操作系统

L02 OS结构

胡燕
大连理工大学 软件学院

2.0-内容提要

本讲主要内容

本讲主要从操作系统功能设计、接口设计、系统结构三个方面展开重点讨论。

- OS服务 → 操作系统的典型功能，体现为多种OS服务
- 系统调用
- OS结构

2.0-内容提要

本讲主要内容

本讲主要从操作系统功能设计、接口设计、系统结构三个方面展开重点讨论。

- OS服务 → 操作系统的典型功能，体现为多种OS服务
- 系统调用 → 操作系统为上层应用提供的服务接口
- OS结构

2.0-内容提要

本讲主要内容

本讲主要从操作系统功能设计、接口设计、系统结构三个方面展开重点讨论。

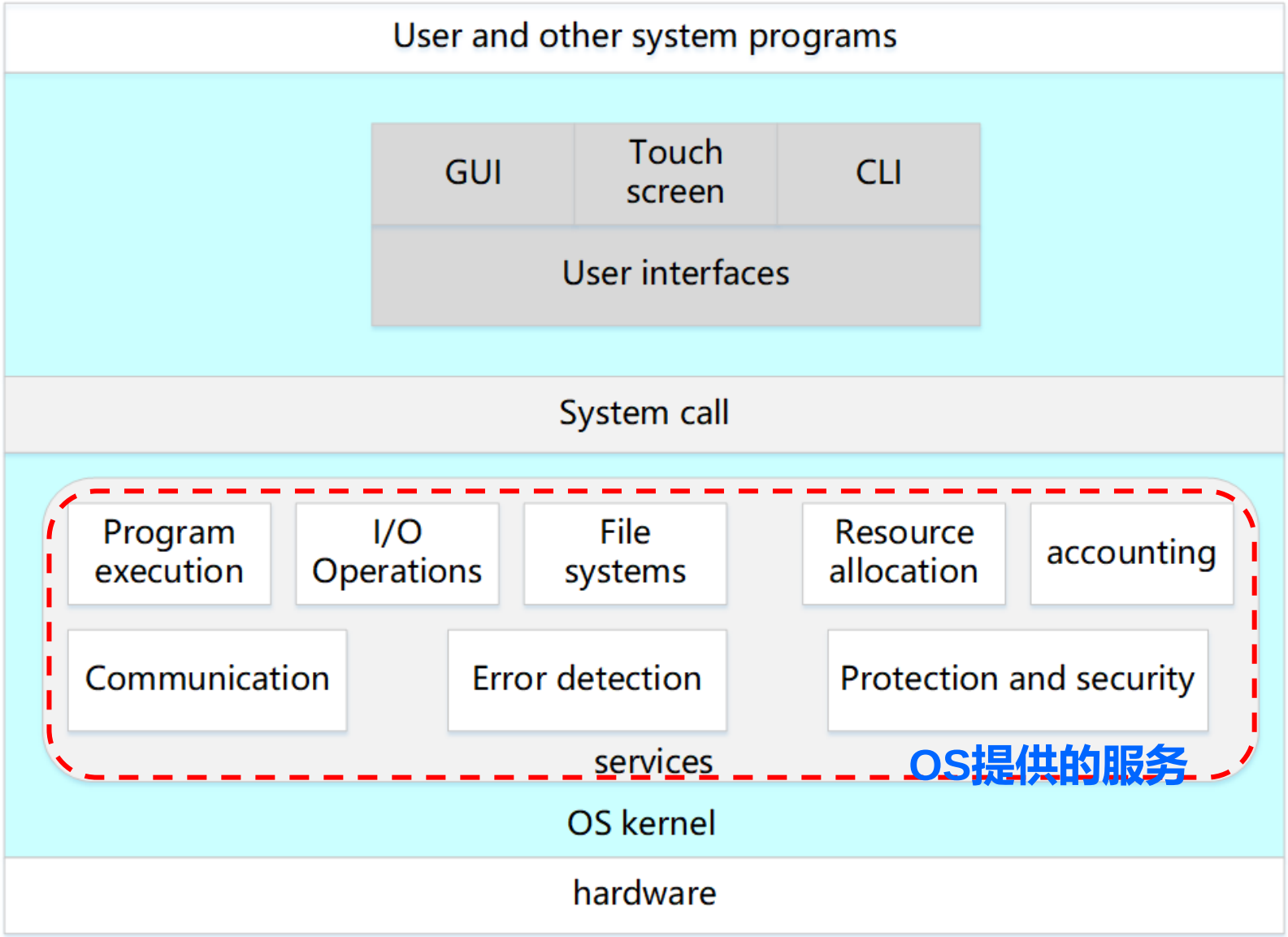
- OS服务 → 操作系统的典型功能，体现为多种OS服务
- 系统调用 → 操作系统为上层应用提供的服务接口
- OS结构 → 操作系统结构设计的发展演变

