电子发烧友 华秋
www.elecfans.com 电子工程师社区

HarmonyOS技术社区
官 方 战 略 合 作 共 建

HUAWEI

< HDC.Together >
华 为 开 发 者 大 会 2 0 2 0

HarmonyOS轻量内核设计

- HarmonyOS轻量内核设计理念
- HarmonyOS轻量内核关键特性

HarmonyOS轻量内核在架构中的位置

本次主要分享HarmonyOS中轻量内核的设计

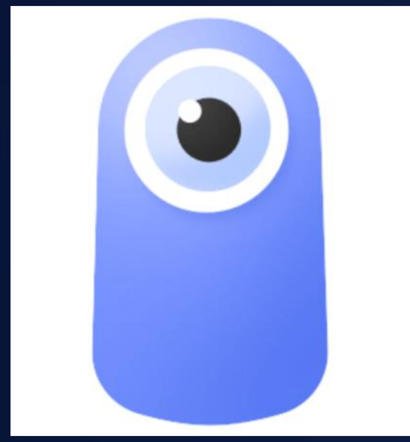
IOT产品发展趋势

硬件

- M核，MPU
- 仅蓝牙&Wifi

功能

联网控制

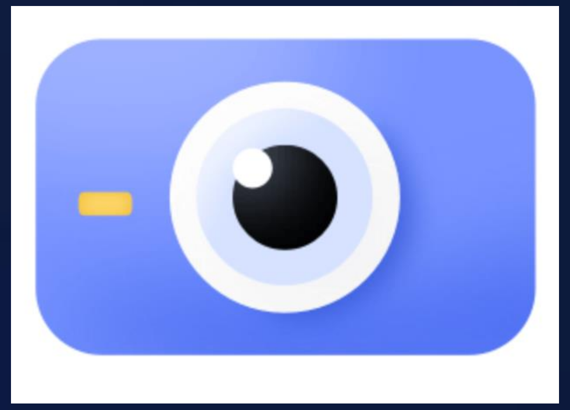


硬件：

- A核， MMU
- 支持摄像头&Wifi

功能：

- 录像、上传到云
- 少量AI



硬件

- 多核， MMU AI加速
- 带屏
- Speaker, MIC

功能

- 触屏交互
- 多媒体
- 多屏互动
- 应用生态

硬件能力越来越强功能越来越丰富

对系统的要求

- 超轻量内核
- 低功耗
- 短距互联协议

→

- 快速启动
- 三方库支持
- 系统组件扩展

→

- 体验：UI交互，多设备联动
- 硬件：支持丰富的外设器件
- 生态：兼容丰富的生态软件

业界IOT设备OS解决方案现状

业界RTOS方案

优点：

- 内存占用小，启动速度快，实时性高；

缺点：

- **生态**：RTOS系统生态不够完善，复杂的驱动和三方库难移植，工作量较大；
- **开发**：应用和内核不能隔离，都在统一内存地址空间，应用代码引起的系统异常，会挂死在内核，导致应用开发与产品开发无法解耦；

业界Linux方案

优点：

- **生态**：驱动和三方库生态软件完善，驱动和三方库容易移植适配；
- **开发**：应用与内核隔离，应用可在自身独立空间中运行，确保不同应用间安全、稳定，可支撑复杂交互体验设备开发；

