

星光麒麟操作系统 1.0 应知应会

— 产品与社区发展中心

2023.02

- 1 智能终端操作系统概念定义
- 2 智能终端操作系统价值重要性
- 3 智能终端操作系统分类
- 4 智能终端操作系统发展轨迹
- 5 智能终端操作系统当前挑战及趋势

智能终端定义

智能终端是一种具有智能操作系统的嵌入式设备，可以连接互联网、下载和执行各种专门开发的应用程序，具有丰富多媒体处理能力和人机交互能力。一类为消费类智能终端产品，如智慧大屏、智能手机、平板电脑、笔记本、一体机、可穿戴设备等，另一类为行业应用和专业设备领域，如包括行业PDA、教育终端、工业终端、金融机具、医疗设备、安防终端等。



其他为：可穿戴、智能家居等等

智能终端OS

- ✓ 智能终端OS是面向万物智联场景，让海量设备从“连接到智能”的操作系统；
- ✓ 向下衔接硬件层的物理设备，向上为应用软件提供运行环境；
- ✓ 智能终端OS实现软件运行、资源管理、资源利用效率最大化；

特点

终端OS方向从功能化到智能化发展



软硬件控制



开放生态



多终端协同



人机交互



多应用生态



AI智能化

常见智能终端OS架构

终端OS扮演“承上启下”作用

应用层

应用框架Framework

系统功能库Function Lib

用户图形交互界面GUI

OS内核层

硬件、BSP



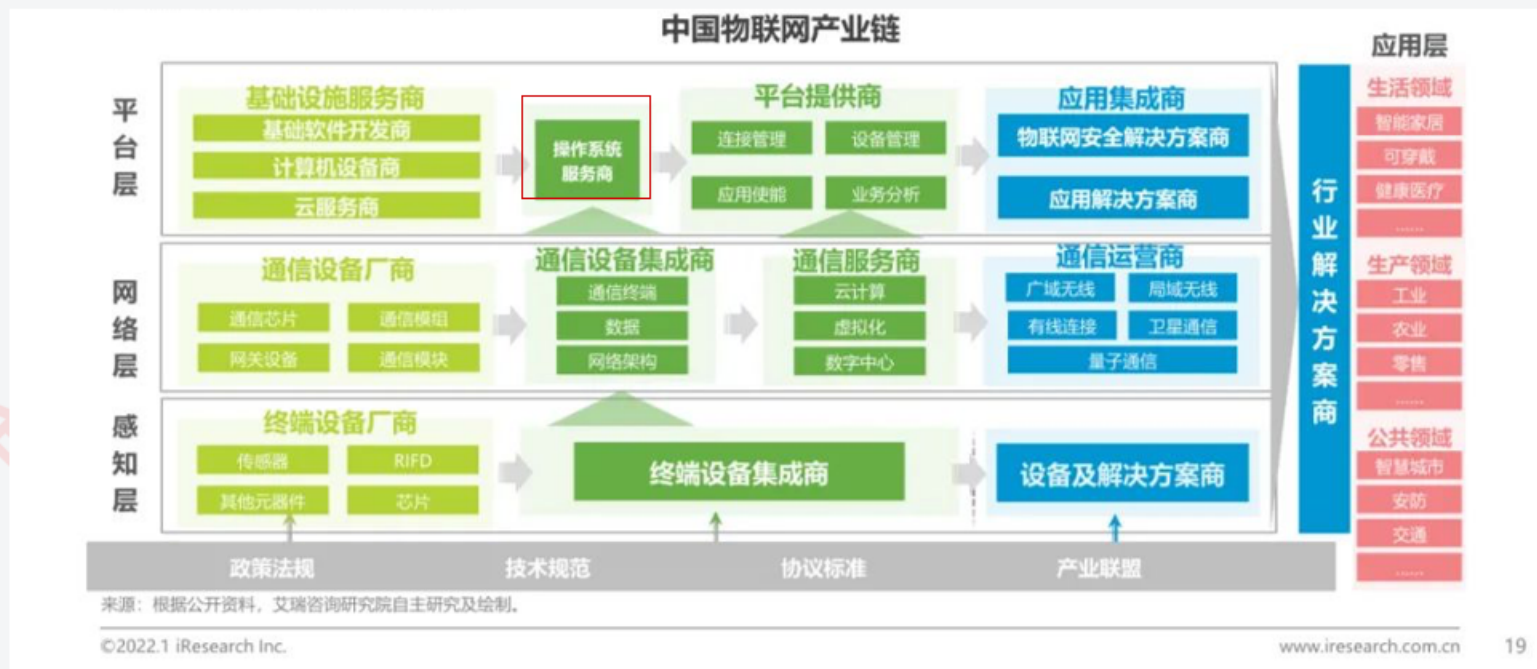
- 1 智能终端操作系统概念定义
- 2 智能终端操作系统价值重要性
- 3 智能终端操作系统分类
- 4 智能终端操作系统发展轨迹
- 5 智能终端操作系统当前挑战及趋势

