id, hid_t space2_id);
> H5_DLL htri_t H5Sget_regular_hyperslab(hid_t spaceid, hsize_t start[], hsize_t stride[], hsize_t count[],
>     hsize_t block[]);
> H5_DLL herr_t H5Sget_select_bounds(hid_t spaceid, hsize_t start[], hsize_t end[]);
> H5_DLL hssize_t H5Sget_select_elem_npoints(hid_t spaceid);
> H5_DLL herr_t H5Sget_select_elem_pointlist(hid_t spaceid, hsize_t startpoint, hsize_t numpoints,
>     hsize_t buf[]);
> H5_DLL herr_t H5Sget_select_hyper_blocklist(hid_t spaceid, hsize_t startblock, hsize_t numblocks,

```

```

>                                     hsize_t buf[]);
> H5_DLL hssize_t H5Sget_select_hyper_nblocks(hid_t spaceid);
18,33c23,26
< H5_DLL herr_t H5Sselect_hyperslab(hid_t space_id, H5S_seloper_t op,
<                                     const hsize_t start[],
<                                     const hsize_t _stride[],
<                                     const hsize_t count[],
<                                     const hsize_t _block[]);
< H5_DLL hid_t H5Scombine_hyperslab(hid_t space_id, H5S_seloper_t op,
<                                     const hsize_t start[],
<                                     const hsize_t _stride[],
<                                     const hsize_t count[],
<                                     const hsize_t _block[]);
< H5_DLL herr_t H5Sselect_select(hid_t space1_id, H5S_seloper_t op,
<                                     hid_t space2_id);
< H5_DLL hid_t H5Scombine_select(hid_t space1_id, H5S_seloper_t op,
<                                     hid_t space2_id);
< H5_DLL herr_t H5Sselect_elements(hid_t space_id, H5S_seloper_t op,
<     size_t num_elem, const hsize_t *coord);
----
> H5_DLL H5S_sel_type H5Sget_select_type(hid_t spaceid);
> H5_DLL int H5Sget_simple_extent_dims(hid_t space_id, hsize_t dims[], hsize_t maxdims[]);
> H5_DLL int H5Sget_simple_extent_ndims(hid_t space_id);
> H5_DLL hssize_t H5Sget_simple_extent_npoints(hid_t space_id);
35,37c28,37
< H5_DLL herr_t H5Sset_extent_none(hid_t space_id);
< H5_DLL herr_t H5Sextent_copy(hid_t dst_id, hid_t src_id);
< H5_DLL htri_t H5Sextent_equal(hid_t sid1, hid_t sid2);
----
> H5_DLL htri_t H5Sis_regular_hyperslab(hid_t spaceid);
> H5_DLL htri_t H5Sis_simple(hid_t space_id);
> H5_DLL herr_t H5Smodify_select(hid_t space1_id, H5S_seloper_t op, hid_t space2_id);
> H5_DLL herr_t H5Soffset_simple(hid_t space_id, const hssize_t *offset);
> H5_DLL herr_t H5Ssel_iter_close(hid_t sel_iter_id);
> H5_DLL hid_t H5Ssel_iter_create(hid_t spaceid, size_t elmt_size, unsigned flags);
> H5_DLL herr_t H5Ssel_iter_get_seq_list(hid_t sel_iter_id, size_t maxseq, size_t maxbytes, size_t *nseq,
>                                     size_t *nbytes, hsize_t *off, size_t *len);
> H5_DLL herr_t H5Ssel_iter_reset(hid_t sel_iter_id, hid_t space_id);
> H5_DLL herr_t H5Sselect_adjust(hid_t spaceid, const hssize_t *offset);
38a39,43
> H5_DLL herr_t H5Sselect_copy(hid_t dst_id, hid_t src_id);
> H5_DLL herr_t H5Sselect_elements(hid_t space_id, H5S_seloper_t op, size_t num_elem, const hsize_t *coord);
> H5_DLL herr_t H5Sselect_hyperslab(hid_t space_id, H5S_seloper_t op, const hsize_t start[],
>                                     const hsize_t stride[], const hsize_t count[], const hsize_t block[]);

```

[illegible]

```

<                                const void *buf );
< H5_HLDLL herr_t  H5TBwrite_records( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t type_size,
<                                const size_t *field_offset,
<                                const size_t *dst_sizes,
<                                const void *buf );
< H5_HLDLL herr_t  H5TBwrite_fields_name( hid_t loc_id,
<                                const char *dset_name,
<                                const char *field_names,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t type_size,
<                                const size_t *field_offset,
<                                const size_t *dst_sizes,
<                                const void *buf );
< H5_HLDLL herr_t  H5TBwrite_fields_index( hid_t loc_id,
<                                const char *dset_name,
<                                hsize_t nfields,
<                                const int *field_index,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t type_size,
<                                const size_t *field_offset,
<                                const size_t *dst_sizes,
<                                const void *buf );
< H5_HLDLL herr_t  H5TBread_table( hid_t loc_id,
<                                const char *dset_name,
<                                size_t dst_size,
<                                const size_t *dst_offset,
<                                const size_t *dst_sizes,
<                                void *dst_buf );
< H5_HLDLL herr_t  H5TBread_fields_name( hid_t loc_id,
<                                const char *dset_name,
<                                const char *field_names,
<                                hsize_t start,
<                                hsize_t nrecords,
<                                size_t type_size,
<                                const size_t *field_offset,
<                                const size_t *dst_sizes,
<                                void *buf );
< H5_HLDLL herr_t  H5TBread_fields_index( hid_t loc_id,

```



```

<         const char *dset_name,
<         hsize_t nfields,
<         const int *field_index,
<         hsize_t start,
<         hsize_t nrecords,
<         size_t type_size,
<         const size_t *field_offset,
<         const size_t *dst_sizes,
<         void *buf );
<
< H5_HLDLL herr_t  H5TBread_records( hid_t loc_id,
<         const char *dset_name,
<         hsize_t start,
<         hsize_t nrecords,
<         size_t type_size,
<         const size_t *dst_offset,
<         const size_t *dst_sizes,
<         void *buf );
<
< H5_HLDLL herr_t  H5TBget_table_info ( hid_t loc_id,
<         const char *dset_name,
<         hsize_t *nfields,
<         hsize_t *nrecords );
<
< H5_HLDLL herr_t  H5TBget_field_info( hid_t loc_id,
<         const char *dset_name,
<         char *field_names[],
<         size_t *field_sizes,
<         size_t *field_offsets,
<         size_t *type_size );
<
< H5_HLDLL herr_t  H5TBdelete_record( hid_t loc_id,
<         const char *dset_name,
<         hsize_t start,
<         hsize_t nrecords );
<
< H5_HLDLL herr_t  H5TBinsert_record( hid_t loc_id,
<         const char *dset_name,
<         hsize_t start,
<         hsize_t nrecords,
<         size_t dst_size,
<         const size_t *dst_offset,
<         const size_t *dst_sizes,
<         void *buf );
<
< H5_HLDLL herr_t  H5TBadd_records_from( hid_t loc_id,
<         const char *dset_name1,
<         hsize_t start1,
<         hsize_t nrecords,
<         const char *dset_name2,

```

```

        hsize_t start2);
< H5_HLDLL herr_t H5TBcombine_tables( hid_t loc_id1,
<
        const char *dset_name1,
<
        hid_t loc_id2,
<
        const char *dset_name2,
<
        const char *dset_name3 );
< H5_HLDLL herr_t H5TBinsert_field( hid_t loc_id,
<
        const char *dset_name,
<
        const char *field_name,
<
        hid_t field_type,
<
        hsize_t position,
<
        const void *fill_data,
<
        const void *buf );
< H5_HLDLL herr_t H5TBdelete_field( hid_t loc_id,
<
        const char *dset_name,
<
        const char *field_name );
< H5_HLDLL herr_t H5TBaget_title( hid_t loc_id,
<
        char *table_title );
< H5_HLDLL htri_t H5TBaget_fill(hid_t loc_id,
<
        const char *dset_name,
<
        hid_t dset_id,
<
        unsigned char *dst_buf);
____

> H5_HLDLL herr_t H5TBmake_table(const char *table_title, hid_t loc_id, const char *dset_name, hsize_t nfields,
>
        hsize_t nrecords, size_t type_size, const char *field_names[],
>
        const size_t *field_offset, const hid_t *field_types, hsize_t chunk_size,
>
        void *fill_data, int compress, const void *buf);
> H5_HLDLL herr_t H5TBappend_records(hid_t loc_id, const char *dset_name, hsize_t nrecords, size_t type_size,
>
        const size_t *field_offset, const size_t *dst_sizes, const void *buf);
> H5_HLDLL herr_t H5TBwrite_records(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords,
>
        size_t type_size, const size_t *field_offset, const size_t *dst_sizes,
>
        const void *buf);
> H5_HLDLL herr_t H5TBwrite_fields_name(hid_t loc_id, const char *dset_name, const char *field_names,
>
        hsize_t start, hsize_t nrecords, size_t type_size,
>
        const size_t *field_offset, const size_t *dst_sizes, const void *buf);
> H5_HLDLL herr_t H5TBwrite_fields_index(hid_t loc_id, const char *dset_name, hsize_t nfields,
>
        const int *field_index, hsize_t start, hsize_t nrecords,
>
        size_t type_size, const size_t *field_offset, const size_t
*dst_sizes,
>
        const void *buf);
> H5_HLDLL herr_t H5TBread_table(hid_t loc_id, const char *dset_name, size_t dst_size, const size_t *dst_offset,
>
        const size_t *dst_sizes, void *dst_buf);
> H5_HLDLL herr_t H5TBread_fields_name(hid_t loc_id, const char *dset_name, const char *field_names,
>
        hsize_t start, hsize_t nrecords, size_t type_size,

```

