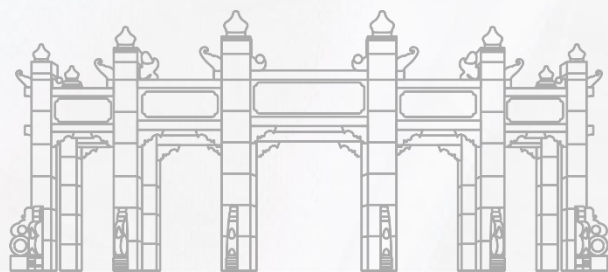
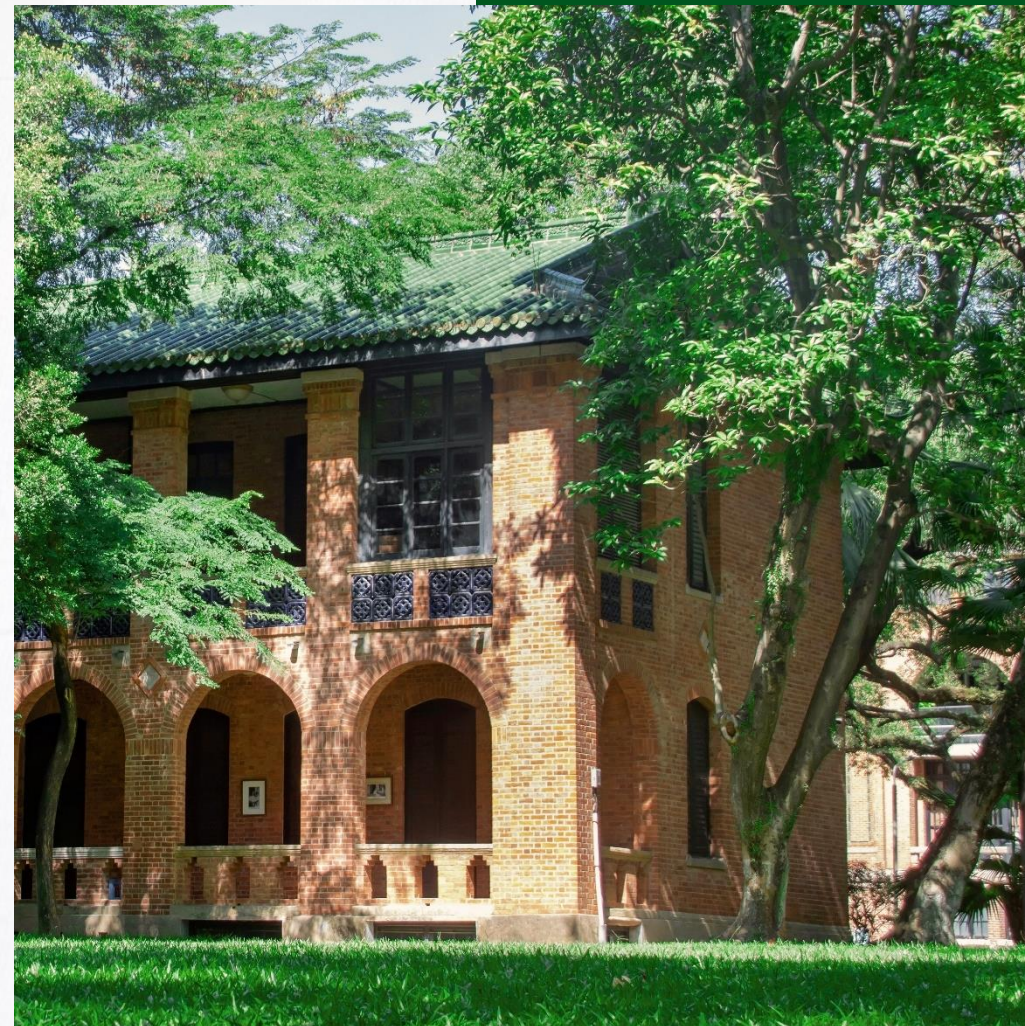


# 鸿蒙鸿蒙操作系统 内核设计



## ● 操作系统的内核

1. 用户最常见到并与之交互的操作系统界面，其实只是操作系统最外面的一层。操作系统最重要的任务，包括管理硬件设备，分配系统资源等，我们称之为操作系统内在最重要的核心功能。而实现这些**核心功能的操作系统模块**，业界一般称之为操作系统“内核”。
2. 操作系统是位于应用和硬件之间的系统软件，向上提供易用的程序接口和运行环境，向下管理硬件资源。内核位于操作系统的下层，**为操作系统上层的程序框架提供硬件资源的并发管理**。