缺点：

- 内存占用大，启动速度慢，实时性偏弱；

HarmonyOS的轻量内核要实现类Linux的开发体验、RTOS的运行效果

内核设计目标：实现类Linux开发体验、RTOS运行效果

设计目标

①生态软件兼容

支持1000+的Posix接口，复杂GNU/Linux生态软件更易移植；

②内核机制增强

增加多进程、虚拟内存、系统调用，实现应用与应用、内核与应用的隔离，支持复杂交互体验设备开发；

③生态器件统一

HDF统一驱动框架，实现一次开发跨设备、跨系统驱动共享；

组件易开发，保持RTOS资源、实时优势

内核架构

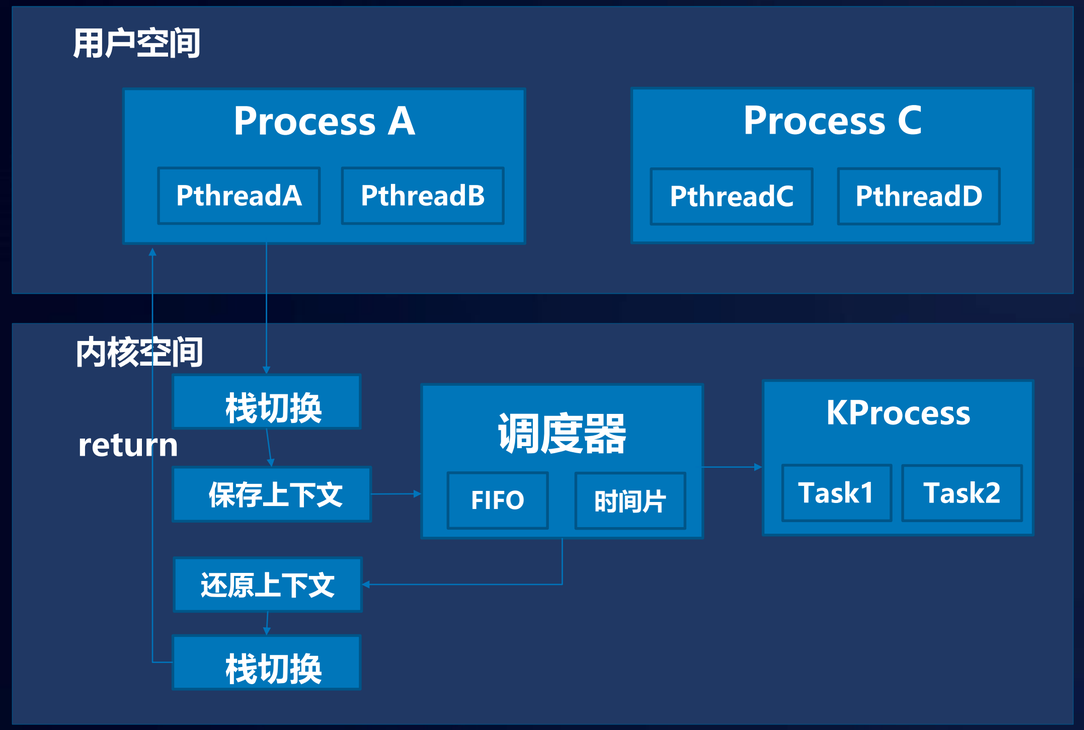
开发者体验变化

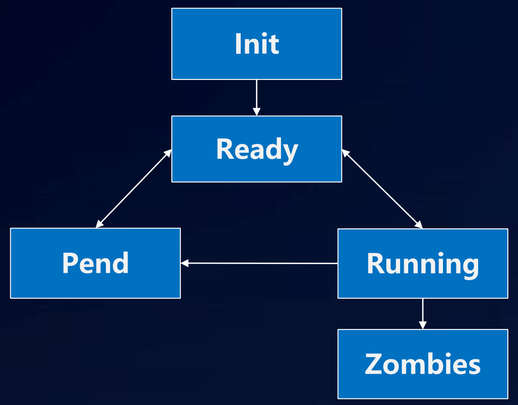
特性&开发生态	RTOS	HarmonyOS内核	带来优势
动态链接	应用和系统一个镜像	系统与应用分离	解耦开发 独立升级 .....
进程隔离	内核容易被业务踩崩溃	业务在用户态，不会把内核踩崩溃	稳定性强 定位容易 .....
开发方式	业务与内核强依赖，要一起编译链接	标准POSIX接口，只需工具链，可编译标准elf文件	开发者开发业务不依赖厂家提供内核 .....
生态软件	三方软件移植困难（C或C++支持不全）	标准POSIX接口，已支持1000+,移植更容易	可利用丰富的开源资源缩短产品研发周期
生态器件	三方器件移植困难	HDF框架，ABI兼容	驱动厂家与ODM厂家工作解耦