终端OS核心作用—上承落地应用，下接海量终端



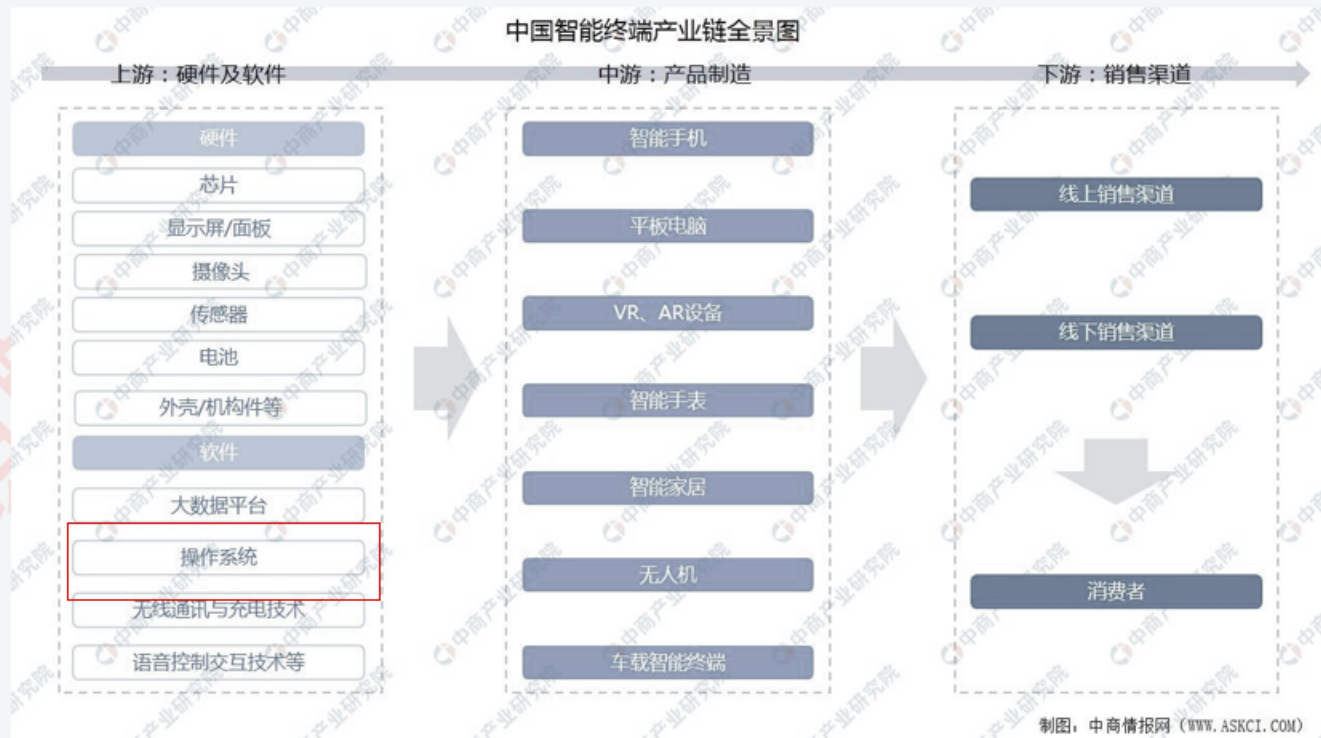
终端OS在物联网场景价值表现

终端OS所处平台层，主要对设备层进行连接与控制，提供智能分析与数据处理能力，将针对场景的核心应用固化为功能模块等，这一层对业务逻辑、统一建模、全链路技术能力、高并发支撑能力等要求较高；



终端OS在终端产业链中位置

- ✓ 在“智能+”浪潮中，OS助力终端设备连接更多万物智联入口，开启“移固融合”新体验；
- ✓ 终端OS研发方向受上游关键硬件方(芯片、主板)及中游终端厂商和ODM厂商影响；