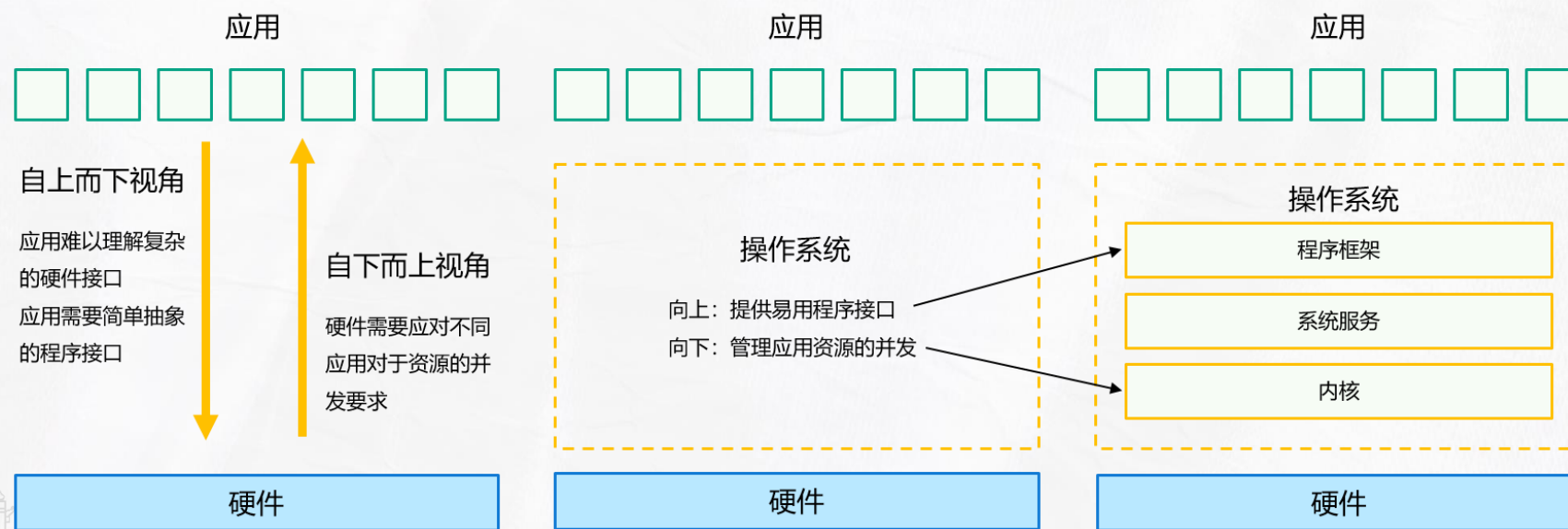


图2-1 操作系统架构



## ● 多内核架构和基本组成

1. 操作系统的内核有很多，但无论是什么内核，基本上有几个最重要的组成单元是每个内核均要具备的。
  - 负责持久化数据，并让应用程序能够方便的访问持久化数据的“文件系统”。
  - 负责管理进程地址空间的“内存管理”。
  - 负责管理多个进程的“进程管理”或者“任务管理”。
  - 负责本机操作系统和另外一个设备上操作系统通信的“网络”。
2. OpenHarmony采用了多内核结构，支持Linux和LiteOS，开发者可按不同产品规格进行选择使用。Linux和LiteOS均具备上述组成单元，只是实现方式有所不同。多个内核通过内核抽象层KAL，向上提供统一的标准接口。
3. 由于OpenHarmony面向多种设备类型，这些设备有着不同的CPU能力，存储大小等。为了更好的适配这些不同的设备类型，内核子系统支持针对不同资源等级的设备选用适合的OS内核，内核抽象层通过屏蔽内核间差异，对上层提供基础的内核能力。