关键特性介绍-多进程

HUAWEI

进程切换原理图：

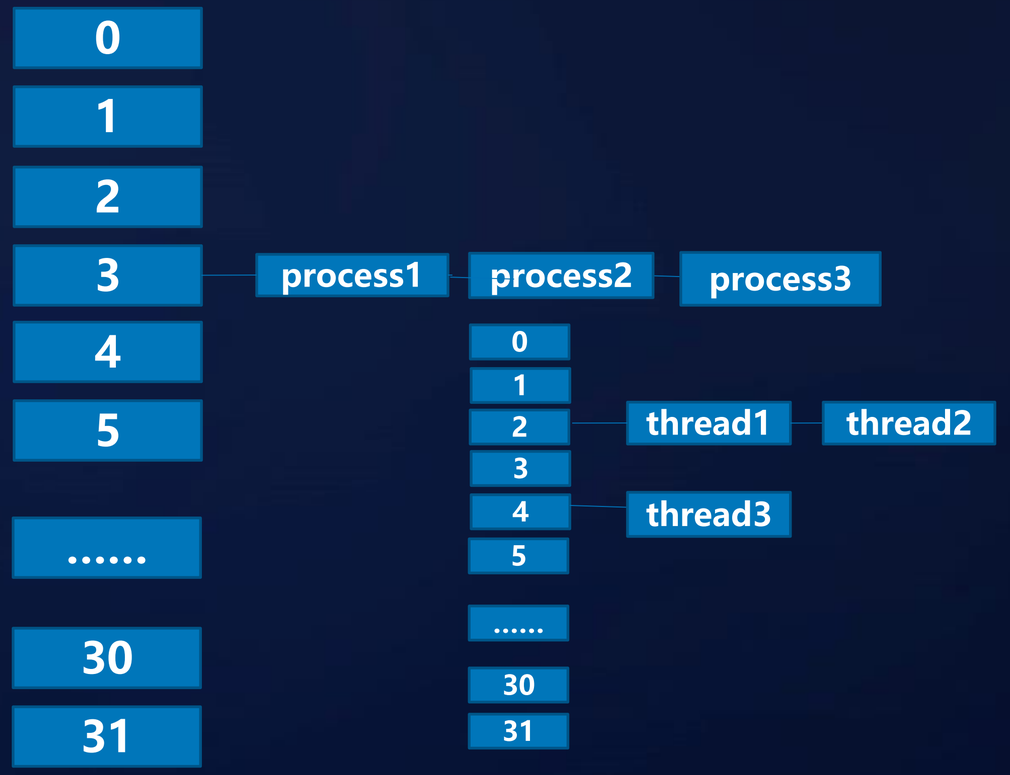




多进程特征：

- **资源隔离**：进程与进程间隔离
- **调试策略**：时间片和FIFO
- **进程和线程优先级**：0-31，0最高，31最低

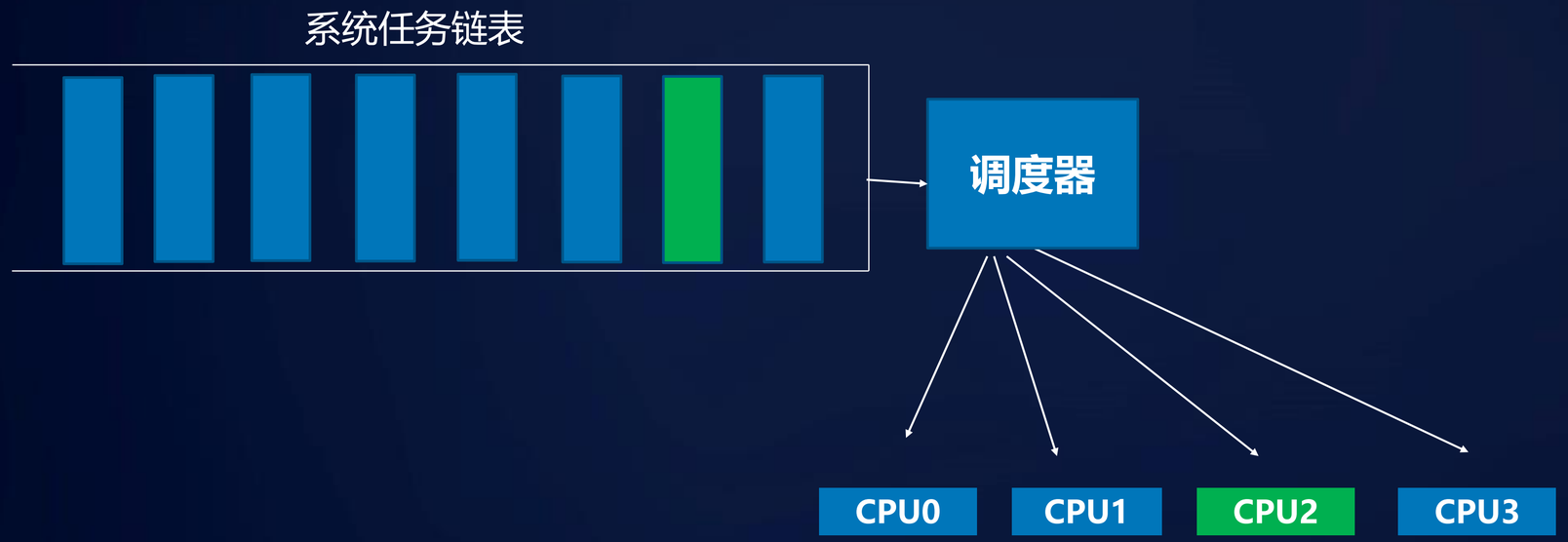