```

>                                     const size_t *field_offset, const size_t *dst_sizes, void *buf);
> H5_HLDLL herr_t H5TBread_fields_index(hid_t loc_id, const char *dset_name, hsize_t nfields,
>                                     const int *field_index, hsize_t start, hsize_t nrecords,
>                                     size_t type_size, const size_t *field_offset, const size_t *dst_sizes,
>                                     void *buf);
> H5_HLDLL herr_t H5TBread_records(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords,
>                                   size_t type_size, const size_t *dst_offset, const size_t *dst_sizes,
>                                   void *buf);
> H5_HLDLL herr_t H5TBget_table_info(hid_t loc_id, const char *dset_name, hsize_t *nfields, hsize_t *nrecords);
> H5_HLDLL herr_t H5TBget_field_info(hid_t loc_id, const char *dset_name, char *field_names[],
>                                   size_t *field_sizes, size_t *field_offsets, size_t *type_size);
> H5_HLDLL herr_t H5TBdelete_record(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords);
> H5_HLDLL herr_t H5TBinsert_record(hid_t loc_id, const char *dset_name, hsize_t start, hsize_t nrecords,
>                                   size_t dst_size, const size_t *dst_offset, const size_t *dst_sizes,
>                                   void *buf);
> H5_HLDLL herr_t H5TBadd_records_from(hid_t loc_id, const char *dset_name1, hsize_t start1, hsize_t nrecords,
>                                       const char *dset_name2, hsize_t start2);
> H5_HLDLL herr_t H5TBcombine_tables(hid_t loc_id1, const char *dset_name1, hid_t loc_id2,
>                                    const char *dset_name2, const char *dset_name3);
> H5_HLDLL herr_t H5TBinsert_field(hid_t loc_id, const char *dset_name, const char *field_name,
>                                  hid_t field_type, hsize_t position, const void *fill_data, const void *buf);
> H5_HLDLL herr_t H5TBdelete_field(hid_t loc_id, const char *dset_name, const char *field_name);
> H5_HLDLL herr_t H5TBaget_title(hid_t loc_id, char *table_title);
> H5_HLDLL htri_t H5TBaget_fill(hid_t loc_id, const char *dset_name, hid_t dset_id, unsigned char *dst_buf);
改变 H5Tpublic.h
31a32
> H5_DLLVAR hid_t H5T_STD_REF_g;
92,93c93,94
< H5_DLL herr_t H5Tcommit2(hid_t loc_id, const char *name, hid_t type_id,
<     hid_t lcpl_id, hid_t tcpl_id, hid_t tapl_id);
---
> H5_DLL herr_t H5Tcommit2(hid_t loc_id, const char *name, hid_t type_id, hid_t lcpl_id, hid_t tcpl_id,
>                           hid_t tapl_id);
100,101c101,103
< H5_DLL herr_t H5Tinsert(hid_t parent_id, const char *name, size_t offset,
<     hid_t member_id);
---
> H5_DLL herr_t H5Tflush(hid_t type_id);
> H5_DLL herr_t H5Trefresh(hid_t type_id);
> H5_DLL herr_t H5Tinsert(hid_t parent_id, const char *name, size_t offset, hid_t member_id);
105,108c107,108
< H5_DLL herr_t H5Tenum_nameof(hid_t type, const void *value, char *name,
<     size_t size);
< H5_DLL herr_t H5Tenum_valueof(hid_t type, const char *name,

```

```

<                                void *value);
____

> H5_DLL herr_t H5Tenum_nameof(hid_t type, const void *value, char *name , size_t size);
> H5_DLL herr_t H5Tenum_valueof(hid_t type, const char *name, void *value );
110,111c110

< H5_DLL hid_t H5Tarray_create2(hid_t base_id, unsigned ndims,
<                                const hsize_t dim[]);
____

> H5_DLL hid_t H5Tarray_create2(hid_t base_id, unsigned ndims, const hsize_t dim[]);
123,124c122

< H5_DLL herr_t H5Tget_pad(hid_t type_id, H5T_pad_t *lsb,
<                                H5T_pad_t *msb);
____

> H5_DLL herr_t H5Tget_pad(hid_t type_id, H5T_pad_t *lsb , H5T_pad_t *msb );
126,128c124,125

< H5_DLL herr_t H5Tget_fields(hid_t type_id, size_t *spos,
<                                size_t *epos, size_t *esize,
<                                size_t *mpos, size_t *msize);
____

> H5_DLL herr_t H5Tget_fields(hid_t type_id, size_t *spos , size_t *epos , size_t *esize ,
>                                size_t *mpos , size_t *msize );
139c136

< H5_DLL herr_t H5Tget_member_value(hid_t type_id, unsigned membno, void *value);
____

> H5_DLL herr_t H5Tget_member_value(hid_t type_id, unsigned membno, void *value );
149,150c146

< H5_DLL herr_t H5Tset_fields(hid_t type_id, size_t spos, size_t epos,
<                                size_t esize, size_t mpos, size_t msize);
____

> H5_DLL herr_t H5Tset_fields(hid_t type_id, size_t spos, size_t epos, size_t esize, size_t mpos, size_t msize);
156,159c152,153

< H5_DLL herr_t H5Tregister(H5T_pers_t pers, const char *name, hid_t src_id,
<                                hid_t dst_id, H5T_conv_t func);
< H5_DLL herr_t H5Tunregister(H5T_pers_t pers, const char *name, hid_t src_id,
<                                hid_t dst_id, H5T_conv_t func);
____

> H5_DLL herr_t H5Tregister(H5T_pers_t pers, const char *name, hid_t src_id, hid_t dst_id, H5T_conv_t func);
> H5_DLL herr_t H5Tunregister(H5T_pers_t pers, const char *name, hid_t src_id, hid_t dst_id, H5T_conv_t func);
162,163c156,158

< H5_DLL herr_t H5Tconvert(hid_t src_id, hid_t dst_id, size_t nelmts,
<                                void *buf, void *background, hid_t plist_id);
____

> H5_DLL herr_t H5Tconvert(hid_t src_id, hid_t dst_id, size_t nelmts, void *buf, void *background,
>                                hid_t plist_id);

```

```

> H5_DLL herr_t H5Ttreclaim(hid_t type_id, hid_t space_id, hid_t plist_id, void *buf);
166,168c161,162
< H5_DLL hid_t H5Tarray_create1(hid_t base_id, int ndims,
<
const hsize_t dim[],
<
const int perm[]);
___
> H5_DLL hid_t H5Tarray_create1(hid_t base_id, int ndims, const hsize_t dim[],
>
const int perm[]);
170a165
>
增加头文件: H5VLconnector.h
增加头文件: H5VLconnector_passthru.h
增加头文件: H5VLnative.h
增加头文件: H5VLpassthru.h
增加头文件: H5VLpublic.h
改变 hdf5_hl.h
0a1
>

```

libpcap-devel

```

改变 bpf.h
2,3c2,9
< PCAP_API int bpf_validate(const struct bpf_insn *, int);
< PCAP_API u_int bpf_filter(const struct bpf_insn *, const u_char *, u_int, u_int);
___
> PCAP_AVAILABLE_0_4
> PCAP_API u_int bpf_filter(const struct bpf_insn *, const u_char *, u_int, u_int);
> PCAP_AVAILABLE_0_6
> PCAP_API int bpf_validate(const struct bpf_insn *, int len);
> PCAP_AVAILABLE_0_4
> PCAP_API char *bpf_image(const struct bpf_insn *, int);
> PCAP_AVAILABLE_0_6
> PCAP_API void bpf_dump(const struct bpf_program *, int);
改变 namedb.h
5c5,7
< PCAP_API bpf_u_int32 **pcap_nametoaddr(const char *);
___
> PCAP_API
> PCAP_DEPRECATED("this is not reentrant; use 'pcap_nametoaddrinfo' instead")
> bpf_u_int32 **pcap_nametoaddr(const char *);
改变 socket.h
1c1
<

```

>

efivar-devel

增加头文件: .abignore

改变 efiboot-creator.h

25d24

<

改变 efiboot.h

1c1,2

<

> extern uint32_t efi_get_libefiboot_version(void)

> __attribute__((__visibility__("default")));

改变 efiboot-loadopt.h

47d46

<

增加头文件: efisec.h

增加头文件: efisec-secdb.h

增加头文件: efisec-types.h

改变 efivar-dp.h

9,11c9,11

< uint32_t hid, uint32_t uid, uint32_t cid,
< const char *hidstr, const char *uidstr,
< const char *cidstr);

> uint32_t hid, uint32_t uid, uint32_t cid,
> const char *hidstr, const char *uidstr,
> const char *cidstr);

190,193c190,195

< extern ssize_t efidp_parse_device_node(char *path, efidp out, size_t size);
< extern ssize_t efidp_parse_device_path(char *path, efidp out, size_t size);
< extern ssize_t efidp_format_device_path(char *buf, size_t size, const_efidp dp,
< ssize_t limit);

> extern ssize_t efidp_parse_device_node(unsigned char *path,
> efidp out, size_t size);
> extern ssize_t efidp_parse_device_path(unsigned char *path,
> efidp out, size_t size);
> extern ssize_t efidp_format_device_path(unsigned char *buf, size_t size,
> const_efidp dp, ssize_t limit);

199d200

<

改变 efivar-guids.h

3c3

```
< extern const efi_guid_t efi_guid_sha512 __attribute__((__visibility__ ("default")));
```

```
> extern const efi_guid_t efi_guid_redhat_2 __attribute__((__visibility__ ("default")));
```

4a5

```
> extern const efi_guid_t efi_guid_sha512 __attribute__((__visibility__ ("default")));
```

7a9,10

```
> extern const efi_guid_t efi_guid_supermicro __attribute__((__visibility__ ("default")));
```

```
> extern const efi_guid_t efi_guid_asus __attribute__((__visibility__ ("default")));
```

15a19

```
> extern const efi_guid_t efi_guid_fives __attribute__((__visibility__ ("default")));
```

18a23,24

```
> extern const efi_guid_t efi_guid_canonical __attribute__((__visibility__ ("default")));
```

```
> extern const efi_guid_t efi_guid_dell __attribute__((__visibility__ ("default")));
```

22a29

```
> extern const efi_guid_t efi_guid_auto_created_boot_option __attribute__((__visibility__ ("default")));
```

25a33

```
> extern const efi_guid_t efi_guid_grub __attribute__((__visibility__ ("default")));
```

29d36

```
< extern const efi_guid_t efi_guid_redhat_2 __attribute__((__visibility__ ("default")));
```

35d41

```
< extern const efi_guid_t efi_guid_sha384 __attribute__((__visibility__ ("default")));
```

36a43,61

```
> extern const efi_guid_t efi_guid_sha384 __attribute__((__visibility__ ("default")));
```

```
> extern const struct efivar_guidname
```

```
>     __attribute__((__visibility__ ("default")))
```

```
>     efi_well_known_guids[41];
```

```
> extern const struct efivar_guidname
```

```
>     __attribute__((__visibility__ ("default")))
```

```
>     efi_well_known_guids_end;
```

```
> extern const uint64_t
```

```
>     __attribute__((__visibility__ ("default")))
```

```
>     efi_n_well_known_guids;
```

```
> extern const struct efivar_guidname
```

```
>     __attribute__((__visibility__ ("default")))
```

```
>     efi_well_known_names[41];
```

```
> extern const struct efivar_guidname
```

```
>     __attribute__((__visibility__ ("default")))
```

```
>     efi_well_known_names_end;
```

```
> extern const uint64_t
```

```
>     __attribute__((__visibility__ ("default")))
```

```
>     efi_n_well_known_names;
```

改变 efivar.h

```

51a52,54
> extern ssize_t efi_variable_export_dmpstore(efi_variable_t *var, uint8_t *data,
>
>         size_t size)
>
>         __attribute__((__nonnull__(1)));
55c58
< extern int efi_variable_set_name(efi_variable_t *var, char *name)
____

> extern int efi_variable_set_name(efi_variable_t *var, unsigned char *name)
57c60
< extern char *efi_variable_get_name(efi_variable_t *var)
____

> extern unsigned char *efi_variable_get_name(efi_variable_t *var)
91a95,96
> extern void efi_error_pop(void);
> extern void efi_set_loglevel(int level);
97a103,106
> static inline void
>
>
> static inline void
>
>
99c108
<
__attribute__((__visibility__("default")));
____

>
__attribute__((__visibility__("default")));
101c110
<
__attribute__((__visibility__("default")));
____