## ● HarmonyOS的整体技术架构

HarmonyOS整体遵从分层设计，从下向上依次为：内核层、系统服务层、框架层和应用层。系统功能按照“系统 > 子系统 > 功能/模块”逐级展开。

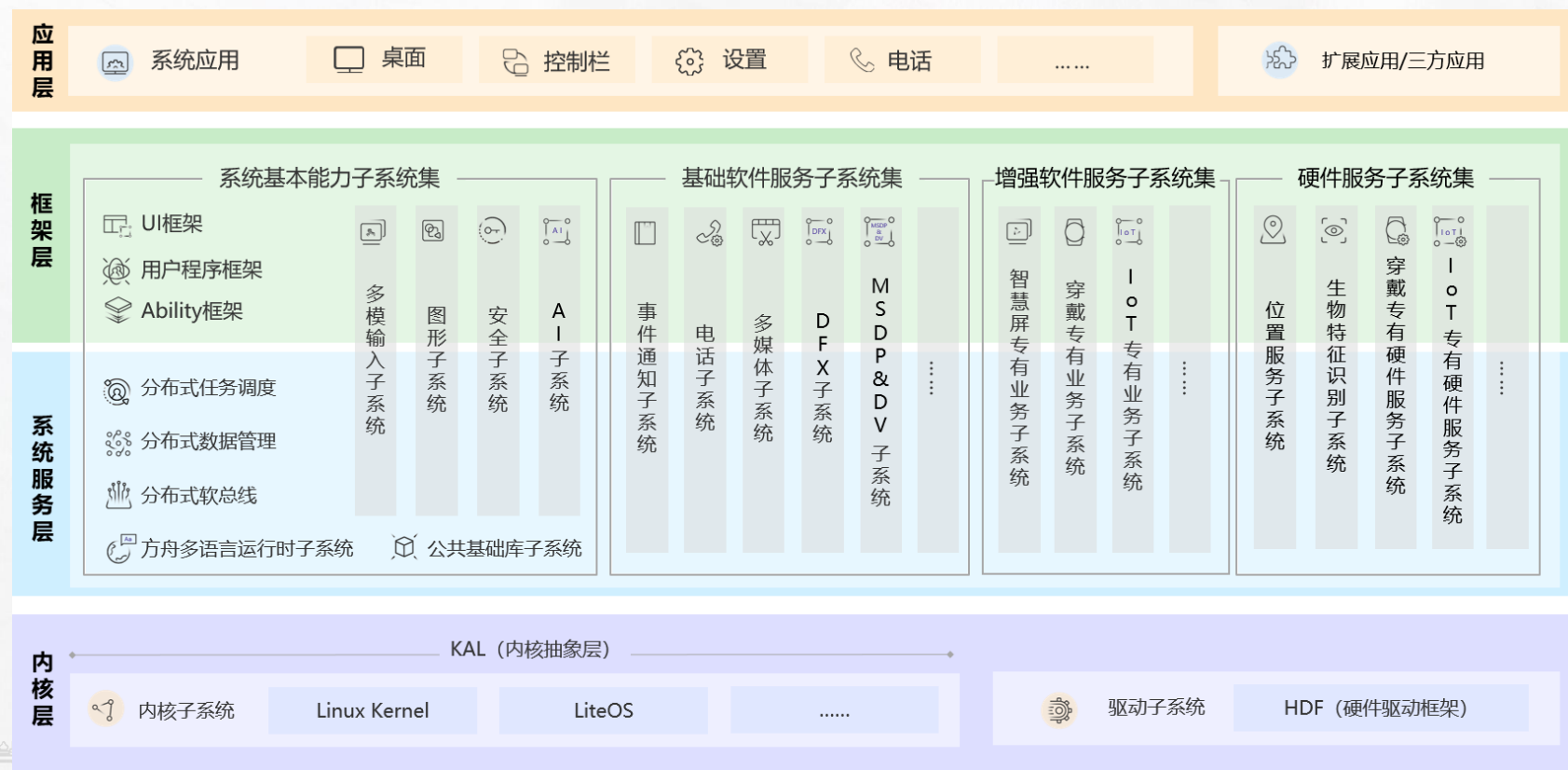


图2-2 HarmonyOS的技术架构

## ● HarmonyOS的整体技术架构

### 1. 内核层

- HarmonyOS采用多内核设计，支持针对不同资源受限设备选用适合的OS内核。如华为自研的LiteOS物联网嵌入式操作系统内核，以及开源的Linux内核等。
- 内核抽象层（KAL，Kernel Abstract Layer）通过屏蔽多内核差异，对上层提供基础的内核能力，包括进程/线程管理、内存管理、文件系统、网络管理和外设管理等。

### 2. 硬件驱动框架（HDF）

- 硬件驱动框架（HDF）是HarmonyOS硬件生态开放的基础，提供统一外设访问能力和驱动开发、管理框架。
- 基于HDF的驱动程序可以不经修改就能跟不同的内核相配合。
- 现有设备的Linux驱动可以在进行很小的修改后移植到HarmonyOS上。



## ● HarmonyOS的整体技术架构

### 3. 系统服务层

- 系统服务层是HarmonyOS的核心能力集合，通过框架层对应用程序提供服务。
- **系统基本能力子系统集**：为分布式应用在HarmonyOS多设备上的运行、调度、迁移等操作提供了基础能力，由分布式软总线、分布式数据管理、分布式任务调度、方舟多语言运行时、公共基础库、多模输入、图形、安全、AI等子系统组成。
- **基础软件服务子系统集**：为HarmonyOS提供公共的、通用的软件服务，由事件通知、电话、多媒体、DFX (Design For X) 等子系统组成。
- **增强软件服务子系统集**：为HarmonyOS提供针对不同设备的、差异化的能力增强型软件服务，由智慧屏专有业务、穿戴专有业务、IoT专有业务等子系统组成。
- **硬件服务子系统集**：为HarmonyOS提供硬件服务，由位置服务、生物特征识别、穿戴专有硬件服务、IoT专有硬件服务等子系统组成。

## ● HarmonyOS的整体技术架构

### 4. 框架层

- 框架层为HarmonyOS应用开发提供了ArkTS/JS/C/C++/Java等多语言的用户程序框架，两种UI框架，以及各种软硬件服务对外开放的多语言框架API。

### 5. 应用层

- 应用层包括系统应用和第三方非系统应用。
- HarmonyOS的应用由一个或多个FA (Feature Ability) 或PA (Particle Ability) 组成。
- FA有UI界面，提供与用户交互的能力。
- PA无UI界面，提供后台运行任务的能力以及统一的数据访问。
- FA在进行用户交互时所需的后台数据访问也需要由对应的PA提供支撑。

## ● HarmonyOS的内核设计

### 多内核结构

- Linux内核：主要用于PC、TV、手机等资源比较丰富的设备。
- LiteOS-m内核：主要用于RAM大小低于1M的极小设备。
- LiteOS-a内核：主要用于RAM大小在1M~128M的物联网设备。

### 内核抽象层KAL (Kernel Abstraction Layer)

- 对于上层应用而言，并不希望内核的不同会影响到上层的开发。
- KAL把屏蔽操作系统底层的差异化，对上层暴露统一接口。



图2-3 HarmonyOS的内核抽象层



## ● HarmonyOS的内核设计

Harmony支持的**系统类型**:

- 轻量系统（mini system）面向MCU类处理器例如 Arm Cortex-M的设备，硬件资源极其有限，支持的设备最小内存为128KiB。可支撑的产品如智能家居领域的连接类模组、传感器设备、穿戴类设备等。
- 小型系统（small system）面向应用处理器例如 Arm Cortex-A的设备，支持的设备最小内存为1MiB，可以提供更高的安全能力、标准的图形框架等。可支撑的产品如智能家居领域的IP Camera、电子猫眼、路由器以及智慧出行域的行车记录仪等。
- 标准系统（standard system）面向应用处理器例如 Arm Cortex-A的设备，支持的设备最小内存为128MiB，可以提供增强的交互能力、3D GPU以及硬件合成能力、更多控件以及动效更丰富的图形能力、完整的应用框架。可支撑的产品如高端的冰箱显示屏。

表2-1系统关系对应表

系统级别	轻量系统	小型系统	标准系统
LiteOS-M	√	×	×
LiteOS-A	×	√	√
Linux	×	√	√

## ● 华为LiteOS内核

Huawei LiteOS是华为**面向IoT领域**，构建的**轻量级物联网操作系统**，可广泛应用于智能家居、个人穿戴、车联网、城市公共服务、制造业等领域。具有高实时性，高稳定性，超小内核，低功耗，支持功能静态裁剪等优势。