OS提供的服务

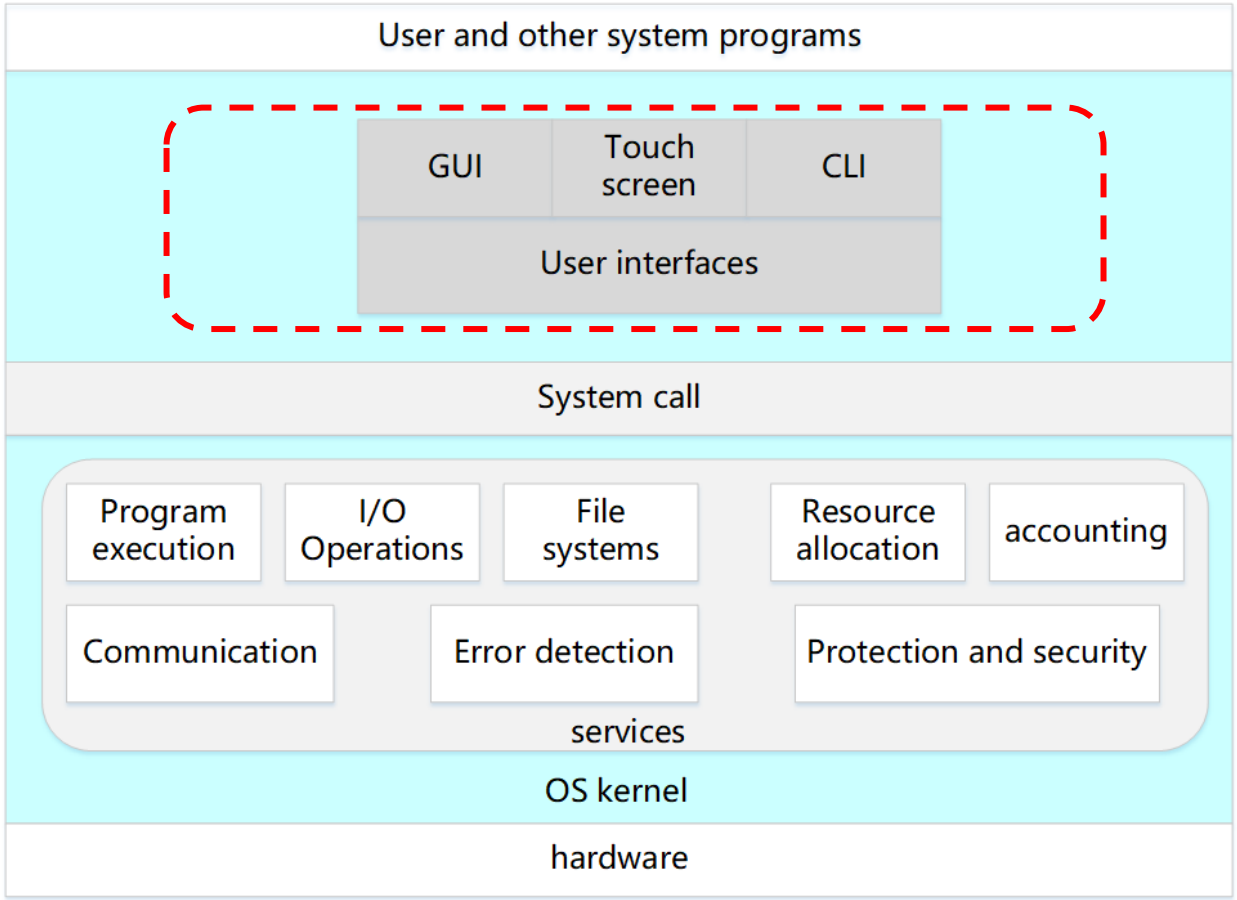
Operating System Services

01

OS提供的服务

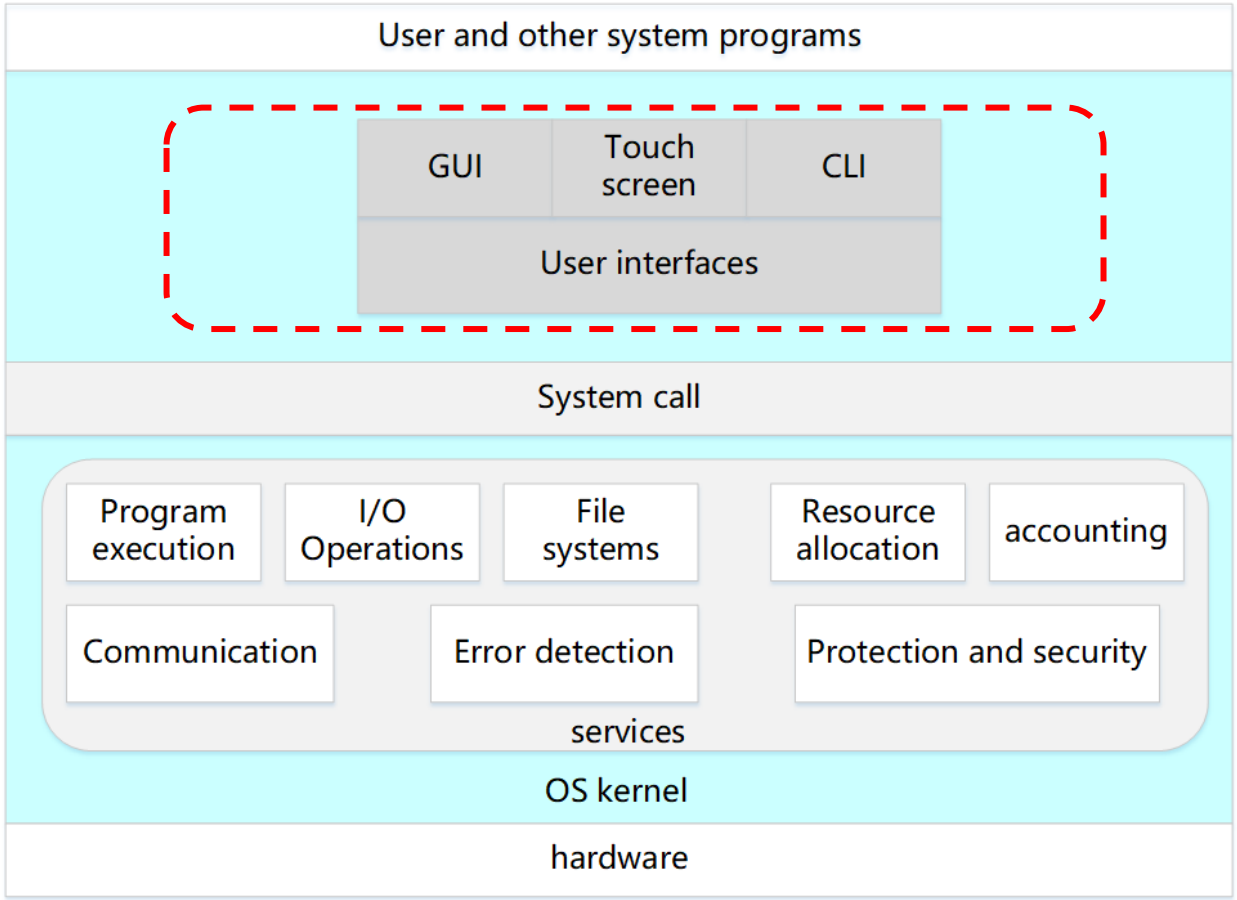

User and other system programs
GUI
Touch screen
CLI
User interfaces
System call
Program execution
I/O Operations
File systems
Resource allocation
accounting
Communication
Error detection
Protection and security
services
OS提供的服务
OS kernel
hardware

OS提供的服务



UI服务

OS提供的服务



UI服务

UI功能通常放在应用层

Windows中，为了提升效率，将基于视窗的UI功能实现放在内核实现，以此提升GUI与内核交互的效率

Linux的GUI服务Xserver，纯粹置于用户态实现

UI: OS的门面.