>
__attribute__((__visibility__("default")));
103,104c112,114
<
__attribute__((__visibility__("default")));;
<
____

>
__attribute__((__visibility__("default")));;
> extern uint32_t efi_get_libefivar_version(void)
>
__attribute__((__visibility__("default")));;
增加头文件: efivar-time.h
增加头文件: efivar-types.h

```

ImageMagick-devel

改变 exception.h

```

28c28,29
<
MagickFatalError(const ExceptionType,const char *,const char *),
____

```



```

>   MagickFatalError(const ExceptionType,const char *,const char *)
>   magick_attribute((__noreturn__)),
改变 log.h
2a3,4
> extern const MagickExport LogInfo
>   **GetLogInfoList(const char *,size_t *,ExceptionInfo *);
5,6c7,8
< extern MagickExport const char
<   *GetLogName(void),
---
> extern MagickExport char
>   *GetLogName(void) magick_attribute((__pure__)),
8,9c10,11
< extern MagickExport const LogInfo
<   **GetLogInfoList(const char *,size_t *,ExceptionInfo *);
---
> extern MagickExport LogEventType
>   GetLogEventMask(void) magick_attribute((__pure__));
改变 memory_.h
39c39
<   if (extent != (size_t *const) NULL)
---
>   if (extent != (size_t *) NULL)
改变 pixel-accessor.h
8c8
< static inline MagickRealType GetPixelLuma(const Image *magick_restrict image,
---
> static inline MagickRealType GetPixelLuma(
16a17,18
> static inline void GetPixelPacketRGBA(const Quantum red,const Quantum green,
>
21c23
< static inline MagickBooleanType IsPixelMonochrome(const PixelPacket *pixel)
---
> static inline MagickBooleanType IsPixelMonochrome(

```

systemd-devel

```

改变 sd-event.h
60a61,63
> int sd_event_source_set_ratelimit(sd_event_source *s, uint64_t interval_usec, unsigned burst);
> int sd_event_source_get_ratelimit(sd_event_source *s, uint64_t *ret_interval_usec, unsigned *ret_burst);
> int sd_event_source_is_ratelimited(sd_event_source *s);

```

python3-pillow-devel

增加头文件: Bcn.h

改变 Imaging.h

3,63c3,403

```
< extern int ImagingMemorySetBlocksMax(ImagingMemoryArena arena, int blocks_max);
< extern void ImagingMemoryClearCache(ImagingMemoryArena arena, int new_size);
< extern Imaging ImagingNew(const char* mode, int xsize, int ysize);
< extern Imaging ImagingNewDirty(const char* mode, int xsize, int ysize);
< extern Imaging ImagingNew2Dirty(const char* mode, Imaging imOut, Imaging imIn);
< extern void      ImagingDelete(Imaging im);
< extern Imaging ImagingNewBlock(const char* mode, int xsize, int ysize);
< extern Imaging ImagingNewPrologue(const char *mode,
<
<                                int xsize, int ysize);
< extern Imaging ImagingNewPrologueSubtype(const char *mode,
<
<                                int xsize, int ysize,
<                                int structure_size);
< extern void ImagingCopyPalette(Imaging destination, Imaging source);
< extern void ImagingHistogramDelete(ImagingHistogram histogram);
< extern void ImagingAccessInit(void);
< extern ImagingAccess ImagingAccessNew(Imaging im);
< extern void _ImagingAccessDelete(Imaging im, ImagingAccess access);
< extern ImagingPalette ImagingPaletteNew(const char *mode);
< extern ImagingPalette ImagingPaletteNewBrowser(void);
< extern ImagingPalette ImagingPaletteDuplicate(ImagingPalette palette);
< extern void      ImagingPaletteDelete(ImagingPalette palette);
< extern int  ImagingPaletteCachePrepare(ImagingPalette palette);
< extern void ImagingPaletteCacheUpdate(ImagingPalette palette,
<
<                                int r, int g, int b);
< extern void ImagingPaletteCacheDelete(ImagingPalette palette);
< extern Imaging ImagingQuantize(Imaging im, int colours, int mode, int kmeans);
< extern void ImagingSectionEnter(ImagingSectionCookie* cookie);
< extern void ImagingSectionLeave(ImagingSectionCookie* cookie);
< extern void* ImagingError_OSError(void);
< extern void* ImagingError_MemoryError(void);
< extern void* ImagingError_ModeError(void);
< extern void* ImagingError_Mismatch(void);
< extern void* ImagingError_ValueError(const char* message);
< extern void ImagingError_Clear(void);
< extern Imaging ImagingAlphaComposite(Imaging imIn1, Imaging imIn2);
< extern Imaging ImagingBlend(Imaging imIn1, Imaging imIn2, float alpha);
< extern Imaging ImagingCopy(Imaging im);
< extern Imaging ImagingConvert(Imaging im, const char* mode, ImagingPalette palette, int dither);
< extern Imaging ImagingConvertInPlace(Imaging im, const char* mode);
```

```

< extern Imaging ImagingConvertMatrix(Imaging im, const char *mode, float m[]);
< extern Imaging ImagingConvertTransparent(Imaging im, const char *mode, int r, int g, int b);
< extern Imaging ImagingCrop(Imaging im, int x0, int y0, int x1, int y1);
< extern Imaging ImagingExpand(Imaging im, int x, int y, int mode);
< extern Imaging ImagingFill(Imaging im, const void* ink);
< extern int ImagingFill2(
<     Imaging into, const void* ink, Imaging mask,
<     int x0, int y0, int x1, int y1);
< extern Imaging ImagingFillBand(Imaging im, int band, int color);
< extern Imaging ImagingFillLinearGradient(const char* mode);
< extern Imaging ImagingFillRadialGradient(const char* mode);
< extern Imaging ImagingFilter(
<     Imaging im, int xsize, int ysize, const FLOAT32* kernel,
<     FLOAT32 offset);
< extern Imaging ImagingFlipLeftRight(Imaging imOut, Imaging imIn);
< extern Imaging ImagingFlipTopBottom(Imaging imOut, Imaging imIn);
< extern Imaging ImagingGaussianBlur(Imaging imOut, Imaging imIn, float radius,
<                                     int passes);
< extern Imaging ImagingGetBand(Imaging im, int band);
< extern Imaging ImagingMerge(const char* mode, Imaging bands[4]);
< extern int ImagingSplit(Imaging im, Imaging bands[4]);
< extern int ImagingGetBBox(Imaging im, int bbox[4]);
——
> extern int
> ImagingMemorySetBlocksMax(ImagingMemoryArena arena, int blocks_max);
> extern void
> ImagingMemoryClearCache(ImagingMemoryArena arena, int new_size);
> extern Imaging
> ImagingNew(const char *mode, int xsize, int ysize);
> extern Imaging
> ImagingNewDirty(const char *mode, int xsize, int ysize);
> extern Imaging
> ImagingNew2Dirty(const char *mode, Imaging imOut, Imaging imIn);
> extern void
> ImagingDelete(Imaging im);
> extern Imaging
> ImagingNewBlock(const char *mode, int xsize, int ysize);
> extern Imaging
> ImagingNewPrologue(const char *mode, int xsize, int ysize);
> extern Imaging
> ImagingNewPrologueSubtype(const char *mode, int xsize, int ysize, int structure_size);
> extern void
> ImagingCopyPalette(Imaging destination, Imaging source);
> extern void

```

```
> ImagingHistogramDelete(ImagingHistogram histogram);
> extern void
> ImagingAccessInit(void);
> extern ImagingAccess
> ImagingAccessNew(Imaging im);
> extern void
> _ImagingAccessDelete(Imaging im, ImagingAccess access);
> extern ImagingPalette
> ImagingPaletteNew(const char *mode);
> extern ImagingPalette
> ImagingPaletteNewBrowser(void);
> extern ImagingPalette
> ImagingPaletteDuplicate(ImagingPalette palette);
> extern void
> ImagingPaletteDelete(ImagingPalette palette);
> extern int
> ImagingPaletteCachePrepare(ImagingPalette palette);
> extern void
> ImagingPaletteCacheUpdate(ImagingPalette palette, int r, int g, int b);
> extern void
> ImagingPaletteCacheDelete(ImagingPalette palette);
> extern Imaging
> ImagingQuantize(Imaging im, int colours, int mode, int kmeans);
> extern void
> ImagingSectionEnter(ImagingSectionCookie *cookie);
> extern void
> ImagingSectionLeave(ImagingSectionCookie *cookie);
> extern void *
> ImagingError_OSError(void);
> extern void *
> ImagingError_MemoryError(void);
> extern void *
> ImagingError_ModeError(void);
> extern void *
> ImagingError_Mismatch(void);
> extern void *
> ImagingError_ValueError(const char *message);
> extern void
> ImagingError_Clear(void);
> extern Imaging
> ImagingAlphaComposite(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingBlend(Imaging imIn1, Imaging imIn2, float alpha);
> extern Imaging
```

```

> ImagingCopy(Imaging im);
> extern Imaging
> ImagingConvert(Imaging im, const char *mode, ImagingPalette palette, int dither);
> extern Imaging
> ImagingConvertInPlace(Imaging im, const char *mode);
> extern Imaging
> ImagingConvertMatrix(Imaging im, const char *mode, float m[]);
> extern Imaging
> ImagingConvertTransparent(Imaging im, const char *mode, int r, int g, int b);
> extern Imaging
> ImagingCrop(Imaging im, int x0, int y0, int x1, int y1);
> extern Imaging
> ImagingExpand(Imaging im, int x, int y, int mode);
> extern Imaging
> ImagingFill(Imaging im, const void *ink);
> extern int
> ImagingFill2(
>     Imaging into, const void *ink, Imaging mask, int x0, int y0, int x1, int y1);
> extern Imaging
> ImagingFillBand(Imaging im, int band, int color);
> extern Imaging
> ImagingFillLinearGradient(const char *mode);
> extern Imaging
> ImagingFillRadialGradient(const char *mode);
> extern Imaging
> ImagingFilter(Imaging im, int xsize, int ysize, const FLOAT32 *kernel, FLOAT32 offset);
> extern Imaging
> ImagingFlipLeftRight(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingFlipTopBottom(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingGaussianBlur(Imaging imOut, Imaging imIn, float radius, int passes);
> extern Imaging
> ImagingGetBand(Imaging im, int band);
> extern Imaging
> ImagingMerge(const char *mode, Imaging bands[4]);
> extern int
> ImagingSplit(Imaging im, Imaging bands[4]);
> extern int
> ImagingGetBBox(Imaging im, int bbox[4]);
> extern ImagingColorItem *
> ImagingGetColors(Imaging im, int maxcolors, int *colors);
> extern int
> ImagingGetExtrema(Imaging im, void *extrema);

```