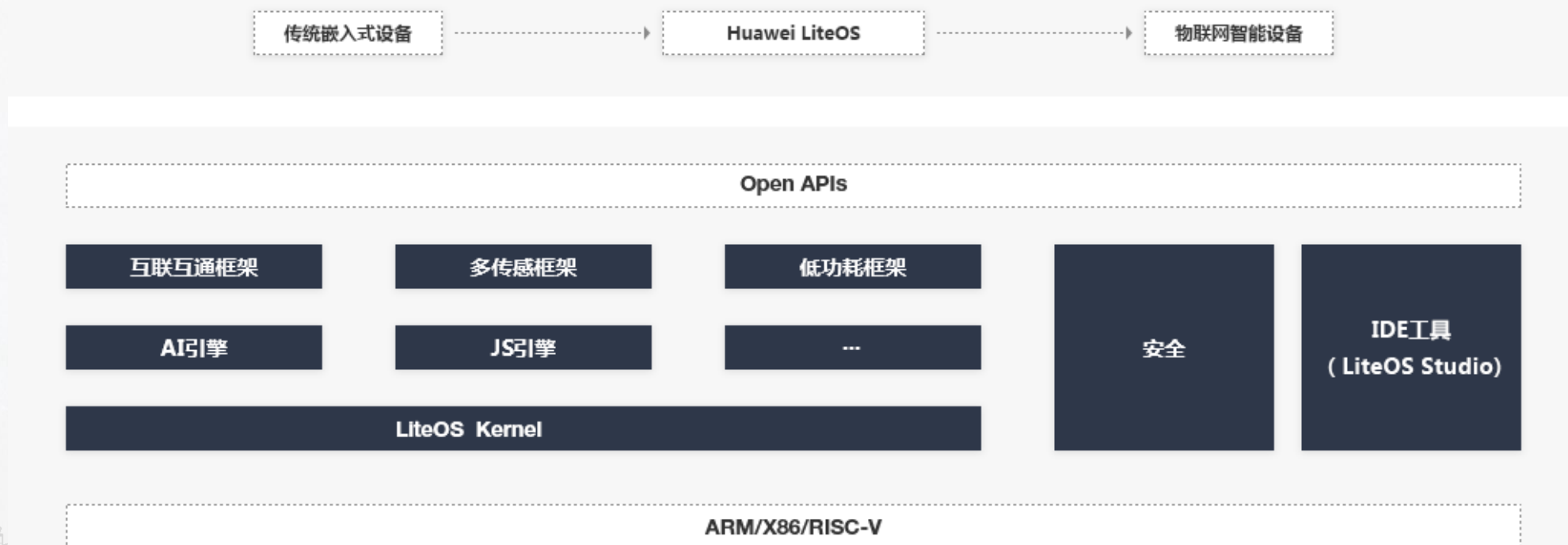


图2-4 LiteOS的产品架构

## ● 华为LiteOS内核

### Huawei LiteOS的功能特性

- **低功耗框架**：LiteOS是轻量级的物联网操作系统，最小内核尺寸仅为6KB，具备快速启动、低功耗等优势，Tickless机制显著降低传感器数据采集功耗。
- **OpenCPU架构**：专为LiteOS小内核架构设计，满足硬件资源受限需求，比如LPWA场景下的水表、气表、车检器等，通过MCU和通信模组二合一的OpenCPU架构，显著降低终端体积和终端成本。
- **安全性设计**：构建低功耗安全传输机制，支持双向认证、FOTA固件差分升级，DTLS、DTLS+等，构建低功耗安全传输机制。
- **端云互通组件**：LiteOS SDK端云互通组件是终端对接到IoT云平台的重要组件，集成了 LwM2M、CoAP、MQTT、mbed TLS、LwIP等全套IoT互联互通协议栈，大大减少开发周期，快速入云。
- **SOTA远程升级**：通过差分方式降低升级包的尺寸，更能适应低带宽网络环境和电池供电环境，经过特别优化差分合并算法，对RAM资源要求更少，满足海量低资源终端的升级诉求。



## ● 华为LiteOS内核

### LiteOS的各个模块

- 任务管理：提供任务的创建、删除、延迟、挂起、恢复等功能，以及锁定和解锁任务调度。支持任务按优先级高低的抢占调度以及同优先级时间片轮转调度。
- 内存管理：提供静态内存和动态内存两种算法，支持内存申请、释放。目前支持的内存管理算法有固定大小的BOX算法、动态申请的bestfit算法和bestfit\_little算法。提供内存统计、内存越界检测功能。
- 硬件相关：提供中断管理、异常管理、系统时钟等功能。中断管理提供中断的创建、删除、使能、禁止、请求位的清除功能。异常管理在系统运行过程中发生异常后，跳转到异常处理模块。Tick是操作系统调度的基本时间单位，对应的时长由每秒Tick数决定，由用户配置。