- 1 智能终端操作系统概念定义
- 2 智能终端操作系统价值重要性
- 3 智能终端操作系统分类
- 4 智能终端操作系统发展轨迹
- 5 智能终端操作系统当前挑战及趋势

智能终端OS支持设备类型

内存规模	设备类型	具体内容	产品示意
128KB	轻量系统类设备	面向MCU类处理器，例如Arm Cortex-M、RISC-V 32位的设备，提供丰富的近距连接能力以及丰富的外设总线访问能力；	
128M~1G	小型系统类设备	面向应用处理器，例如ARM Cortex-A的设备，提供更高的安全能力，提供标准的图形架构，提供视频编码的多媒体能力；	
1G~4G	标准系统类设备	面向应用处理器，例如Arm Cortex-A的设备，提供增强的交互能力，提供3DGPU以及硬件合成能力，提供更多控件以及动效更丰富的图形能力。提供完整的应用框架；	
4G以上	大型系统类设备	面向应用处理器，例如Arm Cortex-A的设备，提供完整的兼容框架；	

智能终端场景常见OS类型

Android

- 覆盖toC&toB场景，消费终端和行业终端设备；
- Android Things和AndroidWear（后改名Wear OS）；

IOS

- 苹果自有生态，统一管理应用；
- 以toC为主；

Open Harmony

- toB 开源OS，行业应用；
- 鸿蒙生态之一；
- 逐渐完善产品和生态能力；

Linux+qt

- 基于Linux开源研发智能终端OS；
- 国内外多厂商开发；
- 市场占比低；

Firefox OS (Html5)

- 基于浏览器端操作系统，H5技术框架背景；
- 广泛应用手机、平板电脑等终端场景；

智能终端场景常见OS类型

	Andorid	IOS	OpenHarmony	Linux+qt	Firefox OS
厂商	谷歌	苹果	华为&开放原子基金会	/	Mozilla
开发背景	基于Linux内核核心代码开发	基于UNIX开发的	基于Linux内核核心代码开发	Linux+qt	基于Linux 以及 Gecko 引擎技术
内核形式	宏内核	混合内核 采用XNU与Darwin	微内核 基于Linux和LiteOS	宏内核	宏内核
芯片支持	Arm/Intel	自研+支持	Arm为主，支持国产SoC芯片	Arm为主，支持国内外芯片	Arm为主，支持国内外芯片
目标市场	toC/B	toC	toB为主	toB/C	toB/C
主要特点	• 开源AOSP • 服务套件-GMS • 硬件类型丰富 • 开发便捷 • 生态完整 • 多系统矩阵	• 闭源，自有生态，统一管理与应用 • 创新智慧互联服务 • 消费智能终端是重点 • 开发工具及套件遵循IOS生态	• 开源分布式操作系统，开放原子基金会管理，非安卓生态； • 统一OS，弹性部署 • 一次开发，多端部署 • 可支持不同终端设备类型 • 鸿蒙系厂商渗透行业市场	• 开源 • 丰富的API接口 • 优良的跨平台特性 • 快速开发和部署 • 模块化、结构化	• 开源OS，Web 页面运行，快速开发与部署 • 基于HTML5，跨终端、跨平台的接入 • 广泛应用移动终端 • 系统响应、速度流畅、轻量化、开发周期短
场景	覆盖消费终端和行业终端场景，个人、企业及行业	苹果自有终端产品，覆盖消费终端场景为主，家庭、办公、娱乐等	基于open鸿蒙，华为及生态伙伴开发不同终端场景商业发行版OS	基于linux 开源框架背景，开发面向不同行业场景的终端OS版本	基于开放的 HTML/JavaScript 来实现应用的开发，能够广泛适配支持不同移动终端设备

- 1 智能终端操作系统概念定义
- 2 智能终端操作系统价值重要性
- 3 智能终端操作系统分类
- 4 智能终端操作系统发展轨迹
- 5 智能终端操作系统当前挑战及趋势

终端OS发展轨迹

1990—1999 掌上终端OS时代

1、市场主要参与厂商为微软的Windows CE&Palm 公司的 Palm OS;



2000—2009 功能终端OS时代

1、市场的主要参与厂商为塞班联盟和微软的Windows Mobile;
2、2007年IOS首个版本正式发布, 2008年安卓首个版本发布;