进程与线程调度管理：



实现服务（应用）之间隔离，通过IPC相互访问

关键特性介绍-多核

多核机制原理图：



多核特征：全局链表，所有CPU共享

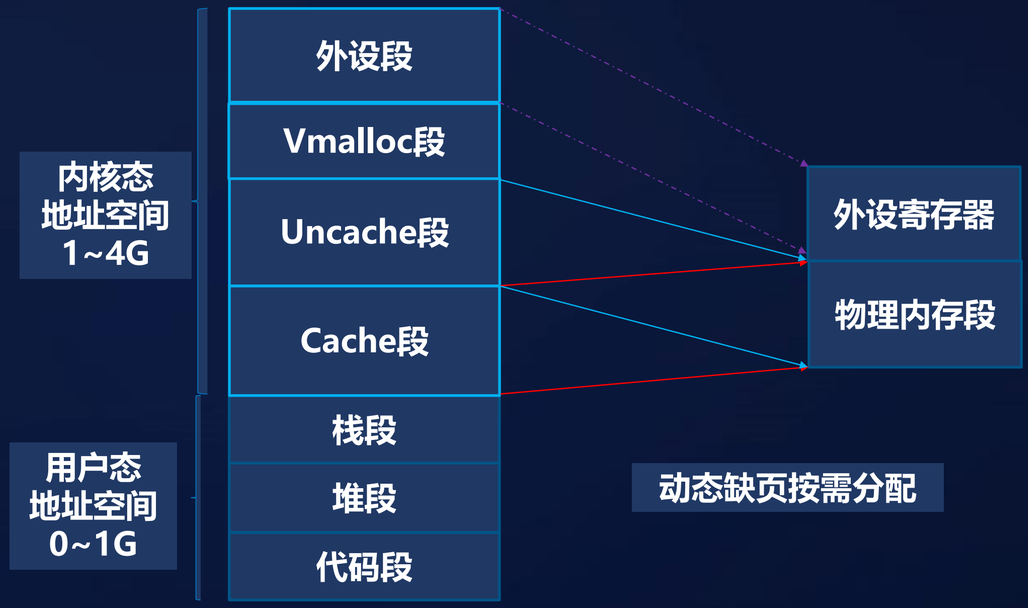
- 调度机制：空闲轮询调度（无负载均衡）；
- 亲核设置：任务和中断支持亲核性设置，可绑核运行。