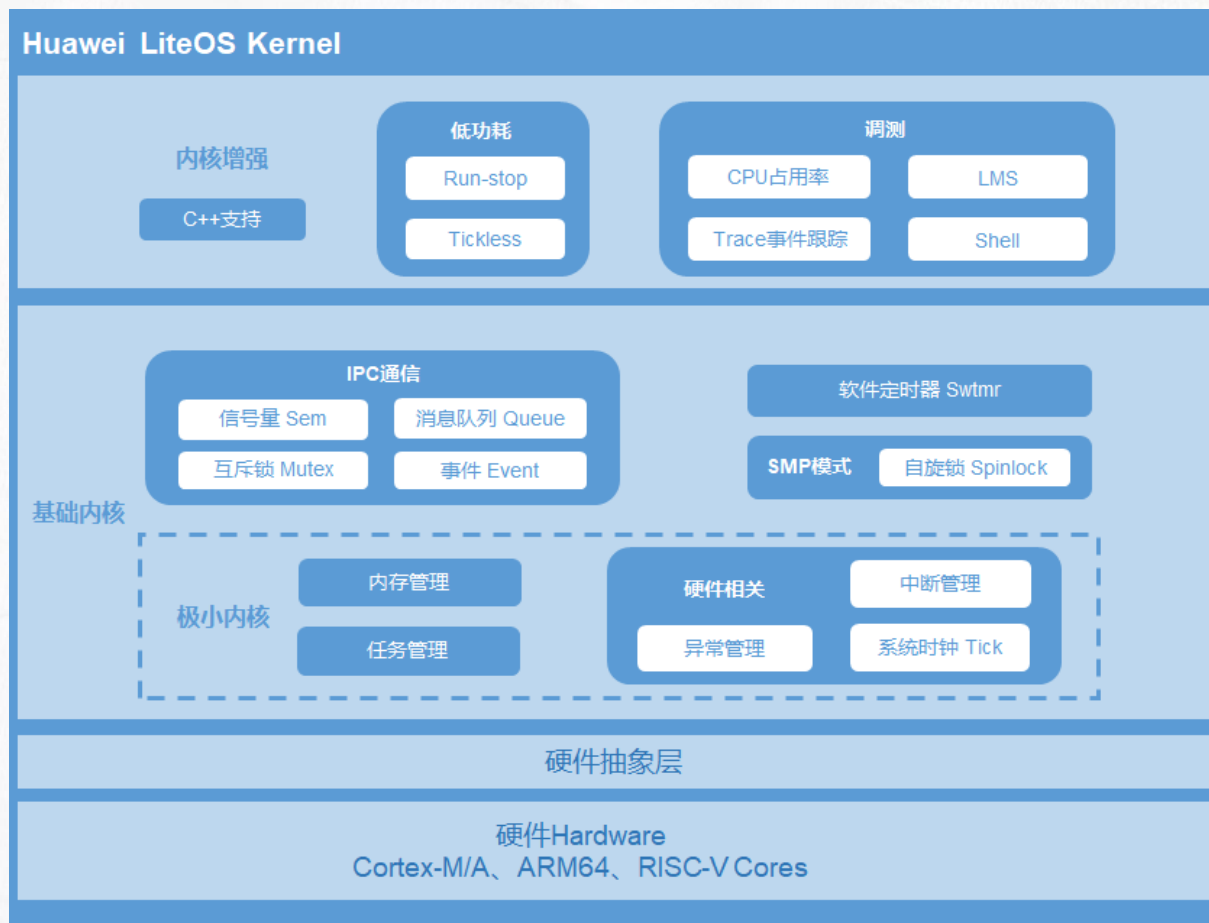


图2-5 LiteOS的内核架构

## ● 华为LiteOS内核

### LiteOS的各个模块

- IPC通信：提供消息队列、事件、信号量和互斥锁功能。支持消息队列的创建、删除、发送和接收功能。支持读事件和写事件功能。支持信号量的创建、删除、申请和释放功能。支持互斥锁的创建、删除、申请和释放功能。
- 软件定时器：提供定时器的创建、删除、启动、停止功能。
- 自旋锁：多核场景下，支持自旋锁的初始化、申请、释放功能。
- 低功耗：Run-stop即休眠唤醒，是Huawei LiteOS提供的保存系统现场镜像以及从系统现场镜像中恢复运行的机制。Tickless机制通过计算下一次有意义的时钟中断的时间，来减少不必要的时钟中断，从而降低系统功耗。打开Tickless功能后，系统会在CPU空闲时启动Tickless机制。