2010—2015 智能终端时代

1、全球主流终端OS厂商为为安卓和苹果; 安卓完成了从3.0到10.0版本迭代; 而苹果IOS完成了从4.0到13.0版本的迭代;
2、新技术和4G网络加持下, 移动终端OS快速发展, 例如大屏、平板、手机、可穿戴等;

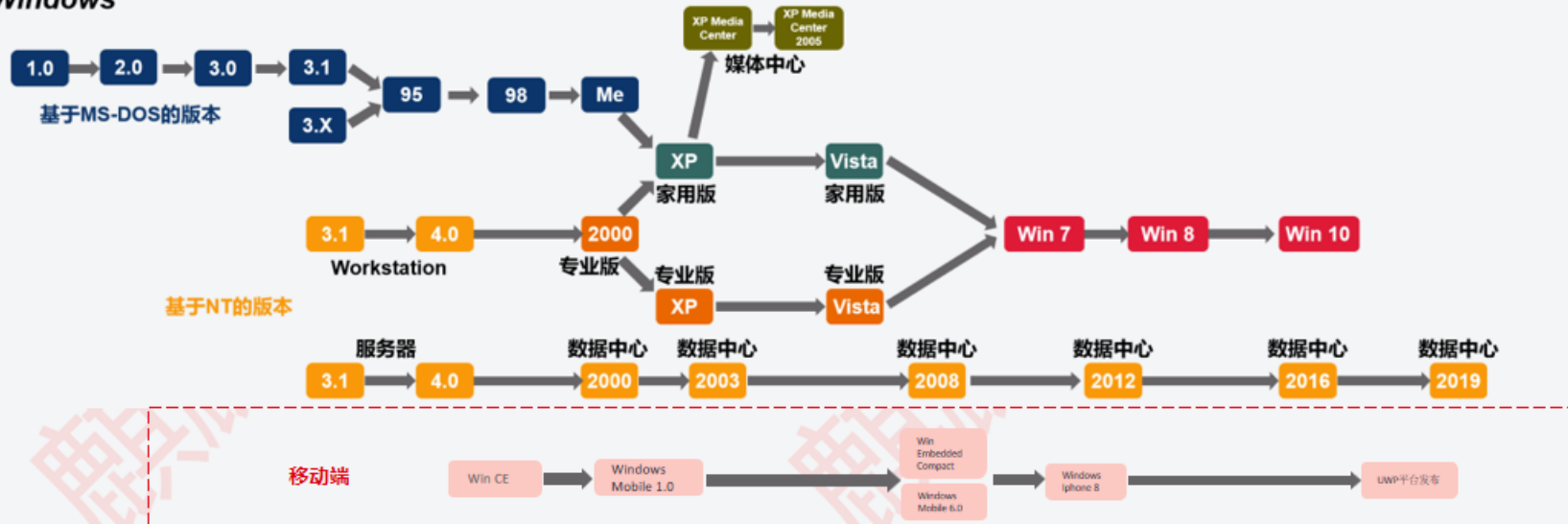


2016—至今 物联终端时代

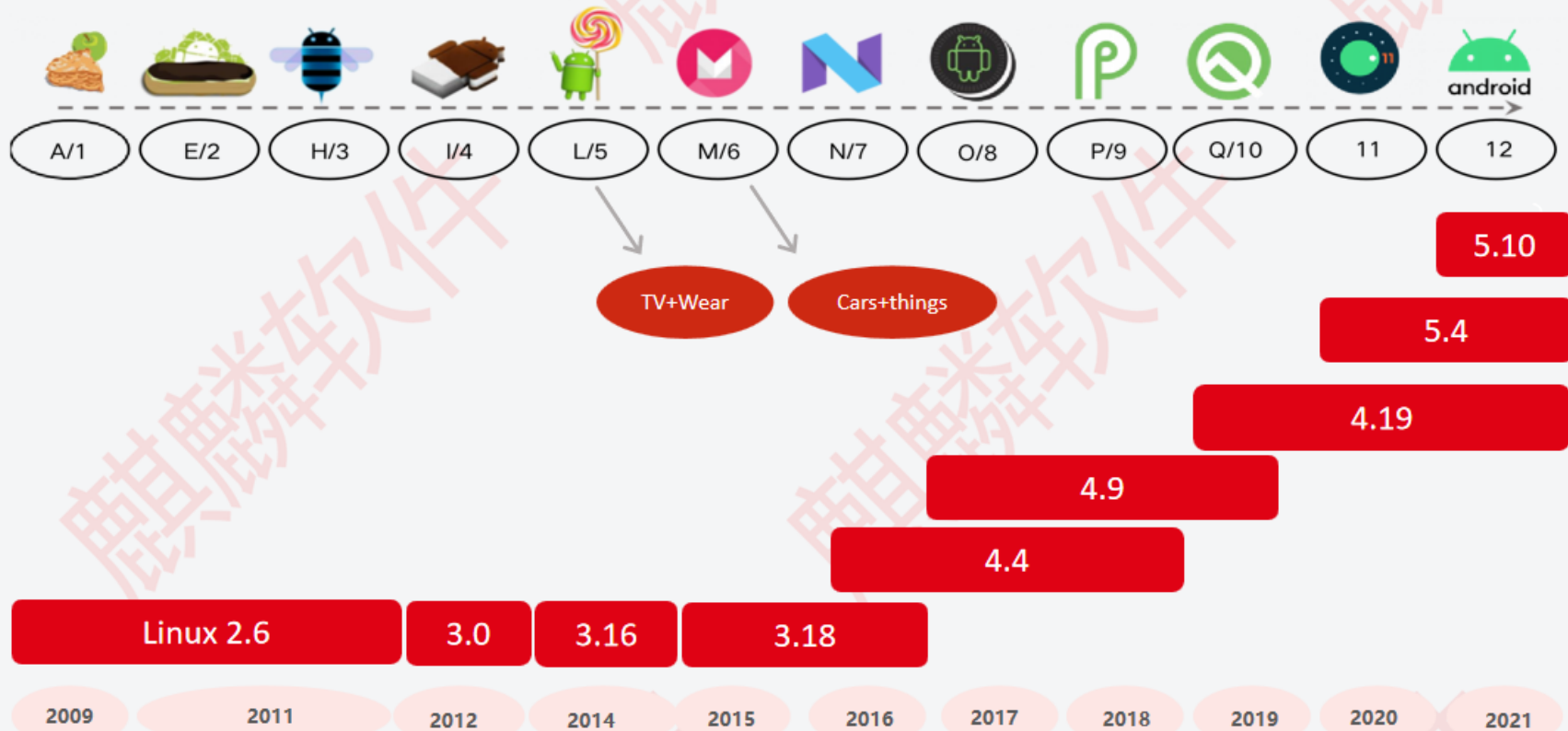
1、2016年谷歌发布物联Fuchsia操作系统, 打通智能多终端场景;
2、2019年华为发布了基于物联网的鸿蒙操作系统HarmonyOS 1+8+N;
3、同期物联网OS快速发展
(1) 传统嵌入式系统+通信协议+其他物联网功能模块;
(2) 基于 Linux、移动端等成熟操作系统的裁剪;
(3) 面向物联网的轻量级 IoT OS;

Windows 移动端发展轨迹

Windows



谷歌Android发展轨迹



苹果IOS发展轨迹

基于网络+芯片+硬件发展，驱动OS不断迭代



- 1 智能终端操作系统概念定义
- 2 智能终端操作系统价值重要性
- 3 智能终端操作系统分类
- 4 智能终端操作系统发展轨迹
- 5 智能终端操作系统当前挑战及趋势

智能终端OS的当前面临挑战



万物智联发展趋势



附相关基础概念

- **MCU**：微控制单元（Microcontroller Unit），又称单片微型计算机（Single Chip Microcomputer）或者单片机，是把中央处理器（CPU）的频率与规格做适当缩减，并将内存（memory）、计数器（Timer）、USB、A/D转换、UART、PLC、DMA等周边接口，甚至LCD驱动电路都整合在单一芯片上，形成芯片级的计算机，为不同的应用场合做不同组合控制。