多线程可并发运行，提升系统效率

关键特性介绍-虚拟内存

内存访问机制原理图：

虚实内存地址空间布局：



虚拟内存特征：

- 内核静态映射：段映射减少页表项，静态映射提升虚实转化效率
- 最优区间分布：0-1G用户态，1-4G内核态，减少用户态进程页表项
- 按需分配：用户态内存通过缺页异常，按需分配

- 服务（应用）地址空间都是0-1G
- 互不干扰
- 物理内存按需申请

关键特性介绍-动态链接

HUAWEI

动态加载链接原理图

动态链接特征：

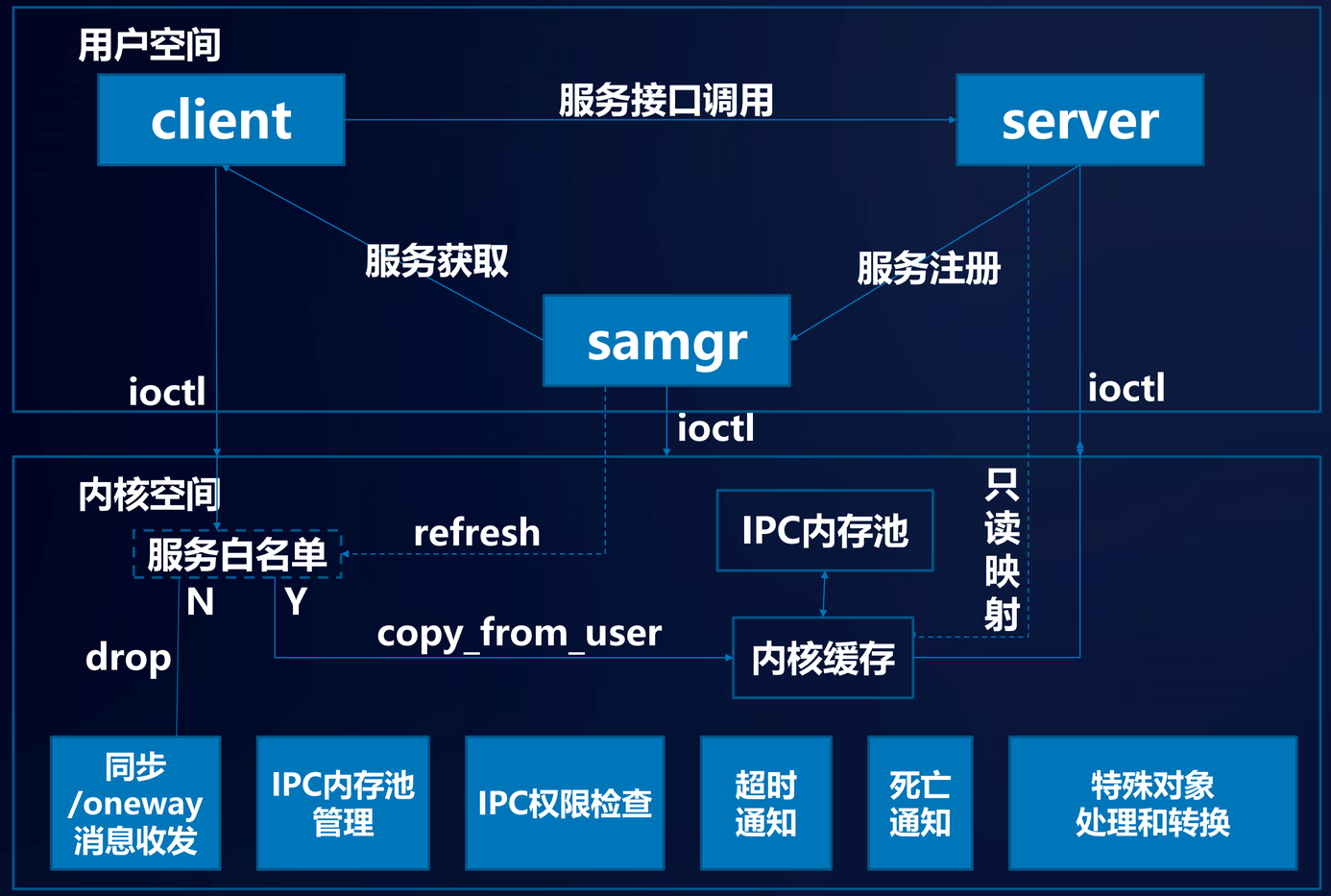
- **按需加载**：多应用共享代码段；加载最小单元为页
- **符号绑定**：支持立即和延迟绑定；
- **加载地址随机化**：进程代码段、数据段、堆栈段地址随机化