```

> extern int
> ImagingGetProjection(Imaging im, UINT8 *xproj, UINT8 *yproj);
> extern ImagingHistogram
> ImagingGetHistogram(Imaging im, Imaging mask, void *extrema);
> extern Imaging
> ImagingModeFilter(Imaging im, int size);
> extern Imaging
> ImagingNegative(Imaging im);
> extern Imaging
> ImagingOffset(Imaging im, int xoffset, int yoffset);
> extern int
> ImagingPaste(Imaging into, Imaging im, Imaging mask, int x0, int y0, int x1, int y1);
> extern Imaging
> ImagingPoint(Imaging im, const char *tablemode, const void *table);
> extern Imaging
> ImagingPointTransform(Imaging imIn, double scale, double offset);
> extern Imaging
> ImagingPutBand(Imaging im, Imaging imIn, int band);
> extern Imaging
> ImagingRankFilter(Imaging im, int size, int rank);
> extern Imaging
> ImagingRotate90(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingRotate180(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingRotate270(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingTranspose(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingTransverse(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingResample(Imaging imIn, int xsize, int ysize, int filter, float box[4]);
> extern Imaging
> ImagingReduce(Imaging imIn, int xscale, int yscale, int box[4]);
> extern Imaging
> ImagingTransform(
>     Imaging imOut,
>     Imaging imIn,
>     int method,
>     int x0,
>     int y0,
>     int x1,
>     int y1,
>     double a[8],

```

```

>     int filter,
>     int fill);
> extern Imaging
> ImagingUnsharpMask(Imaging imOut, Imaging im, float radius, int percent, int threshold);
> extern Imaging
> ImagingBoxBlur(Imaging imOut, Imaging imIn, float radius, int n);
> extern Imaging
> ImagingColorLUT3D_linear(
>     Imaging imOut,
>     Imaging imIn,
>     int table_channels,
>     int size1D,
>     int size2D,
>     int size3D,
>     INT16 *table);
> extern Imaging
> ImagingCopy2(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingConvert2(Imaging imOut, Imaging imIn);
> extern Imaging
> ImagingChopLighter(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopDarker(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopDifference(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopMultiply(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopScreen(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopAdd(Imaging imIn1, Imaging imIn2, float scale, int offset);
> extern Imaging
> ImagingChopSubtract(Imaging imIn1, Imaging imIn2, float scale, int offset);
> extern Imaging
> ImagingChopAddModulo(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopSubtractModulo(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopSoftLight(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopHardLight(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingOverlay(Imaging imIn1, Imaging imIn2);
> extern Imaging

```

```
> ImagingChopAnd(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopOr(Imaging imIn1, Imaging imIn2);
> extern Imaging
> ImagingChopXor(Imaging imIn1, Imaging imIn2);
> extern int
> ImagingDrawArc(
>     Imaging im,
>     int x0,
>     int y0,
>     int x1,
>     int y1,
>     float start,
>     float end,
>     const void *ink,
>     int width,
>     int op);
> extern int
> ImagingDrawBitmap(Imaging im, int x0, int y0, Imaging bitmap, const void *ink, int op);
> extern int
> ImagingDrawChord(
>     Imaging im,
>     int x0,
>     int y0,
>     int x1,
>     int y1,
>     float start,
>     float end,
>     const void *ink,
>     int fill,
>     int width,
>     int op);
> extern int
> ImagingDrawEllipse(
>     Imaging im,
>     int x0,
>     int y0,
>     int x1,
>     int y1,
>     const void *ink,
>     int fill,
>     int width,
>     int op);
> extern int
```



```

> ImagingDrawLine(Imaging im, int x0, int y0, int x1, int y1, const void *ink, int op);
> extern int
> ImagingDrawWideLine(
>     Imaging im, int x0, int y0, int x1, int y1, const void *ink, int width, int op);
> extern int
> ImagingDrawPieslice(
>     Imaging im,
>     int x0,
>     int y0,
>     int x1,
>     int y1,
>     float start,
>     float end,
>     const void *ink,
>     int fill,
>     int width,
>     int op);
> extern int
> ImagingDrawPoint(Imaging im, int x, int y, const void *ink, int op);
> extern int
> ImagingDrawPolygon(Imaging im, int points, int *xy, const void *ink, int fill, int width, int op);
> extern int
> ImagingDrawRectangle(
>     Imaging im,
>     int x0,
>     int y0,
>     int x1,
>     int y1,
>     const void *ink,
>     int fill,
>     int width,
>     int op);
> extern ImagingOutline
> ImagingOutlineNew(void);
> extern void
> ImagingOutlineDelete(ImagingOutline outline);
> extern int
> ImagingDrawOutline(
>     Imaging im, ImagingOutline outline, const void *ink, int fill, int op);
> extern int
> ImagingOutlineMove(ImagingOutline outline, float x, float y);
> extern int
> ImagingOutlineLine(ImagingOutline outline, float x, float y);
> extern int

```

```

> ImagingOutlineCurve(
>     ImagingOutline outline, float x1, float y1, float x2, float y2, float x3, float y3);
> extern int
> ImagingOutlineTransform(ImagingOutline outline, double a[6]);
> extern int
> ImagingOutlineClose(ImagingOutline outline);
> extern Imaging
> ImagingEffectSpread(Imaging imIn, int distance);
> extern Imaging
> ImagingEffectNoise(int xsize, int ysize, float sigma);
> extern Imaging
> ImagingEffectMandelbrot(int xsize, int ysize, double extent[4], int quality);
> extern Imaging
> ImagingOpenPPM(const char *filename);
> extern int
> ImagingSavePPM(Imaging im, const char *filename);
> extern int
> ImagingBcnDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingBitDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingEpsEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingFliDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingGifDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingGifEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingHexDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingJpegDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingJpegDecodeCleanup(ImagingCodecState state);
> extern int
> ImagingJpegUseJCSExtensions(void);
> extern int
> ImagingJpegEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingJpeg2KDecode(
>     Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingJpeg2KDecodeCleanup(ImagingCodecState state);
> extern int

```

```

> ImagingJpeg2KEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingJpeg2KEncodeCleanup(ImagingCodecState state);
> extern int
> ImagingLibTiffDecode(
>     Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingLibTiffEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingMpegDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingMspDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingPackbitsDecode(
>     Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingPcdDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingPcxDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingPcxEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingRawDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingRawEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingSgiRleDecode(
>     Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingSunRleDecode(
>     Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingTgaRleDecode(
>     Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingTgaRleEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingXbmDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingXbmEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingZipDecode(Imaging im, ImagingCodecState state, UINT8 *buffer, Py_ssize_t bytes);
> extern int
> ImagingZipDecodeCleanup(ImagingCodecState state);

```

```

> extern int
> ImagingZipEncode(Imaging im, ImagingCodecState state, UINT8 *buffer, int bytes);
> extern int
> ImagingZipEncodeCleanup(ImagingCodecState state);
> extern void
> ImagingPackBGR(UINT8 *out, const UINT8 *in, int pixels);
> extern void
> ImagingUnpackYCC(UINT8 *out, const UINT8 *in, int pixels);
> extern void
> ImagingUnpackYCCA(UINT8 *out, const UINT8 *in, int pixels);
> extern void
> ImagingConvertRGB2YCbCr(UINT8 *out, const UINT8 *in, int pixels);
> extern void
> ImagingConvertYCbCr2RGB(UINT8 *out, const UINT8 *in, int pixels);
> extern ImagingShuffler
> ImagingFindUnpacker(const char *mode, const char *rawmode, int *bits_out);
> extern ImagingShuffler
> ImagingFindPacker(const char *mode, const char *rawmode, int *bits_out);
> extern Py_ssize_t
> _imaging_read_pyFd(PyObject *fd, char *dest, Py_ssize_t bytes);
> extern Py_ssize_t
> _imaging_write_pyFd(PyObject *fd, char *src, Py_ssize_t bytes);
> extern int
> _imaging_seek_pyFd(PyObject *fd, Py_ssize_t offset, int whence);
> extern Py_ssize_t
> _imaging_tell_pyFd(PyObject *fd);
> extern UINT8 *clip8_lookups;
> }

```

改变 ImagingUtils.h

3c3

```

<    __asm__("xorps %0, %0; cvtsi2ss %1, %0" : "=x"(x) : "r"(v) );

```

```

>    __asm__("xorps %0, %0; cvtsi2ss %1, %0" : "=x"(x) : "r"(v));

```

6c6

```

< static float inline _i2f(int v) { return (float) v; }

```

```

> static float inline _i2f(int v) { return (float)v; }

```

改变 ImDib.h

2,8c2,15

```

< extern char* ImagingGetModeDIB(int size_out[2]);

```

```

< extern ImagingDIB ImagingNewDIB(const char *mode, int xsize, int ysize);

```

```

< extern void ImagingDeleteDIB(ImagingDIB im);

```

```

< extern void ImagingDrawDIB(ImagingDIB dib, void *dc, int dst[4], int src[4]);

```

```

< extern void ImagingExposeDIB(ImagingDIB dib, void *dc);

```

```
< extern int ImagingQueryPaletteDIB(ImagingDIB dib, void *dc);
< extern void ImagingPasteDIB(ImagingDIB dib, Imaging im, int xy[4]);
```

——

```
> extern char *
> ImagingGetModeDIB(int size_out[2]);
> extern ImagingDIB
> ImagingNewDIB(const char *mode, int xsize, int ysize);
> extern void
> ImagingDeleteDIB(ImagingDIB im);
> extern void
> ImagingDrawDIB(ImagingDIB dib, void *dc, int dst[4], int src[4]);
> extern void
> ImagingExposeDIB(ImagingDIB dib, void *dc);
> extern int
> ImagingQueryPaletteDIB(ImagingDIB dib, void *dc);
> extern void
> ImagingPasteDIB(ImagingDIB dib, Imaging im, int xy[4]);
```

改变 ImPlatform.h

1d0

<

改变 QuantHash.h

1,11c1,23

```
< HashTable * hashtable_new(HashFunc hf,HashCmpFunc cf);
< void hashtable_free(HashTable *h);
< void hashtable_foreach(HashTable *h,IteratorFunc i,void *u);
< void hashtable_foreach_update(HashTable *h,IteratorUpdateFunc i,void *u);
< int hashtable_insert(HashTable *h,HashKey_t key,HashVal_t val);
< int hashtable_lookup(const HashTable *h,const HashKey_t key,HashVal_t *valp);
< int hashtable_insert_or_update_computed(HashTable *h,HashKey_t key,ComputeFunc newFunc,ComputeFunc
existsFunc);
< void *hashtable_set_user_data(HashTable *h,void *data);
< void *hashtable_get_user_data(const HashTable *h);
< uint32_t hashtable_get_count(const HashTable *h);
< void hashtable_rehash_compute(HashTable *h,CollisionFunc cf);
```

——

```
> HashTable *
> hashtable_new(HashFunc hf, HashCmpFunc cf);
> void
> hashtable_free(HashTable *h);
> void
> hashtable_foreach(HashTable *h, IteratorFunc i, void *u);
> void
> hashtable_foreach_update(HashTable *h, IteratorUpdateFunc i, void *u);
> int
```

```

> hashtable_insert(HashTable *h, HashKey_t key, HashVal_t val);
> int
> hashtable_lookup(const HashTable *h, const HashKey_t key, HashVal_t *valp);
> int
> hashtable_insert_or_update_computed(
>     HashTable *h, HashKey_t key, ComputeFunc newFunc, ComputeFunc existsFunc);
> void *
> hashtable_set_user_data(HashTable *h, void *data);
> void *
> hashtable_get_user_data(const HashTable *h);
> uint32_t
> hashtable_get_count(const HashTable *h);
> void
> hashtable_rehash_compute(HashTable *h, CollisionFunc cf);

```

改变 QuantHeap.h