- **SoC**：系统级芯片或片上系统（System on a Chip），国内外学术界一般倾向将SoC定义为将微处理器、模拟IP核、数字IP核和存储器（或片外存储控制接口）集成在单一芯片上，它通常是客户定制的，或是面向特定用途的标准产品。
- **嵌入式系统**（Embedded system）：以应用为中心、计算机系统为基础，软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。
- **操作系统**（Operating System，简称OS）：管理计算机硬件和软件资源的基础软件。操作系统需要处理内存管理、配置系统资源、控制输入输出设备、操作网络与管理文件系统等基本事务。
- **实时操作系统**（Real Time Operating System，简称RTOS）：操作系统的一种，系统保证在一定时间限制内完成。实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成操作；软实时则只要按照任务的优先级，尽可能快地完成操作即可。
- **物联网操作系统**（IoT OS）：目前对于IoT OS并无明确的统一定义，可以认为其是在嵌入式基础上，面向物联网开发和功能实现需求（低功耗、通讯协议、二次开发、云端对接、高可靠等），使嵌入式系统具备网络连接和通信能力的操作系统总称。
- **宏内核**（Monolithic kernel）：操作系统核心架构的一种，特性是整个核心程序都是以核心空间（Kernel Space）的身份及监管者模式（Supervisor Mode）来运行。用户服务和内核服务运行在同一地址空间中。