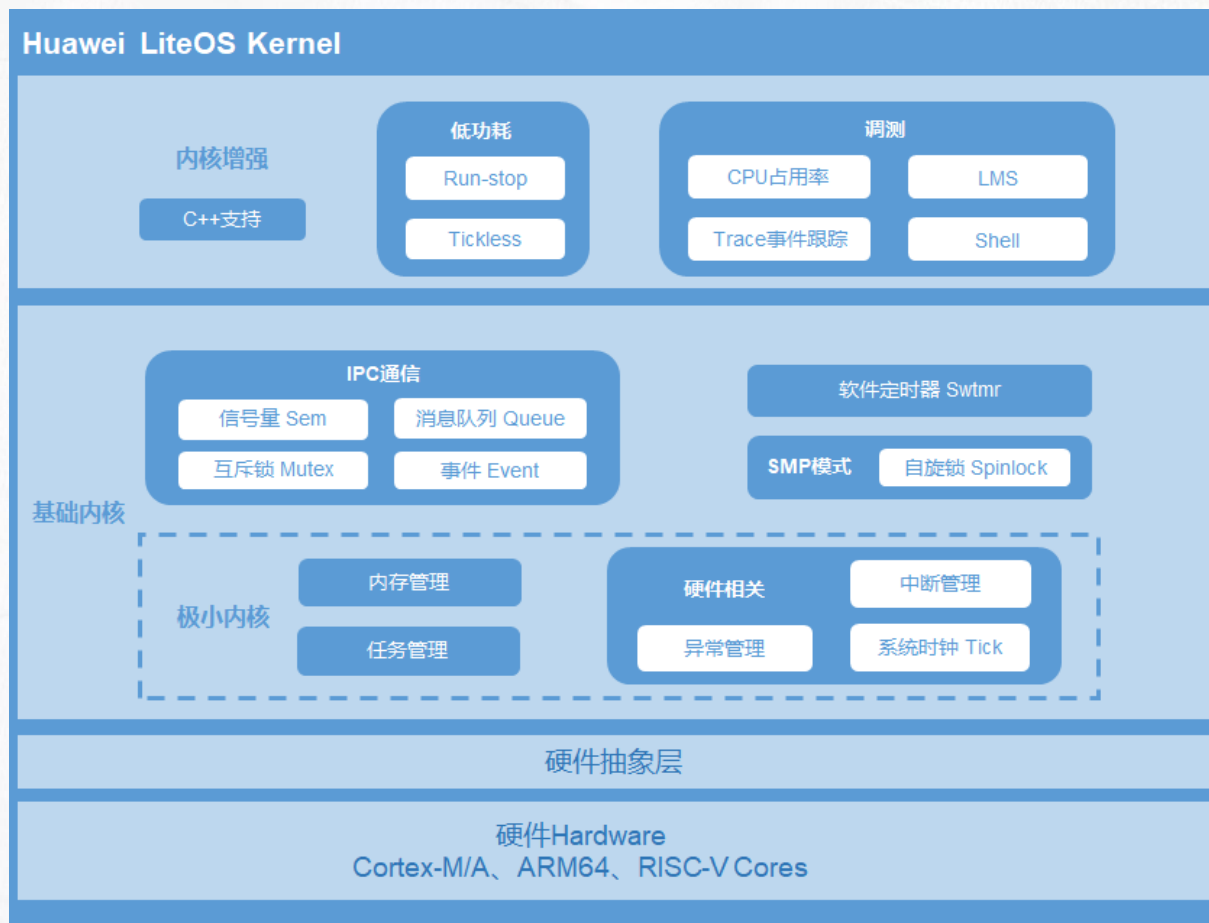


图2-5 LiteOS的内核架构

## ● 华为LiteOS内核

### LiteOS的各个模块

- 维测：可以获取系统或者指定任务的CPU占用率。Trace事件跟踪是实时获取事件发生的上下文，并写入缓冲区，支持自定义缓冲区，跟踪指定模块的事件，开启/停止Trace，清除/输出trace缓冲区数据等。LMS则是实时检测内存操作合法性，LMS能够检测的内存问题包括缓冲区溢出 (buffer overflow)，释放后使用 (use after free)，多重释放 (double free) 和释放野指针 (wild pointer)。Huawei LiteOS Shell使用串口接收用户输入的命令，通过命令的方式调用、执行相应的应用程序。Huawei LiteOS Shell支持常用的基本调试功能，同时支持用户添加自定义命令。
- C++支持：Huawei LiteOS支持部分STL特性、异常和RTTI特性，其他特性由编译器支持。