```

1,4c1,8
< void ImagingQuantHeapFree(Heap *);
< int ImagingQuantHeapRemove(Heap *,void **);
< int ImagingQuantHeapAdd(Heap *,void *);
< int ImagingQuantHeapTop(Heap *,void **);
---
> void
> ImagingQuantHeapFree(Heap *);
> int
> ImagingQuantHeapRemove(Heap *, void **);
> int
> ImagingQuantHeapAdd(Heap *, void *);
> int
> ImagingQuantHeapTop(Heap *, void **);

```

改变 QuantOctree.h

```

1,7c1,2
< int quantize_octree(Pixel *,
<     uint32_t,
<     uint32_t,
<     Pixel **,
<     uint32_t *,
<     uint32_t **,
<     int);
---
> int
> quantize_octree(Pixel *, uint32_t, uint32_t, Pixel **, uint32_t *, uint32_t **, int);

```

改变 QuantPngQuant.h

```

1c1,3
< int quantize_pngquant(Pixel *,

```

```

——
> int
> quantize_pngquant(
>     Pixel *,
减少头文件: raqm.h
改变 TiffDecode.h
1,4c1,9
< extern int ImagingLibTiffInit(ImagingCodecState state, int fp, uint32 offset);
< extern int ImagingLibTiffEncodeInit(ImagingCodecState state, char *filename, int fp);
< extern int ImagingLibTiffMergeFieldInfo(ImagingCodecState state, TIFFDataType field_type, int key, int
is_var_length);
< extern int ImagingLibTiffSetField(ImagingCodecState state, ttag_t tag, ...);
——
> extern int
> ImagingLibTiffInit(ImagingCodecState state, int fp, uint32_t offset);
> extern int
> ImagingLibTiffEncodeInit(ImagingCodecState state, char *filename, int fp);
> extern int
> ImagingLibTiffMergeFieldInfo(
>     ImagingCodecState state, TIFFDataType field_type, int key, int is_var_length);
> extern int
> ImagingLibTiffSetField(ImagingCodecState state, ttag_t tag, ...);

```

bash-devel

```

改变 arrayfunc.h
4a5
> extern SHELL_VAR *arrayvar_copyval (SHELL_VAR *, SHELL_VAR *);
改变 shell.h
16a17
> extern int pretty_print_mode;
改变 y.tab.h
2c2
< extern YYSTYPE yylval;
——
> extern YYSTYPE yylval;

```

httpd-devel

```

改变 http_core.h
12a13
> AP_DECLARE(apr_size_t) ap_get_read_buf_size(const request_rec *r);
改变 http_protocol.h
53c53,70

```

```

<     return ap_rwrite(str, (int)strlen(str), r);
____
>     apr_size_t len;
>     len = strlen(str);
>     for (;;) {
>         if (len <= INT_MAX) {
>             return ap_rwrite(str, (int)len, r);
>         }
>         else {
>             int rc;
>             rc = ap_rwrite(str, INT_MAX, r);
>             if (rc < 0) {
>                 return rc;
>             }
>             else {
>                 str += INT_MAX;
>                 len -= INT_MAX;
>             }
>         }
>     }

```

改变 mod_proxy.h

56a57,60

```

> PROXY_DECLARE(int) ap_proxy_worker_can_upgrade(apr_pool_t *p,
>
>                                     const proxy_worker *worker,
>                                     const char *upgrade,
>                                     const char *dflt);

```

158a163,187

```

> PROXY_DECLARE(apr_status_t) ap_proxy_tunnel_create(proxy_tunnel_rec **tunnel,
>
>                                     request_rec *r, conn_rec *c_o,
>                                     const char *scheme);
> PROXY_DECLARE(int) ap_proxy_tunnel_run(proxy_tunnel_rec *tunnel);
> APR_DECLARE_OPTIONAL_FN(int, ap_proxy_clear_connection,
>
>     (request_rec *r, apr_table_t *headers));
> PROXY_DECLARE(int) ap_proxy_is_socket_connected(apr_socket_t *socket);
> int ap_proxy_lb_workers(void);
> PROXY_DECLARE(apr_port_t) ap_proxy_port_of_scheme(const char *scheme);
> PROXY_DECLARE (const char *) ap_proxy_show_hcmethod(hcmethod_t method);
> PROXY_DECLARE(const char *) ap_proxy_de_socketfy(apr_pool_t *p, const char *url);
> PROXY_DECLARE(apr_status_t) ap_proxy_buckets_lifetime_transform(request_rec *r,
>
>                                     apr_bucket_brigade *from,
>                                     apr_bucket_brigade *to);
> PROXY_DECLARE(apr_status_t) ap_proxy_transfer_between_connections(
>
>                                     request_rec *r,
>                                     conn_rec *c_i,

```


增加头文件: pcompress.h

增加头文件: pdl.h

增加头文件: pfexec.h

增加头文件: pgpu.h

增加头文件: pif.h

增加头文件: pinstalldirs.h

增加头文件: pinstalldirs_types.h

增加头文件: plog.h

增加头文件: pmdl.h

增加头文件: pmix_alfg.h

增加头文件: pmix_argv.h

增加头文件: pmix_atomic.h

增加头文件: pmix_attributes.h

增加头文件: pmix_base.h

增加头文件: pmix_basename.h

增加头文件: pmix_bitmap.h

增加头文件: pmix_client_ops.h

增加头文件: pmix_cmd_line.h

改变 pmix_common.h

8,47c8

```
< PMIX_EXPORT void PMIx_Register_event_handler(pmix_status_t codes[], size_t ncodes,
<
<                                     pmix_info_t info[], size_t ninfo,
<                                     pmix_notification_fn_t evhdlr,
<                                     pmix_hdlr_reg_cbfunc_t cbfunc,
<                                     void *cbdata);
< PMIX_EXPORT void PMIx_Deregister_event_handler(size_t evhdlr_ref,
<
<                                     pmix_op_cbfunc_t cbfunc,
<                                     void *cbdata);
< PMIX_EXPORT pmix_status_t PMIx_Notify_event(pmix_status_t status,
<
<                                     const pmix_proc_t *source,
<                                     pmix_data_range_t range,
<                                     const pmix_info_t info[], size_t ninfo,
<                                     pmix_op_cbfunc_t cbfunc, void *cbdata);
< PMIX_EXPORT const char* PMIx_Error_string(pmix_status_t status);
< PMIX_EXPORT const char* PMIx_Proc_state_string(pmix_proc_state_t state);
< PMIX_EXPORT const char* PMIx_Scope_string(pmix_scope_t scope);
< PMIX_EXPORT const char* PMIx_Persistence_string(pmix_persistence_t persist);
< PMIX_EXPORT const char* PMIx_Data_range_string(pmix_data_range_t range);
< PMIX_EXPORT const char* PMIx_Info_directives_string(pmix_info_directives_t directives);
< PMIX_EXPORT const char* PMIx_Data_type_string(pmix_data_type_t type);
< PMIX_EXPORT const char* PMIx_Alloc_directive_string(pmix_alloc_directive_t directive);
< PMIX_EXPORT const char* PMIx_IOF_channel_string(pmix_iof_channel_t channel);
< PMIX_EXPORT const char* PMIx_Get_version(void);
< PMIX_EXPORT pmix_status_t PMIx_Store_internal(const pmix_proc_t *proc,
```

```

<                                     const pmix_key_t key, pmix_value_t *val);
< PMIX_EXPORT pmix_status_t PMIx_Data_pack(const pmix_proc_t *target,
<                                     pmix_data_buffer_t *buffer,
<                                     void *src, int32_t num_vals,
<                                     pmix_data_type_t type);
< PMIX_EXPORT pmix_status_t PMIx_Data_unpack(const pmix_proc_t *source,
<                                     pmix_data_buffer_t *buffer, void *dest,
<                                     int32_t *max_num_values,
<                                     pmix_data_type_t type);
< PMIX_EXPORT pmix_status_t PMIx_Data_copy(void **dest, void *src,
<                                     pmix_data_type_t type);
< PMIX_EXPORT pmix_status_t PMIx_Data_print(char **output, char *prefix,
<                                     void *src, pmix_data_type_t type);
< PMIX_EXPORT pmix_status_t PMIx_Data_copy_payload(pmix_data_buffer_t *dest,
<                                     pmix_data_buffer_t *src);
< static inline void pmix_darray_destruct(pmix_data_array_t *m);
----
> static inline void pmix_strncpy(char *dest,
68,102c13,18
< static inline void pmix_darray_destruct(pmix_data_array_t *m)
----
> static inline size_t pmix_keylen(const char *src)
106c23
< static inline void pmix_strncpy(char *dest, const char *src, size_t len)
----
> static inline size_t pmix_nslon(const char *src)
增加头文件: pmix_config_bottom.h
增加头文件: pmix_config.h
增加头文件: pmix_config_top.h
增加头文件: pmix_context_fns.h
增加头文件: pmix_deprecated.h
增加头文件: pmix_dictionary.h
增加头文件: pmix_environ.h
增加头文件: pmix_error.h
增加头文件: pmix_event.h
增加头文件: pmix_event_strings.h
减少头文件: pmix_extend.h
增加头文件: pmix_fd.h
增加头文件: pmix_few.h
增加头文件: pmix_frameworks.h
增加头文件: pmix_getcwd.h
增加头文件: pmix_getid.h
增加头文件: pmix_globals.h
改变 pmix.h

```

8c8,10

```
< PMIX_EXPORT pmix_status_t PMIx_Put(pmix_scope_t scope, const pmix_key_t key, pmix_value_t *val);
```

```
> PMIX_EXPORT pmix_status_t PMIx_Put(pmix_scope_t scope,  
>                                     const char key[],  
>                                     pmix_value_t *val);
```

15c17

```
< PMIX_EXPORT pmix_status_t PMIx_Get(const pmix_proc_t *proc, const pmix_key_t key,
```

```
> PMIX_EXPORT pmix_status_t PMIx_Get(const pmix_proc_t *proc, const char key[],
```

18c20

```
< PMIX_EXPORT pmix_status_t PMIx_Get_nb(const pmix_proc_t *proc, const pmix_key_t key,
```

```
> PMIX_EXPORT pmix_status_t PMIx_Get_nb(const pmix_proc_t *proc, const char key[],
```

49c51,52

```
< PMIX_EXPORT pmix_status_t PMIx_Resolve_peers(const char *nodename, const pmix_nspace_t nspace,
```

```
> PMIX_EXPORT pmix_status_t PMIx_Resolve_peers(const char *nodename,  
>                                             const pmix_nspace_t nspace,
```

51a55,56

```
> PMIX_EXPORT pmix_status_t PMIx_Query_info(pmix_query_t queries[], size_t nqueries,  
>                                           pmix_info_t **results, size_t *nresults);
```

60c65,66

```
<                                             pmix_info_t *info, size_t ninfo);
```

```
>                                             pmix_info_t *info, size_t ninfo,  
>                                             pmix_info_t **results, size_t *nresults);
```

65c71,72