附相关基础概念

- **微内核** (Micro kernel) : 仅提供操作系统的最核心功能, 如进程调度、进程间通信、存储管理等, 而诸如文件管理、网络支持等服务以接口形式和微内核通信。因此与宏内核相比尺寸小、可扩展性好。
- **进程间通信** (Interprocess communication, IPC) : 程序运行中, 每个进程各自有不同的用户地址空间, 任何一个进程的全局变量在另一个进程中都看不到, 所以进程之间要交换数据必须通过内核, 在内核中开辟一块缓冲区, 进程1把数据从用户空间拷到内核缓冲区, 进程2再从内核缓冲区把数据读走, 这一过程称为进程间通信。
- **系统总线** (System Bus) : 系统总线是连接计算机系统的主要组件, 包括数据总线 (Data Bus)、地址总线 (Address Bus) 和控制总线 (Control Bus)。
- **随机存取存储器** (Random Access Memory, RAM) : 是与CPU等计算单元交换数据的内部存储器。作为操作系统或其他正在运行中程序的临时数据存储介质。RAM具有数据易失性, 在计算机系统中暂时存储程序、数据和中间结果, 断电所存储的数据将随之丢失。
- **只读存储器** (Read-Only Memory, ROM) : ROM存储数据稳定, 断电后所存数据也不会改变, 并且结构较简单, 常用于存储各种固定程序和数据。
- **可移植操作系统接口** (Portable Operating System Interface, POSIX) : POSIX标准定义了操作系统应该为应用程序提供的接口标准, 是IEEE为要在各种UNIX操作系统上运行的软件而定义的一系列API标准的总称。
- **软件开发工具包** (Software Development Kit, SDK) : 为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具集合。

附相关基础概念

- **应用程序接口** (Application Programming Interface, API) : 一组定义、程序及协议的集合, 通过 API接口实现计算机软件之间的 相互通信。开发者无需理解软件内部工作的机制细节, 也无需访问源码即可访问程序实现功能。API可降低系统各部分的相互依赖, 提高组成单元的内聚性, 降低耦合度。
- **集成开发环境** (Integrated Development Environment, IDE) : IDE为程序开发提供开发环境, 包括代码编辑器、编译器、调试器 和用户图形界面等工具, 集成有代码编写功能、分析功能、编译功能、调试功能等一体化开发套件。
- **中间件** (Middleware) : 中间件是独立的系统级软件, 连接操作系统层和应用程序层, 将不同操作系统提供应用的接口标准化, 协议统一化。
- **硬件抽象层** (Hardware Abstraction Layer, HAL) : 介于操作系统内核与硬件电路之间的接口层, 目的在于将硬件抽象化。在隐藏 了特定平台的硬件接口细节后, 为操作系统提供虚拟硬件平台, 使其与硬件无关, 实现多硬件平台移植的目的。
- **图形用户界面** (Graphical User Interface, GUI) : 以图形化形式提供用户与计算机之间的交互, 由相应硬件平台和操作系统来支持显示。
- **谷歌移动服务** (GoogleMobile Service, GMS) : GMS是Google开发并推动Android的动力, 为谷歌play所应用的底层框架, 闭源 运行。GMS提供包括GooglePlay、Search、Gmail、Talk、Maps、Street View、YouTube、Android Market等服务。海外Android 平台严重依赖GMS。
- **安卓开放源代码项目** (Android Open-Source Project, AOSP) : 安卓项目的开放源代码部分, 对应于GMS。

万物智联 星光守护

打造中国操作系统核心力量



400-089-1870



天津市滨海新区塘沽海洋科技园信安创业广场3号楼



www.kylinos.cn