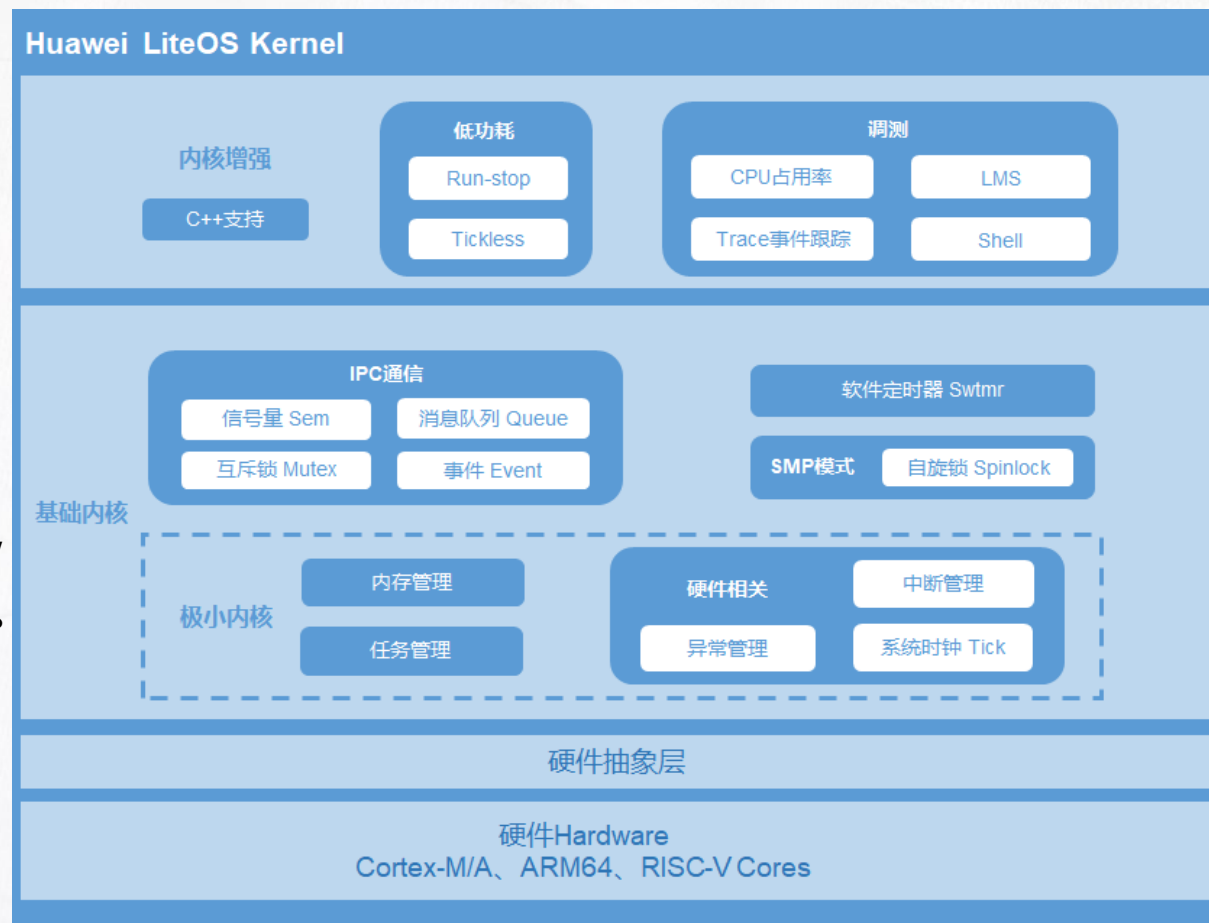


图2-5 LiteOS的内核架构



## ● 华为LiteOS内核

### Huawei LiteOS内核启动流程

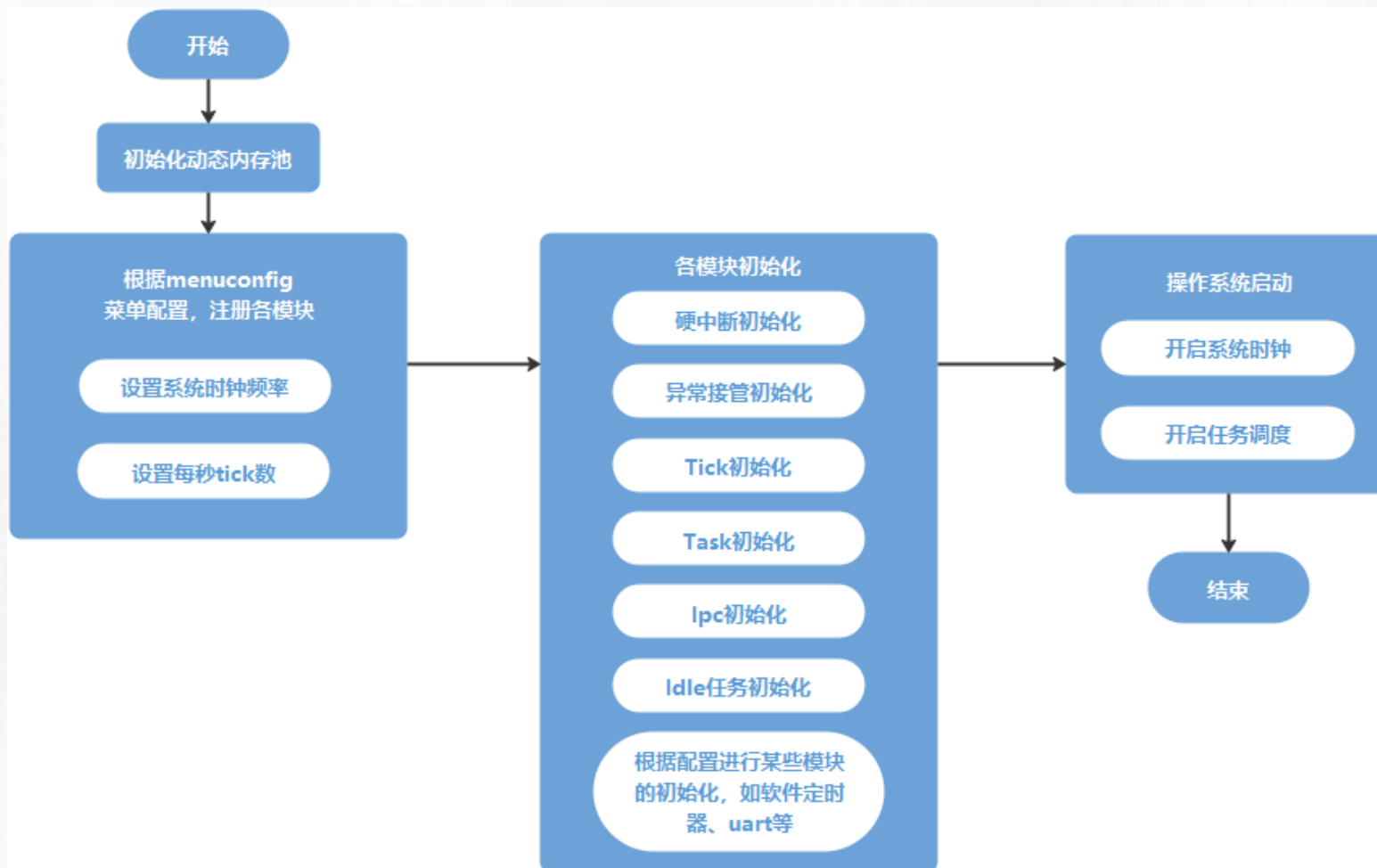


图2-6 LiteOS的内核启动流程

## ● 华为LiteOS内核

### LiteOS-m内核（轻型系统内核）

- 其设计目标是支持极小设备的运行，目标设备的RAM空间一般低于1M。
- 支持基本的任务管理、内存管理、时间管理、中断管理、队列管理、通信机制、事件管理、定时器等操作系统的基础组件。
- 适应低配置、低功耗、低成本的物联网场景。