```
<                                             const pmix_info_t directives[], size_t ndirs);
```

```
>                                             const pmix_info_t directives[], size_t ndirs,  
>                                             pmix_info_t **results, size_t *nresults);
```

70c77,78

```
<                                             const pmix_info_t directives[], size_t ndirs);
```

```
>                                             const pmix_info_t directives[], size_t ndirs,  
>                                             pmix_info_t **results, size_t *nresults);
```

75c83,85

```
<                                             pmix_credential_cbfunc_t cbfunc, void *cbdata);
```

```
>                                             pmix_byte_object_t *credential);  
> PMIX_EXPORT pmix_status_t PMIx_Get_credential_nb(const pmix_info_t info[], size_t ninfo,  
>                                             pmix_credential_cbfunc_t cbfunc, void *cbdata);
```

78,84c88,121

```

<                                     pmix_validation_cbfunc_t cbfunc, void *cbdata);
< PMIX_EXPORT pmix_status_t PMIx_IOF_pull(const pmix_proc_t procs[], size_t nprocs,
<                                     const pmix_info_t directives[], size_t ndirs,
<                                     pmix_iof_channel_t channel, pmix_iof_cbfunc_t cbfunc,
<                                     pmix_hdlr_reg_cbfunc_t regcbfunc, void *regcbdata);
< PMIX_EXPORT pmix_status_t PMIx_IOF_deregister(size_t iofhdlr,
<                                     const pmix_info_t directives[], size_t ndirs,
<
-----
>                                     pmix_info_t **results, size_t *nresults);
> PMIX_EXPORT pmix_status_t PMIx_Validate_credential_nb(const pmix_byte_object_t *cred,
>                                     const pmix_info_t info[], size_t ninfo,
>                                     pmix_validation_cbfunc_t cbfunc, void
*cbdata);
> PMIX_EXPORT pmix_status_t PMIx_Group_construct(const char grp[],
>                                     const pmix_proc_t procs[], size_t nprocs,
>                                     const pmix_info_t directives[], size_t ndirs,
>                                     pmix_info_t **results, size_t *nresults);
> PMIX_EXPORT pmix_status_t PMIx_Group_construct_nb(const char grp[],
>                                     const pmix_proc_t procs[], size_t nprocs,
>                                     const pmix_info_t info[], size_t ninfo,
>                                     pmix_info_cbfunc_t cbfunc, void *cbdata);
> PMIX_EXPORT pmix_status_t PMIx_Group_invite(const char grp[],
>                                     const pmix_proc_t procs[], size_t nprocs,
>                                     const pmix_info_t info[], size_t ninfo,
>                                     pmix_info_t **results, size_t *nresult);
> PMIX_EXPORT pmix_status_t PMIx_Group_invite_nb(const char grp[],
>                                     const pmix_proc_t procs[], size_t nprocs,
>                                     const pmix_info_t info[], size_t ninfo,
>                                     pmix_info_cbfunc_t cbfunc, void *cbdata);
> PMIX_EXPORT pmix_status_t PMIx_Group_join(const char grp[],
>                                     const pmix_proc_t *leader,
>                                     pmix_group_opt_t opt,
>                                     const pmix_info_t info[], size_t ninfo,
>                                     pmix_info_t **results, size_t *nresult);
> PMIX_EXPORT pmix_status_t PMIx_Group_join_nb(const char grp[],
>                                     const pmix_proc_t *leader,
>                                     pmix_group_opt_t opt,
>                                     const pmix_info_t info[], size_t ninfo,
>                                     pmix_info_cbfunc_t cbfunc, void *cbdata);
> PMIX_EXPORT pmix_status_t PMIx_Group_leave(const char grp[],
>                                     const pmix_info_t info[], size_t ninfo);
> PMIX_EXPORT pmix_status_t PMIx_Group_leave_nb(const char grp[],
>                                     const pmix_info_t info[], size_t ninfo,

```



```
> pmix_info_t info[], size_t ninfo,  
pmix_device_dist_cbfunc_t cbfunc,  
void *cbdata);  
  
> PMIX_EXPORT pmix_status_t PMIx_Load_topology(pmix_topology_t *topo);  
> PMIX_EXPORT pmix_status_t PMIx_Parse_cpuset_string(const char *cpuset_string,  
pmix_cpuset_t *cpuset);  
  
> PMIX_EXPORT pmix_status_t PMIx_Get_cpuset(pmix_cpuset_t *cpuset, pmix_bind_envelope_t ref);  
> PMIX_EXPORT pmix_status_t PMIx_Get_relative_locality(const char *locality1,  
const char *locality2,  
pmix_locality_t *locality);  
  
> PMIX_EXPORT void PMIx_Progress(void);  
  
> PMIX_EXPORT const char* PMIx_Error_string(pmix_status_t status);  
> PMIX_EXPORT pmix_status_t PMIx_Error_code(const char *errname);  
> PMIX_EXPORT const char* PMIx_Proc_state_string(pmix_proc_state_t state);  
> PMIX_EXPORT const char* PMIx_Scope_string(pmix_scope_t scope);  
> PMIX_EXPORT const char* PMIx_Persistence_string(pmix_persistence_t persist);  
> PMIX_EXPORT const char* PMIx_Data_range_string(pmix_data_range_t range);  
> PMIX_EXPORT const char* PMIx_Data_type_string(pmix_data_type_t type);  
> PMIX_EXPORT const char* PMIx_Alloc_directive_string(pmix_alloc_directive_t directive);  
> PMIX_EXPORT const char* PMIx_IOF_channel_string(pmix_iof_channel_t channel);  
> PMIX_EXPORT const char* PMIx_Job_state_string(pmix_job_state_t state);  
> PMIX_EXPORT const char* PMIx_Get_attribute_string(const char *attribute);  
> PMIX_EXPORT const char* PMIx_Get_attribute_name(const char *attrstring);  
> PMIX_EXPORT const char* PMIx_Link_state_string(pmix_link_state_t state);  
> PMIX_EXPORT const char* PMIx_Device_type_string(pmix_device_type_t type);  
> PMIX_EXPORT const char* PMIx_Value_comparison_string(pmix_value_cmp_t cmp);  
> PMIX_EXPORT char* PMIx_Info_string(const pmix_info_t *info);  
> PMIX_EXPORT char* PMIx_Value_string(const pmix_value_t *val);  
> PMIX_EXPORT char* PMIx_Info_directives_string(pmix_info_directives_t directives);  
> PMIX_EXPORT char* PMIx_App_string(const pmix_app_t *app);  
> PMIX_EXPORT const char* PMIx_Get_version(void);  
  
> PMIX_EXPORT pmix_status_t PMIx_Store_internal(const pmix_proc_t *proc,  
const char key[], pmix_value_t *val);  
  
> PMIX_EXPORT pmix_status_t PMIx_Value_get_size(const pmix_value_t *val,  
size_t *size);  
  
> PMIX_EXPORT pmix_status_t PMIx_Info_get_size(const pmix_info_t *val,  
size_t *size);  
  
> PMIX_EXPORT pmix_status_t PMIx_Data_pack(const pmix_proc_t *target,  
pmix_data_buffer_t *buffer,  
void *src, int32_t num_vals,  
pmix_data_type_t type);  
  
> PMIX_EXPORT pmix_status_t PMIx_Data_unpack(const pmix_proc_t *source,  
pmix_data_buffer_t *buffer, void *dest,  
int32_t *max_num_values,
```

```

>                                     pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Data_copy(void **dest, void *src,
>                                     pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Data_print(char **output, char *prefix,
>                                     void *src, pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Data_copy_payload(pmix_data_buffer_t *dest,
>                                     pmix_data_buffer_t *src);
> PMIX_EXPORT pmix_status_t PMIx_Data_unload(pmix_data_buffer_t *buffer,
>                                     pmix_byte_object_t *payload);
> PMIX_EXPORT pmix_status_t PMIx_Data_load(pmix_data_buffer_t *buffer,
>                                     pmix_byte_object_t *payload);
> PMIX_EXPORT pmix_status_t PMIx_Data_embed(pmix_data_buffer_t *buffer,
>                                     const pmix_byte_object_t *payload);
> PMIX_EXPORT bool PMIx_Data_compress(const uint8_t *inbytes,
>                                     size_t size,
>                                     uint8_t **outbytes,
>                                     size_t *nbytes);
> PMIX_EXPORT bool PMIx_Data_decompress(const uint8_t *inbytes,
>                                     size_t size,
>                                     uint8_t **outbytes,
>                                     size_t *nbytes);
> PMIX_EXPORT void PMIx_Load_key(pmix_key_t key, const char *src);
> PMIX_EXPORT bool PMIx_Check_key(const char *key, const char *str);
> PMIX_EXPORT bool PMIx_Check_reserved_key(const char *key);
> PMIX_EXPORT void PMIx_Load_namespace(pmix_namespace_t nspace, const char *str);
> PMIX_EXPORT bool PMIx_Check_namespace(const char *key1, const char *key2);
> PMIX_EXPORT bool PMIx_Nspace_invalid(const char *namespace);
> PMIX_EXPORT void PMIx_Load_procid(pmix_proc_t *p,
>                                     const char *ns,
>                                     pmix_rank_t rk);
> PMIX_EXPORT void PMIx_Xfer_procid(pmix_proc_t *dst,
>                                     const pmix_proc_t *src);
> PMIX_EXPORT bool PMIx_Check_procid(const pmix_proc_t *a,
>                                     const pmix_proc_t *b);
> PMIX_EXPORT bool PMIx_Check_rank(pmix_rank_t a,
>                                     pmix_rank_t b);
> PMIX_EXPORT bool PMIx_Procid_invalid(const pmix_proc_t *p);
> PMIX_EXPORT int PMIx_Argv_count(char **a);
> PMIX_EXPORT pmix_status_t PMIx_Argv_append_nosize(char ***argv, const char *arg);
> PMIX_EXPORT pmix_status_t PMIx_Argv_prepend_nosize(char ***argv, const char *arg);
> PMIX_EXPORT pmix_status_t PMIx_Argv_append_unique_nosize(char ***argv, const char *arg);
> PMIX_EXPORT void PMIx_Argv_free(char **argv);
> PMIX_EXPORT char **PMIx_Argv_split_inter(const char *src_string,
>                                     int delimiter,

```