标准ELF格式，在Linux交叉编译出来的ELF可以直接在系统中运行

关键特性介绍-进程通信

HUAWEI

LiteIPC原理图：



IPC机制：

LiteIPC、Mqueue、Pipe、Fifo、Signal

LiteIPC特征：

- **轻量实现**：ROM和RAM占用不超过30K
- **安全通信**：基于白名单控制服务端访问权限
- **高性能**：通过内存映射实现单次拷贝

关键特性介绍-系统调用

HUAWEI

系统调用原理图：

系统调用特征：

- VDSO：减少系统调用开销；

测试函数	原生直接调用	增加系统调用	优化后
clock_gettime gettimeofday	1000ns	1500ns	1000ns

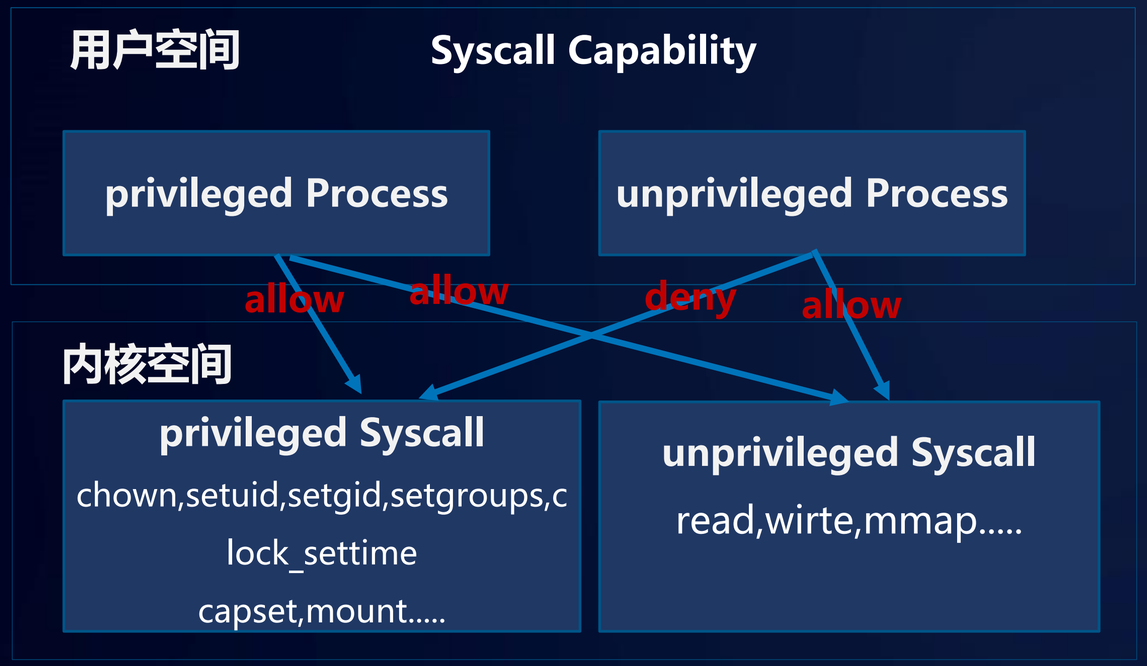
相关数据来源于华为实验室，仅供参考

服务（应用）与内核分离，服务（应用）不能随意访问内核

关键特性介绍-权限管理

HUAWEI

Capability原理图



DAC原理示意

Process	File	DAC
APP1 uid:1 gid:1 group[1]	/data/app1 uid:1 gid:1 rwx:r_x:r_x	app 1 access ok[user] app 2 read ok[group] app 3 read ok[group]
APP2 uid:2 gid:2 group[1,3]	/data/app2 uid:2 gid:2 rwx:___:___	app 1 access fail[other] app 2 access ok[user] app 3 access fail[group]
APP3 uid:3 gid:3 group[1,2,3]	/data/app3 uid:3 gid:3 rwx:r_x:___	app 1 access fail[other] app 2 read ok[group] app 3 access ok[user]