UI分类：

01

CLI

全称：

Command Line Interface

(命令行界面)

……

02

GUI

全称：

Graphical User Interface

(图形用户界面)

……

03

TSI

全称：

Touch Screen Interface

(触控界面)

……

投票 最多可选1项

设置

在学习的过程中，你使用CLI的属于哪一种情况?

A 从未使用

B 偶尔使用

C 经常使用，但更喜欢GUI

D 经常使用，非常喜欢CLI

提交

命令行的优势：

- 效率 （例如，批量拷贝文件，便于用脚本组织指令，运行开销小）
- 稳定，通用性强 （在类似系统中通用）

命令行的劣势：

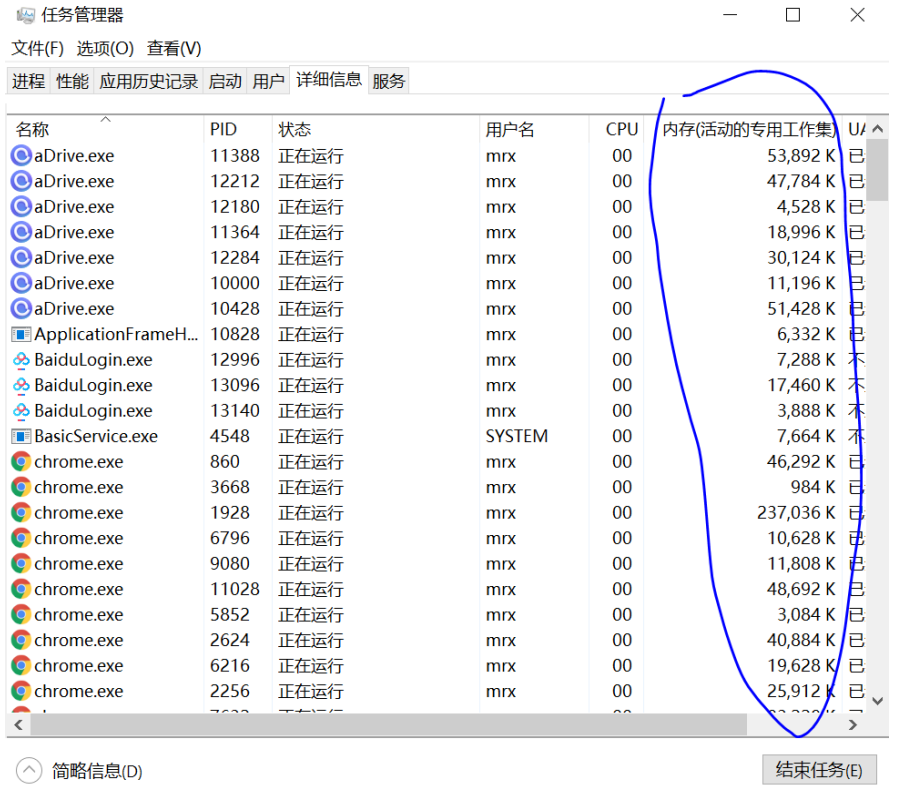
- 对新手不友好 （需要熟练使用较多命令，了解其参数）

服务模块1.2：程序执行

阶段1：磁盘 => 内存.