```

>
> bool include_empty);
> PMIX_EXPORT char **PMIx_Argv_split_with_empty(const char *src_string, int delimiter);
> PMIX_EXPORT char **PMIx_Argv_split(const char *src_string, int delimiter);
> PMIX_EXPORT char *PMIx_Argv_join(char **argv, int delimiter);
> PMIX_EXPORT char **PMIx_Argv_copy(char **argv);
> PMIX_EXPORT pmix_status_t PMIx_Setenv(const char *name,
>
> const char *value,
>
> bool overwrite,
>
> char ***env);
> PMIX_EXPORT void PMIx_Value_construct(pmix_value_t *val);
> PMIX_EXPORT void PMIx_Value_destruct(pmix_value_t *val);
> PMIX_EXPORT pmix_value_t* PMIx_Value_create(size_t n);
> PMIX_EXPORT void PMIx_Value_free(pmix_value_t *v, size_t n);
> PMIX_EXPORT pmix_boolean_t PMIx_Value_true(const pmix_value_t *v);
> PMIX_EXPORT pmix_status_t PMIx_Value_load(pmix_value_t *val,
>
> const void *data,
>
> pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Value_unload(pmix_value_t *val,
>
> void **data,
>
> size_t *sz);
> PMIX_EXPORT pmix_status_t PMIx_Value_xfer(pmix_value_t *dest,
>
> const pmix_value_t *src);
> PMIX_EXPORT pmix_value_cmp_t PMIx_Value_compare(pmix_value_t *v1,
>
> pmix_value_t *v2);
> PMIX_EXPORT void PMIx_Data_array_init(pmix_data_array_t *p,
>
> pmix_data_type_t type);
> PMIX_EXPORT void PMIx_Data_array_construct(pmix_data_array_t *p,
>
> size_t num, pmix_data_type_t type);
> PMIX_EXPORT void PMIx_Data_array_destruct(pmix_data_array_t *d);
> PMIX_EXPORT pmix_data_array_t* PMIx_Data_array_create(size_t n, pmix_data_type_t type);
> PMIX_EXPORT void PMIx_Data_array_free(pmix_data_array_t *p);
> PMIX_EXPORT void PMIx_Info_construct(pmix_info_t *p);
> PMIX_EXPORT void PMIx_Info_destruct(pmix_info_t *p);
> PMIX_EXPORT pmix_info_t* PMIx_Info_create(size_t n);
> PMIX_EXPORT void PMIx_Info_free(pmix_info_t *p, size_t n);
> PMIX_EXPORT pmix_boolean_t PMIx_Info_true(const pmix_info_t *p);
> PMIX_EXPORT pmix_status_t PMIx_Info_load(pmix_info_t *info,
>
> const char *key,
>
> const void *data,
>
> pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Info_xfer(pmix_info_t *dest,
>
> const pmix_info_t *src);
> PMIX_EXPORT void PMIx_Info_required(pmix_info_t *p);
> PMIX_EXPORT void PMIx_Info_optional(pmix_info_t *p);

```

```

> PMIX_EXPORT bool PMIx_Info_is_required(const pmix_info_t *p);
> PMIX_EXPORT bool PMIx_Info_is_optional(const pmix_info_t *p);
> PMIX_EXPORT void PMIx_Info_processed(pmix_info_t *p);
> PMIX_EXPORT bool PMIx_Info_was_processed(const pmix_info_t *p);
> PMIX_EXPORT void PMIx_Info_set_end(pmix_info_t *p);
> PMIX_EXPORT bool PMIx_Info_is_end(const pmix_info_t *p);
> PMIX_EXPORT void PMIx_Info_qualifier(pmix_info_t *p);
> PMIX_EXPORT bool PMIx_Info_is_qualifier(const pmix_info_t *p);
> PMIX_EXPORT void PMIx_Info_persistent(pmix_info_t *p);
> PMIX_EXPORT bool PMIx_Info_is_persistent(const pmix_info_t *p);
> PMIX_EXPORT void PMIx_Coord_construct(pmix_coord_t *m);
> PMIX_EXPORT void PMIx_Coord_destruct(pmix_coord_t *m);
> PMIX_EXPORT pmix_coord_t* PMIx_Coord_create(size_t dims,
>
>                                     size_t number);
> PMIX_EXPORT void PMIx_Coord_free(pmix_coord_t *m, size_t number);
> PMIX_EXPORT void PMIx_Topology_construct(pmix_topology_t *t);
> PMIX_EXPORT void PMIx_Topology_destruct(pmix_topology_t *topo);
> PMIX_EXPORT pmix_topology_t* PMIx_Topology_create(size_t n);
> PMIX_EXPORT void PMIx_Topology_free(pmix_topology_t *t, size_t n);
> PMIX_EXPORT void PMIx_Cpuset_construct(pmix_cpuset_t *cpuset);
> PMIX_EXPORT void PMIx_Cpuset_destruct(pmix_cpuset_t *cpuset);
> PMIX_EXPORT pmix_cpuset_t* PMIx_Cpuset_create(size_t n);
> PMIX_EXPORT void PMIx_Cpuset_free(pmix_cpuset_t *c, size_t n);
> PMIX_EXPORT void PMIx_Geometry_construct(pmix_geometry_t *g);
> PMIX_EXPORT void PMIx_Geometry_destruct(pmix_geometry_t *g);
> PMIX_EXPORT pmix_geometry_t* PMIx_Geometry_create(size_t n);
> PMIX_EXPORT void PMIx_Geometry_free(pmix_geometry_t *g, size_t n);
> PMIX_EXPORT void PMIx_Device_distance_construct(pmix_device_distance_t *d);
> PMIX_EXPORT void PMIx_Device_distance_destruct(pmix_device_distance_t *d);
> PMIX_EXPORT pmix_device_distance_t* PMIx_Device_distance_create(size_t n);
> PMIX_EXPORT void PMIx_Device_distance_free(pmix_device_distance_t *d, size_t n);
> PMIX_EXPORT void PMIx_Byte_object_construct(pmix_byte_object_t *b);
> PMIX_EXPORT void PMIx_Byte_object_destruct(pmix_byte_object_t *g);
> PMIX_EXPORT pmix_byte_object_t* PMIx_Byte_object_create(size_t n);
> PMIX_EXPORT void PMIx_Byte_object_free(pmix_byte_object_t *g, size_t n);
> PMIX_EXPORT void PMIx_Byte_object_load(pmix_byte_object_t *b,
>
>                                     char *d, size_t sz);
> PMIX_EXPORT void PMIx_Endpoint_construct(pmix_endpoint_t *e);
> PMIX_EXPORT void PMIx_Endpoint_destruct(pmix_endpoint_t *e);
> PMIX_EXPORT pmix_endpoint_t* PMIx_Endpoint_create(size_t n);
> PMIX_EXPORT void PMIx_Endpoint_free(pmix_endpoint_t *e, size_t n);
> PMIX_EXPORT void PMIx_Envvar_construct(pmix_envvar_t *e);
> PMIX_EXPORT void PMIx_Envvar_destruct(pmix_envvar_t *e);
> PMIX_EXPORT pmix_envvar_t* PMIx_Envvar_create(size_t n);

```

```

> PMIX_EXPORT void PMIx_Envvar_free(pmix_envvar_t *e, size_t n);
> PMIX_EXPORT void PMIx_Envvar_load(pmix_envvar_t *e,
>
>                                     char *var,
>                                     char *value,
>                                     char separator);
> PMIX_EXPORT void PMIx_Data_buffer_construct(pmix_data_buffer_t *b);
> PMIX_EXPORT void PMIx_Data_buffer_destruct(pmix_data_buffer_t *b);
> PMIX_EXPORT pmix_data_buffer_t* PMIx_Data_buffer_create(void);
> PMIX_EXPORT void PMIx_Data_buffer_release(pmix_data_buffer_t *b);
> PMIX_EXPORT void PMIx_Data_buffer_load(pmix_data_buffer_t *b,
>
>                                     char *bytes, size_t sz);
> PMIX_EXPORT void PMIx_Data_buffer_unload(pmix_data_buffer_t *b,
>
>                                     char **bytes, size_t *sz);
> PMIX_EXPORT void PMIx_Proc_construct(pmix_proc_t *p);
> PMIX_EXPORT void PMIx_Proc_destruct(pmix_proc_t *p);
> PMIX_EXPORT pmix_proc_t* PMIx_Proc_create(size_t n);
> PMIX_EXPORT void PMIx_Proc_free(pmix_proc_t *p, size_t n);
> PMIX_EXPORT void PMIx_Proc_load(pmix_proc_t *p,
>
>                                     char *nspace, pmix_rank_t rank);
> PMIX_EXPORT void PMIx_Multicluster_nspace_construct(pmix_nspace_t target,
>
>                                     pmix_nspace_t cluster,
>                                     pmix_nspace_t nspace);
> PMIX_EXPORT void PMIx_Multicluster_nspace_parse(pmix_nspace_t target,
>
>                                     pmix_nspace_t cluster,
>                                     pmix_nspace_t nspace);
> PMIX_EXPORT void PMIx_Proc_info_construct(pmix_proc_info_t *p);
> PMIX_EXPORT void PMIx_Proc_info_destruct(pmix_proc_info_t *p);
> PMIX_EXPORT pmix_proc_info_t* PMIx_Proc_info_create(size_t n);
> PMIX_EXPORT void PMIx_Proc_info_free(pmix_proc_info_t *p, size_t n);
> PMIX_EXPORT void PMIx_Proc_stats_construct(pmix_proc_stats_t *p);
> PMIX_EXPORT void PMIx_Proc_stats_destruct(pmix_proc_stats_t *p);
> PMIX_EXPORT pmix_proc_stats_t* PMIx_Proc_stats_create(size_t n);
> PMIX_EXPORT void PMIx_Proc_stats_free(pmix_proc_stats_t *p, size_t n);
> PMIX_EXPORT void PMIx_Disk_stats_construct(pmix_disk_stats_t *p);
> PMIX_EXPORT void PMIx_Disk_stats_destruct(pmix_disk_stats_t *p);
> PMIX_EXPORT pmix_disk_stats_t* PMIx_Disk_stats_create(size_t n);
> PMIX_EXPORT void PMIx_Disk_stats_free(pmix_disk_stats_t *p, size_t n);
> PMIX_EXPORT void PMIx_Net_stats_construct(pmix_net_stats_t *p);
> PMIX_EXPORT void PMIx_Net_stats_destruct(pmix_net_stats_t *p);
> PMIX_EXPORT pmix_net_stats_t* PMIx_Net_stats_create(size_t n);
> PMIX_EXPORT void PMIx_Net_stats_free(pmix_net_stats_t *p, size_t n);
> PMIX_EXPORT void PMIx_Pdata_construct(pmix_pdata_t *p);
> PMIX_EXPORT void PMIx_Pdata_destruct(pmix_pdata_t *p);
> PMIX_EXPORT pmix_pdata_t* PMIx_Pdata_create(size_t n);

```