CAPS特征：

- 进程粒度的特权划分和管控；

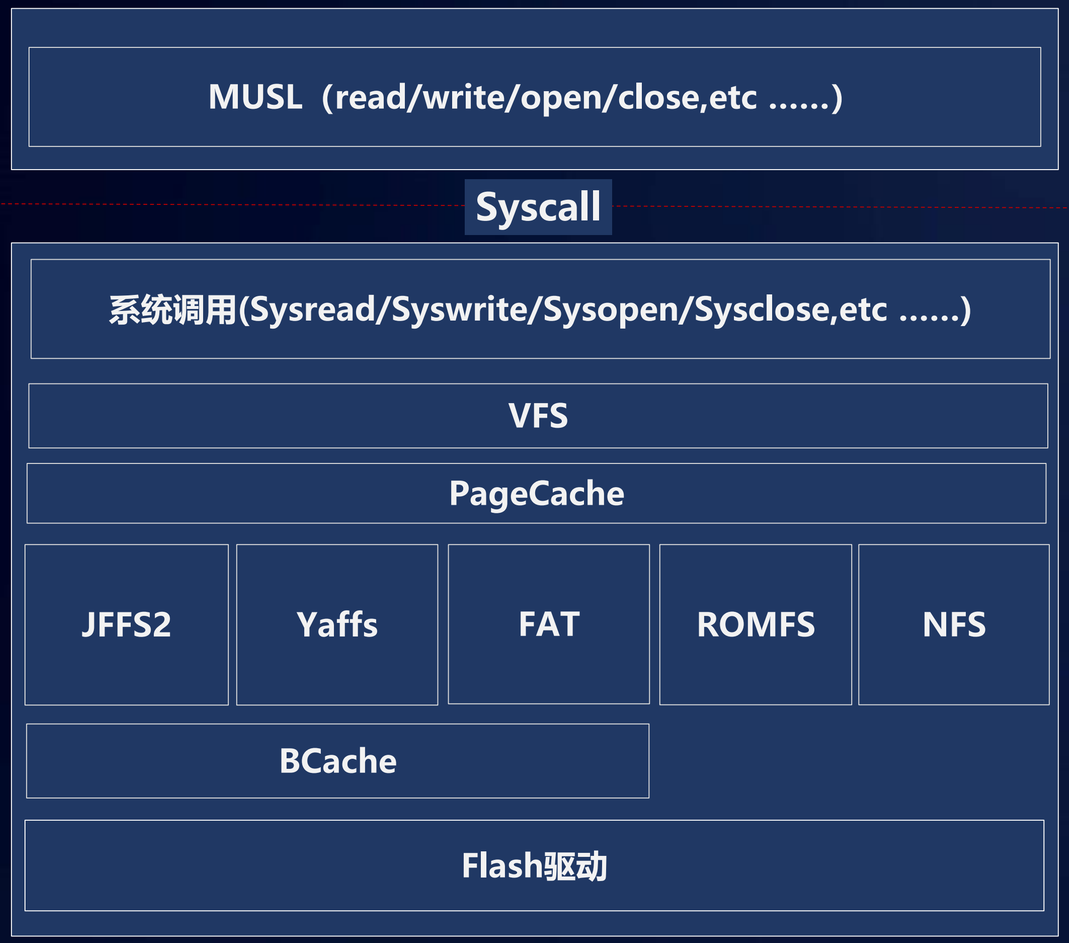
DAC特征：

- 根据进程uid的配置，灵活划分文件资源归属和管控；
- 提供UGO三种权限配置，满足基本的文件共享需求及posix规范

关键特性介绍-文件系统

HUAWEI

文件系统原理图



文件系统目录树

文件系统特征：

- VFS：管理根目录树，挂载点内目录由FS管理；
- BCache和PageCache提升文件系统读写性能

关键特性介绍-POSIX

MUSL 库 官方性能和Size评测

Performance comparison	musl	uClibc	dietlibc	glibc
Tiny allocation & free	0.005	0.004	0.013	0.002
Big allocation & free	0.027	0.018	0.023	0.016
Allocation contention, local	0.048	0.134	0.393	0.041
Allocation contention, shared	0.050	0.132	0.394	0.062
Zero-fill (memset)	0.023	0.048	0.055	0.012
String length (strlen)	0.081	0.098	0.161	0.048
Byte search (strchr)	0.142	0.243	0.198	0.028
Substring (strstr)	0.057	1.273	1.030	0.088
Thread creation/joining	0.248	0.126	45.761	0.142

Bloat comparison	musl	uClibc	dietlibc	glibc
Complete .a set	426k	500k	120k	2.0M †
Complete .so set	527k	560k	185k	7.9M †
Smallest static C program	1.8k	5k	0.2k	662k
Static hello (using printf)	13k	70k	6k	662k