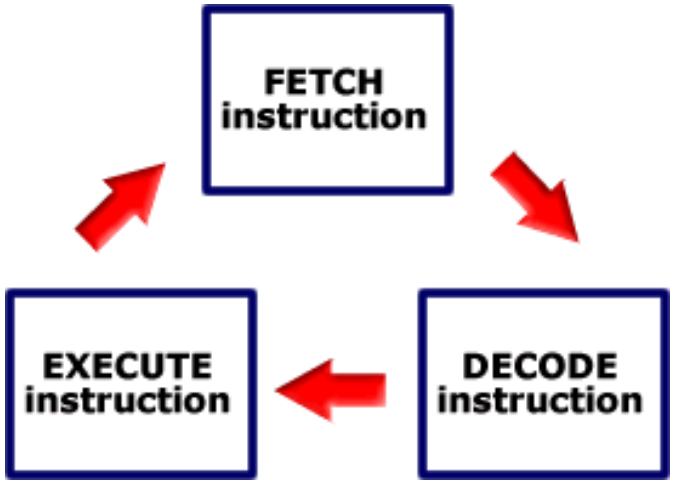
可执行程序（二进制代码）

脚本语言（文本形式，解释执行）



阶段1：磁盘 => 内存.

阶段2：CPU从内存中取指执行.



操作系统为该执行过程提供支持。

服务模块1.2：程序执行

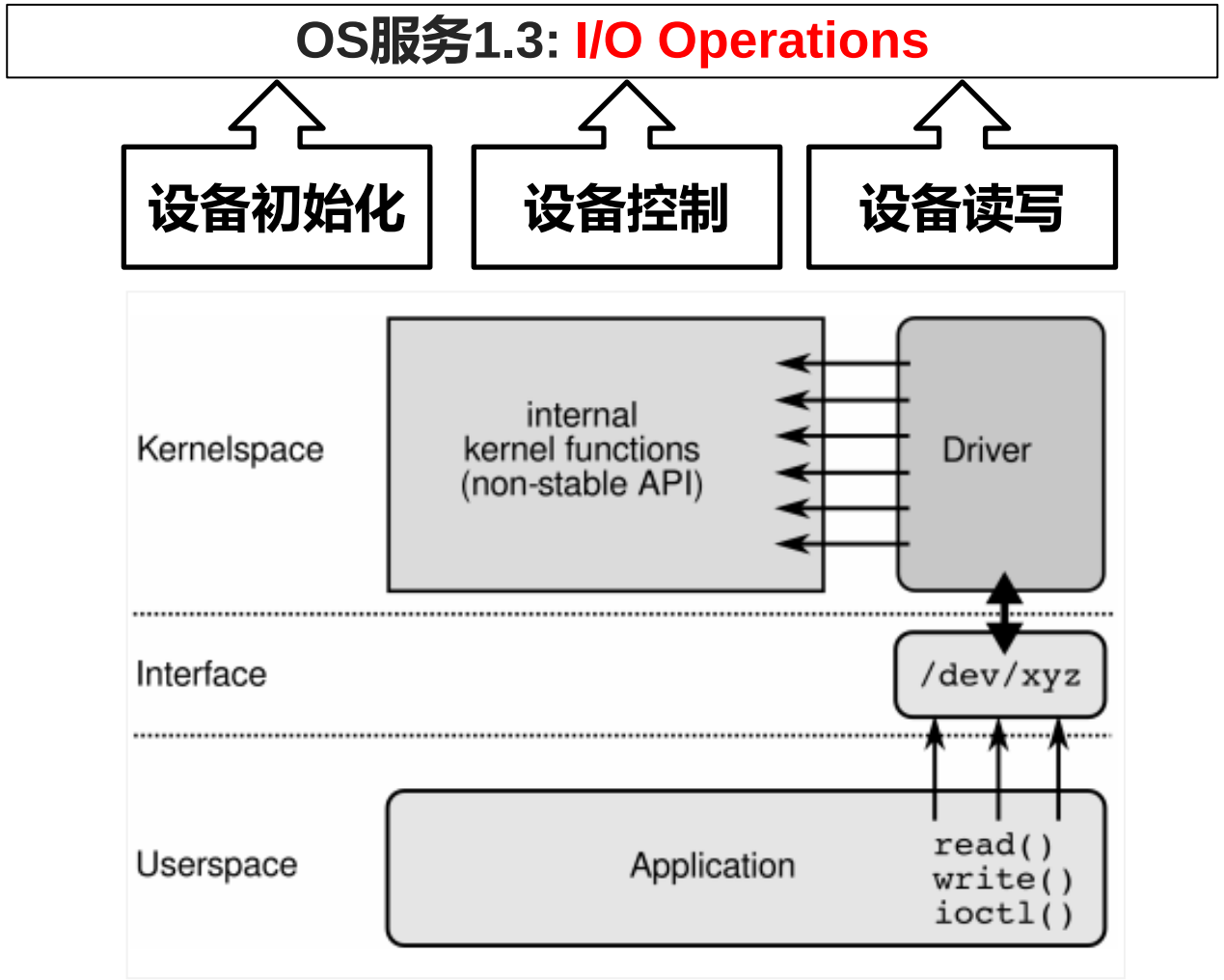
阶段1：磁盘 => 内存.