```

> PMIX_EXPORT void PMIx_Pdata_free(pmix_pdata_t *p, size_t n);
> PMIX_EXPORT void PMIx_App_construct(pmix_app_t *p);
> PMIX_EXPORT void PMIx_App_destruct(pmix_app_t *p);
> PMIX_EXPORT pmix_app_t* PMIx_App_create(size_t n);
> PMIX_EXPORT void PMIx_App_info_create(pmix_app_t *p, size_t n);
> PMIX_EXPORT void PMIx_App_free(pmix_app_t *p, size_t n);
> PMIX_EXPORT void PMIx_App_release(pmix_app_t *p);
> PMIX_EXPORT void* PMIx_Info_list_start(void);
> PMIX_EXPORT pmix_status_t PMIx_Info_list_add(void *ptr,
>
>                                     const char *key,
>
>                                     const void *value,
>
>                                     pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Info_list_prepend(void *ptr,
>
>                                     const char *key,
>
>                                     const void *value,
>
>                                     pmix_data_type_t type);
> PMIX_EXPORT pmix_status_t PMIx_Info_list_insert(void *ptr, pmix_info_t *info);
> PMIX_EXPORT pmix_status_t PMIx_Info_list_xfer(void *ptr,
>
>                                     const pmix_info_t *info);
> PMIX_EXPORT pmix_status_t PMIx_Info_list_convert(void *ptr, pmix_data_array_t *par);
> PMIX_EXPORT void PMIx_Info_list_release(void *ptr);
> PMIX_EXPORT pmix_info_t* PMIx_Info_list_get_info(void *ptr, void *prev, void **next);

```

增加头文件: pmix_hash.h

增加头文件: pmix_hash_string.h

增加头文件: pmix_hash_table.h

增加头文件: pmix_hotel.h

增加头文件: pmix_hwloc.h

增加头文件: pmix_if.h

增加头文件: pmix_init_util.h

增加头文件: pmix_iof.h

增加头文件: pmix_keyval_parse.h

增加头文件: pmix_list.h

增加头文件: pmix_mca_base_alias.h

增加头文件: pmix_mca_base_component_repository.h

增加头文件: pmix_mca_base_framework.h

增加头文件: pmix_mca_base_var_enum.h

增加头文件: pmix_mca_base_var_group.h

增加头文件: pmix_mca_base_var.h

增加头文件: pmix_mca_base_vari.h

增加头文件: pmix_mutex.h

增加头文件: pmix_mutex_unix.h

增加头文件: pmix_name_fns.h

增加头文件: pmix_net.h

增加头文件: pmix_object.h

增加头文件: pmix_os_dirpath.h
增加头文件: pmix_os_path.h
增加头文件: pmix_output.h
增加头文件: pmix_parse_options.h
增加头文件: pmix_path.h
增加头文件: pmix_pointer_array.h
增加头文件: pmix_portable_platform.h
增加头文件: pmix_portable_platform_real.h
增加头文件: pmix_prefetch.h
增加头文件: pmix_printf.h
增加头文件: pmix_progress_threads.h
增加头文件: pmix_ptty.h
减少头文件: pmix_rename.h
增加头文件: pmix_ring_buffer.h
增加头文件: pmix_rte.h
改变 pmix_server.h

37a38,51

```
> PMIX_EXPORT pmix_status_t PMIx_Register_attributes(char *function, char *attrs[]);  
> PMIX_EXPORT pmix_status_t PMIx_server_generate_locality_string(const pmix_cpuset_t *cpuset,  
>                                                                    char **locality);  
> PMIX_EXPORT pmix_status_t PMIx_server_generate_cpuset_string(const pmix_cpuset_t *cpuset,  
>                                                                    char **cpuset_string);  
> PMIX_EXPORT pmix_status_t PMIx_server_define_process_set(const pmix_proc_t *members,  
>                                                                    size_t nmembers, const char *pset_name);  
> PMIX_EXPORT pmix_status_t PMIx_server_delete_process_set(char *pset_name);  
> PMIX_EXPORT pmix_status_t PMIx_server_register_resources(pmix_info_t info[], size_t ninfo,  
>                                                                    pmix_op_cbfunc_t cbfunc,  
>                                                                    void *cbdata);  
> PMIX_EXPORT pmix_status_t PMIx_server_deregister_resources(pmix_info_t info[], size_t ninfo,  
>                                                                    pmix_op_cbfunc_t cbfunc,  
>                                                                    void *cbdata);
```

增加头文件: pmix_server_ops.h
增加头文件: pmix_show_help.h
增加头文件: pmix_socket_errno.h
增加头文件: pmix_stdatomic.h
增加头文件: pmix_stdint.h
增加头文件: pmix_string_copy.h
增加头文件: pmix_strnlen.h
增加头文件: pmix_threads.h
增加头文件: pmix_timings.h
改变 pmix_tool.h

3a4

```
> PMIX_EXPORT pmix_status_t PMIx_tool_set_server_module(pmix_server_module_t *module);
```

5,6c6,23

```

< PMIX_EXPORT pmix_status_t PMIx_tool_connect_to_server(pmix_proc_t *proc,
<
pmix_info_t info[], size_t ninfo);
——
> PMIX_EXPORT bool PMIx_tool_is_connected(void);
> PMIX_EXPORT pmix_status_t PMIx_tool_attach_to_server(pmix_proc_t *myproc, pmix_proc_t *server,
>
pmix_info_t info[], size_t ninfo);
> PMIX_EXPORT pmix_status_t PMIx_tool_disconnect(const pmix_proc_t *server);
> PMIX_EXPORT pmix_status_t PMIx_tool_get_servers(pmix_proc_t *servers[], size_t *nservers);
> PMIX_EXPORT pmix_status_t PMIx_tool_set_server(const pmix_proc_t *server,
>
pmix_info_t info[], size_t ninfo);
> PMIX_EXPORT pmix_status_t PMIx_IOF_pull(const pmix_proc_t procs[], size_t nprocs,
>
const pmix_info_t directives[], size_t ndirs,
>
pmix_iof_channel_t channel, pmix_iof_cbfunc_t cbfunc,
>
pmix_hdlr_reg_cbfunc_t regcbfunc, void *regcbdata);
> PMIX_EXPORT pmix_status_t PMIx_IOF_deregister(size_t iofhdlr,
>
const pmix_info_t directives[], size_t ndirs,
>
pmix_op_cbfunc_t cbfunc, void *cbdata);
> PMIX_EXPORT pmix_status_t PMIx_IOF_push(const pmix_proc_t targets[], size_t ntargets,
>
pmix_byte_object_t *bo,
>
const pmix_info_t directives[], size_t ndirs,
>
pmix_op_cbfunc_t cbfunc, void *cbdata);

```

增加头文件: pmix_tool_ops.h

增加头文件: pmix_tsd.h

增加头文件: pmix_types.h

增加头文件: pmix_value_array.h

增加头文件: pmix_vmem.h

增加头文件: pnet.h

增加头文件: preg.h

增加头文件: preg_types.h

增加头文件: prn.h

增加头文件: psec.h

增加头文件: psensor.h

增加头文件: pshmem.h

增加头文件: psquash.h

增加头文件: pstat.h

增加头文件: pstrg.h

增加头文件: ptl_base_handshake.h

增加头文件: ptl.h

增加头文件: ptl_types.h

fcitx-devel

增加头文件: gb18030.h

kernel-headers

减少文件: cm4000_cs.h

减少文件: cxgb3-abi.h

减少文件: dn.h

减少文件: lightnvm.h

减少文件: netfilter_decnet.h

增加文件: nexthop.h

增加文件: raw_gadget.h

增加文件: rpl.h

增加文件: rpl_iptunnel.h

增加文件: scsi_bsg_ufs.h

改动 semem_com.h:

1c1,2

```
< static __inline__ unsigned long range_end(struct range_struct *range){
```

```
----
```

```
> static __inline__ unsigned long range_end(struct range_struct *range)
```

```
> {
```

4c5,7

```
< static __inline__ unsigned long semem_range_flags(unsigned long attr, unsigned long status){
```

```
----
```

```
> static __inline__ unsigned long semem_range_flags(unsigned long attr,
```

```
>                               unsigned long status)
```

```
> {
```

7c10,11

```
< static __inline__ unsigned long semem_range_attr(struct semem_range *range){
```

```
----
```

```
> static __inline__ unsigned long semem_range_attr(struct semem_range *range)
```

```
> {
```

10c14,15

```
< static __inline__ unsigned long semem_range_status(struct semem_range *range){
```

```
----
```

```
> static __inline__ unsigned long semem_range_status(struct semem_range *range)
```

```
> {
```

13c18,19

```
< static __inline__ unsigned long semem_range_end(struct semem_range *a){
```

```
----
```

```
> static __inline__ unsigned long semem_range_end(struct semem_range *a)
```

```
> {
```

16c22,23

```
< static __inline__ unsigned long com_range_end(struct range_struct *a){
```

```
----
```

```
> static __inline__ unsigned long com_range_end(struct range_struct *a)
```

```
> {
```

19c26,28

```

< static __inline__ int is_range_overlap(struct range_struct *a, struct range_struct *b){
____
> static __inline__ int is_range_overlap(struct range_struct *a,
>                                     struct range_struct *b)
> {
21c30
<                                     ((b->start >= a->start) && (b->start < (a->start + a->size)));
____
>                                     ((b->start >= a->start) && (b->start < (a->start + a->size)));
23c32,33
< static __inline__ int is_range_equal(struct range_struct *a, struct range_struct *b){
____
> static __inline__ int is_range_equal(struct range_struct *a, struct range_struct *b)
> {
26c36,38
< static __inline__ int is_semem_range_equal(struct semem_range *a, struct semem_range *b){
____
> static __inline__ int is_semem_range_equal(struct semem_range *a,
>                                     struct semem_range *b)
> {
29c41,42
< static __inline__ int range_valid_check(struct range_struct *a){
____
> static __inline__ int range_valid_check(struct range_struct *a)
> {
32c45,46
< static __inline__ int semem_range_valid_check(struct semem_range *a){
____
> static __inline__ int semem_range_valid_check(struct semem_range *a)
> {
35c49,51
< static __inline__ void get_semem_range(struct semem_range *se_range, struct range_struct *range){
____
> static __inline__ void get_semem_range(struct semem_range *se_range,
>                                     struct range_struct *range)
> {
减少文件: semem_ioctl.h

```

变动包清单

```

libssh-devel
rpm-devel
libkysec-devel
nagios-devel
poppler-glib-devel

```


postgresql-devel
check-devel
libabigail-devel
poppler-qt5-devel
libtommath-devel
nss-softoken-devel
bind-devel
kmod-devel
kysec-common-devel
espeak-ng-devel
grpc-devel
hsdimm-lite-devel
evolution-data-server-devel
openssl-devel
mariadb-devel
proftpd-devel
openvswitch-devel
libwinpr-devel
freerdp-devel
poppler-cpp-devel
jemalloc-devel
dovecot-devel
gnutls-devel
libmtp-devel
samba-devel
gdm-devel
armadillo-devel
jsoncpp-devel
poppler-devel
cryptsetup-devel
papi-devel
NetworkManager-libnm-devel
avahi-devel
python3-devel
qt5-qtbase-devel
libvncserver-devel
liborcus-devel
libgcrypt-devel
sleuthkit-devel
hdf5-mpich-devel
protobuf-c-devel
libkysec-extend-devel
ImageMagick-c++-devel
libbpf-devel

protobuf-devel
varnish-devel
nettle-devel
libtar-devel
eom-devel
libdwarf-devel
nodejs-devel
virglrenderer-devel
libkysecwhlist-devel
php-devel
hdf5-devel
libnghttp2-devel
libwebsockets-devel
libreofficekit-devel
lcr-devel
libdvdnav-devel
e2fsprogs-devel
wireshark-devel
multipath-tools-devel
hiredis-devel
asio-devel
libevhttp-devel
hdf5-openmpi-devel
libpcap-devel
efivar-devel
ImageMagick-devel
systemd-devel
python3-pillow-devel
bash-devel
httpd-devel
bind-export-devel
nss-devel
pmix-devel
fcitx-devel
kernel-headers