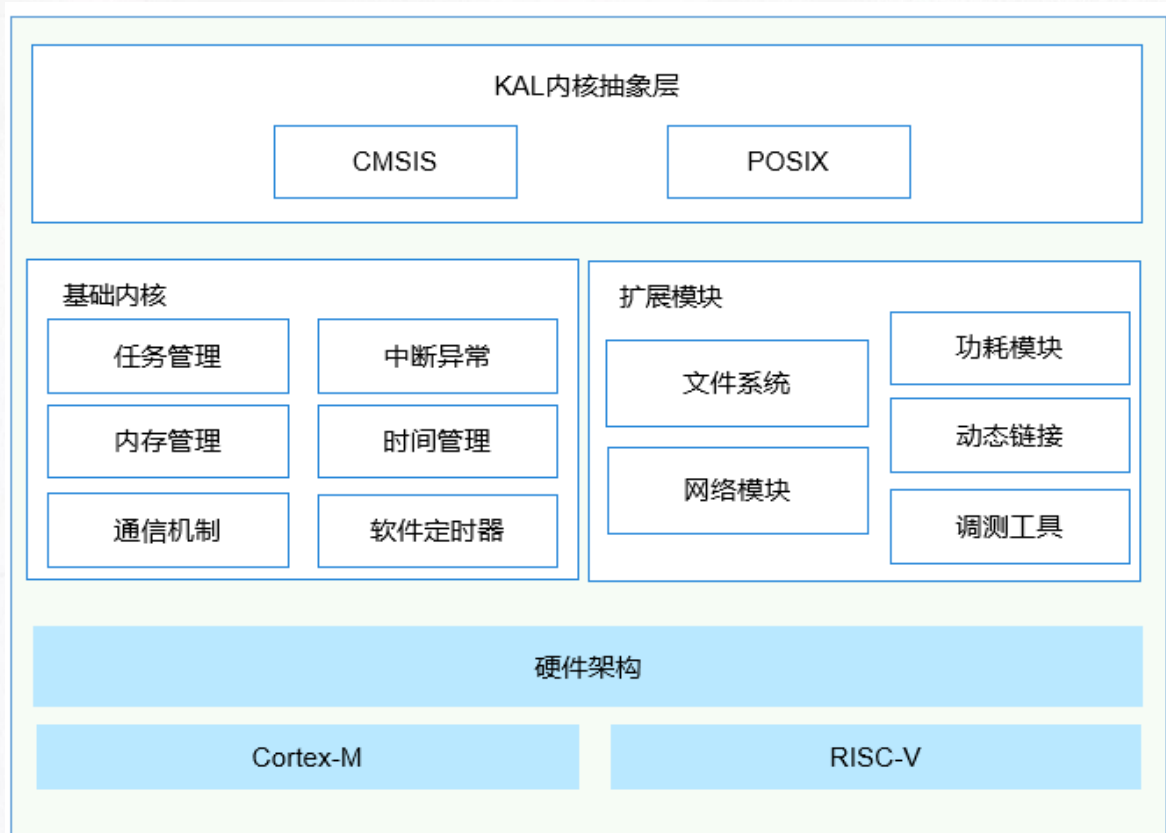


图2-7 LiteOS-M架构图

## ● 华为LiteOS内核

### LiteOS-a内核（小型系统内核）

- 主要用于RAM大小在1M~128M的物联网设备。
- MMU支持。通过MMU支持内核态和用户态分离，支持虚拟单元。
- 支持独立进程。调度对象分别为进程、线程。
- 支持文件系统。包括虚拟文件和块设备等。
- 引入统一驱动框架HDF，为设备厂商提供了更统一的接入方式，使驱动更加容易移植。
- 轻量级内核与硬件高度解耦，新增单板，内核代码不用修改。

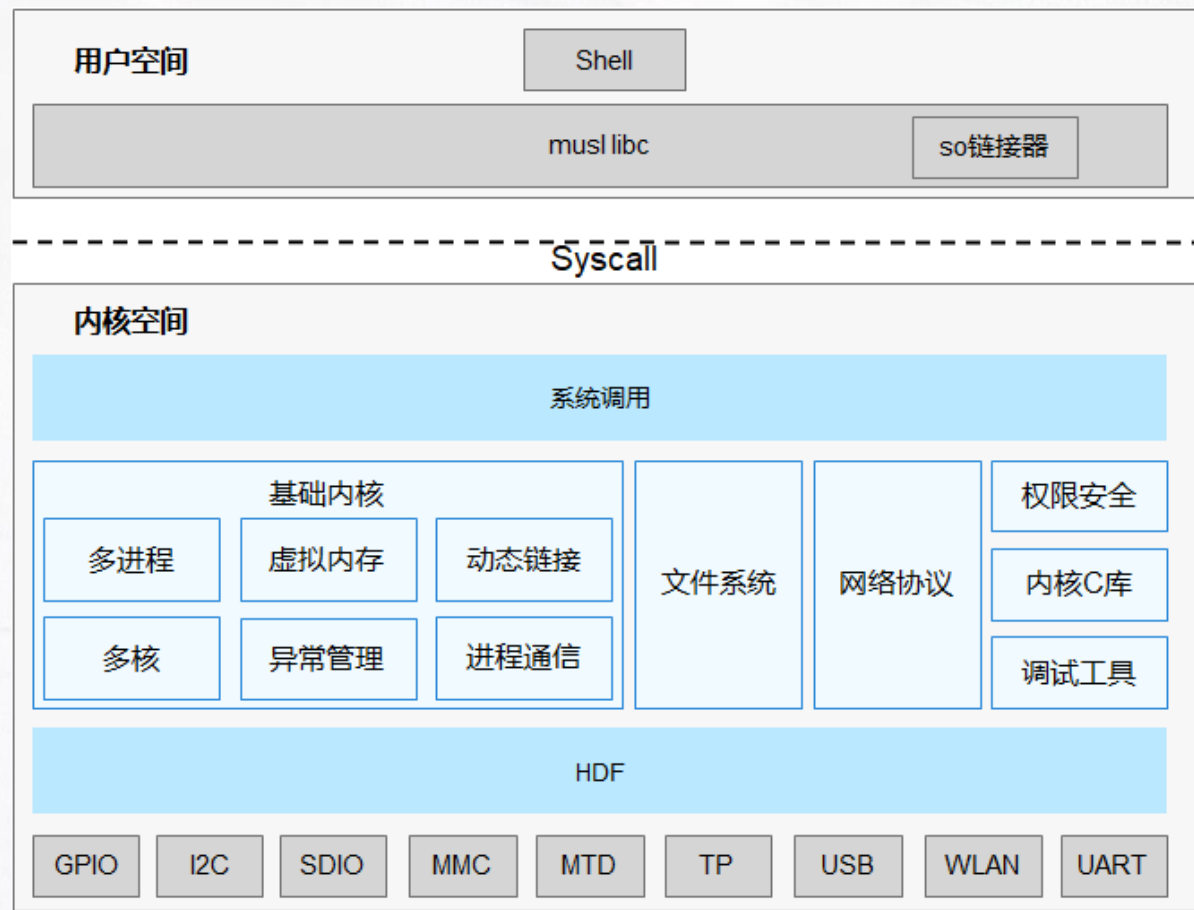


图2-8 LiteOS-A架构图



● 华为LiteOS内核

LiteOS-a与LiteOS-m的区别

表2-2 LiteOS-a与LiteOS-m的区别

项目	LiteOS-m	LiteOS-a
尺寸	128K	128M
调度	抢占式多任务调度	抢占式多线程、多进程调度，FIFO，轮询
虚拟内存	不支持	支持
动态加载	不支持	支持
Shell	不支持	支持
VFS	不支持	支持
多核	不支持	支持
标准库	CMSIS	libc



中山大學  
SUN YAT-SEN UNIVERSITY

# 谢谢观看

SUN YAT-SEN UNIVERSITY