阶段2：CPU从内存中取指执行.

阶段3：程序退出执行.

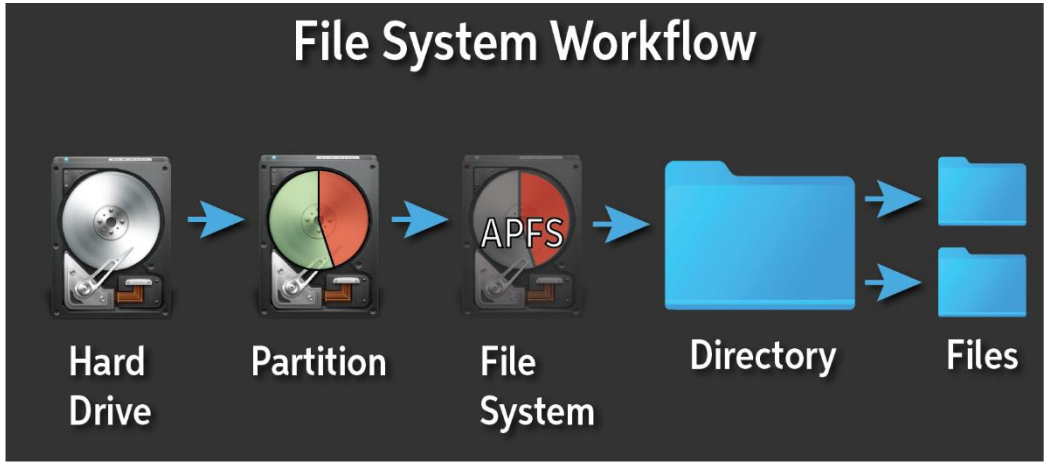
操作系统要在程序退出执行后，进行资源回收。

现代操作系统将I/O操作封装在内核



服务模块1.4：文件系统

现代操作系统通过文件系统来管理各类持久化信息。



文件系统操作：

- 文件命名，以及按照文件名对文件进行访问
- 文件读写
- 目录遍历（Traversal, Search）
- …

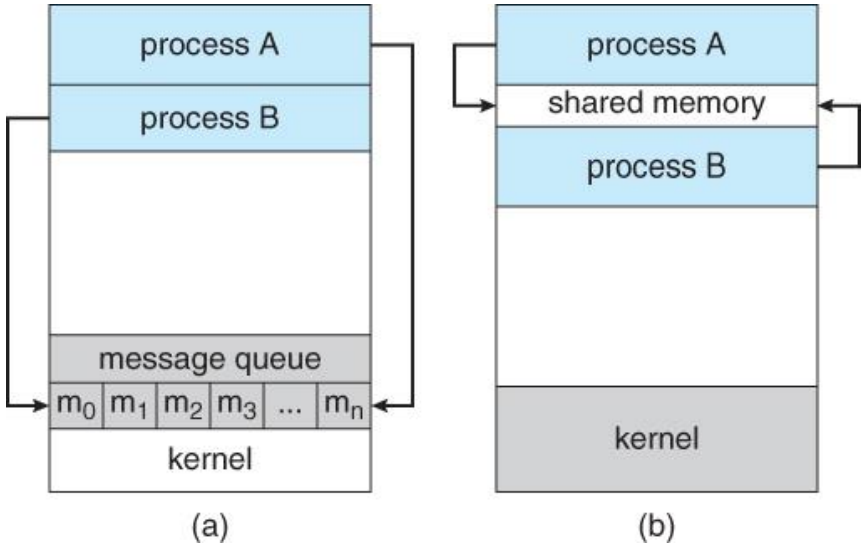
文件系统实现：

- 如何在物理存储介质（如磁盘）上组织文件系统结构
- 如何保证文件访问的性能

通信（Communication）模块对操作系统的意义

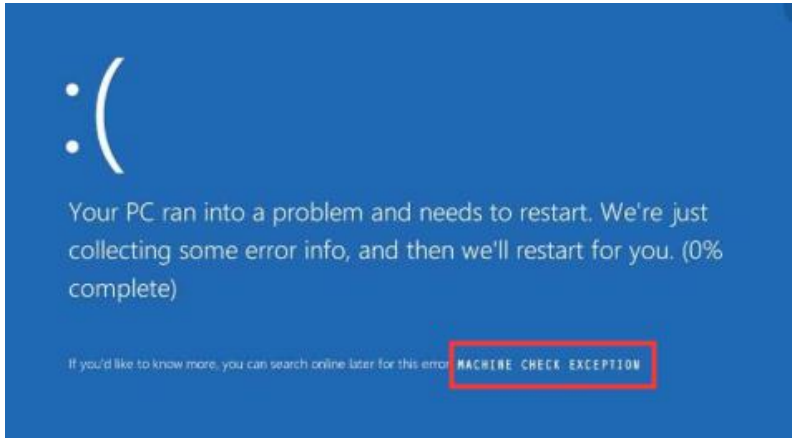
多任务（MultiTasking）
是现代操作系统的基本属性。

任务间通信协同是必要的基本支持。



错误检测与处理机制：

- 硬件错误
  - -Processor machine check exception
  - -Chipset error signals
  - -I/O bus error reporting
  - -I/O device errors
- 软件错误
  - -divide-by-zero
  - -segmentation fault