Dynamic overhead (min. dirty)	20k	40k	40k	48k
Static overhead (min. dirty)	8k	12k	8k	28k
Static stdio overhead (min. dirty)	8k	24k	16k	36k
Configurable featureset	no	yes	minimal	minimal

POSIX头文件类型	POSIX 接口个数	Musl 接口个数	Bionic 接口个数
标准C头文件	689	789	744
基本头文件	226	297	342
XSI扩展头文件	82	107	141
其他	215	185	167
汇总	1212	1378	1394

相关数据来源于华为实验室，仅供参考

系统使用C库策略

- 用户态程序：C使用全量Musl，C++使用libc++
- 内核态：使用部分Musl

当前系统支持标准POSIX接口1000+，部分不常用接口待增加

Hello，HarmonyOS

HUAWEI

```
?ubuntu-15:~$ cat hello.c
#include<stdio.h>
int main()
{
        printf("hello,harmony\n");
        return 0;
}
ubuntu-15:~$ arm-linux-musleabi-gcc -o hello hello.c
@ubuntu-15:~$
```

```
OHOS # mount 192.168.1.1:/nfs /user nfs
Mount nfs on 192.168.1.1:/nfs, uid:0, gid:0
Mount nfs finished.
```

```
OHOS # ./hello
OHOS # hello,harmony
OHOS #
OHOS #
```

示例说明

- Linux上开发可执行程序 hello，并拷贝到PC nfs目录
- 单板侧挂载 PC上nfs目录
- 单板侧直接运行./hello

更多鸿蒙技术文章、课程、直播，都在 HarmonyOS社区

欢迎关注HarmonyOS开发者微信公众号