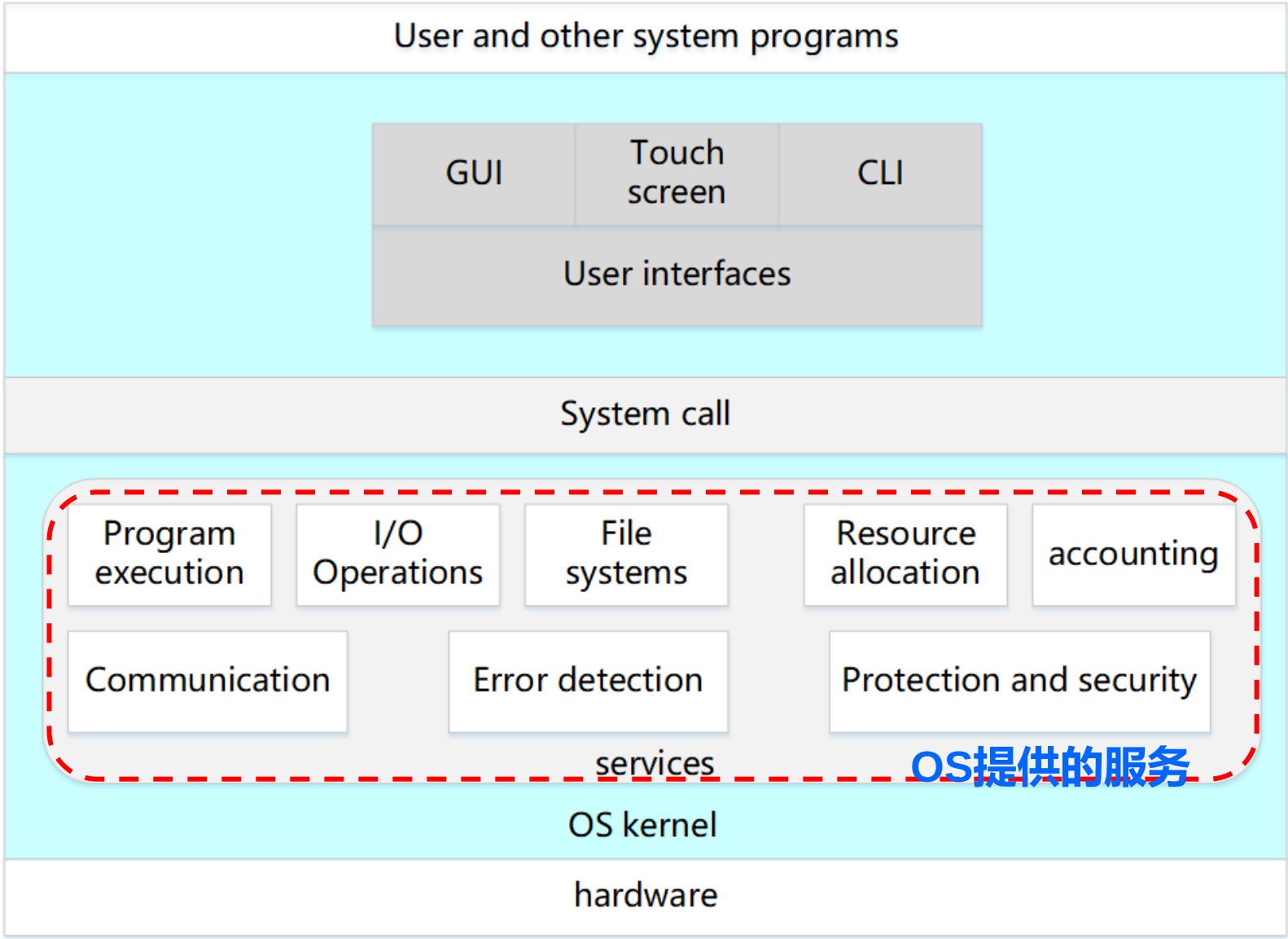


错误处理: Interrupt Handling/Exception Handling

Resource Allocation、Logging、Protection&Security。

Resource Allocation	CPU资源的分配 内存资源分配 磁盘空间分配
Logging	通过日志记录并统计，便于在系统出现问题时， 通过技术手段找到问题根源并加以修复
Protection and Security	对系统资源实施访问控制，保证系统安全

OS提供的服务：Review

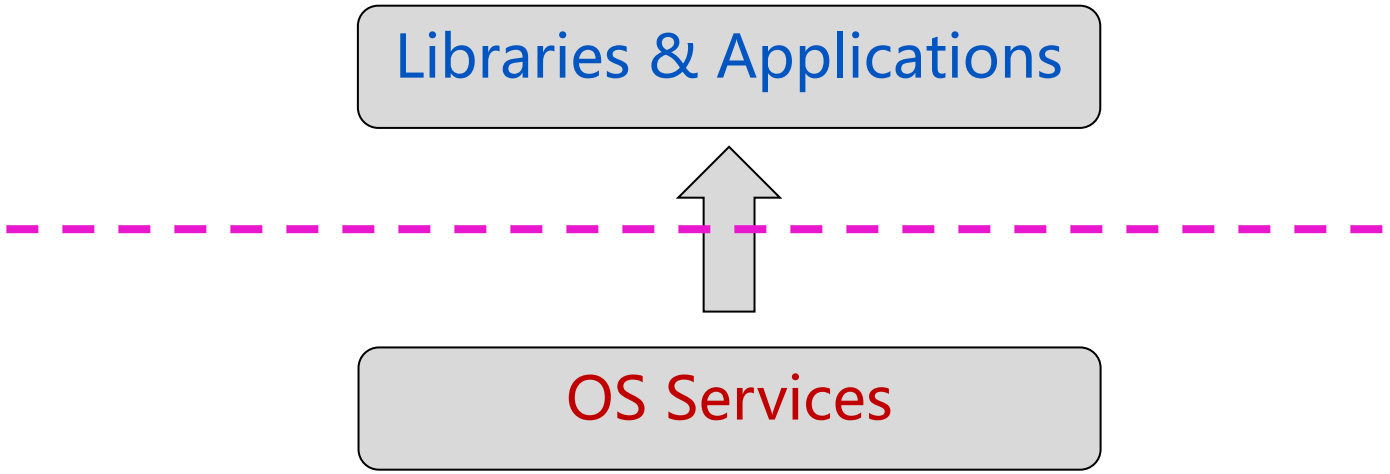


系统调用

System Call

02

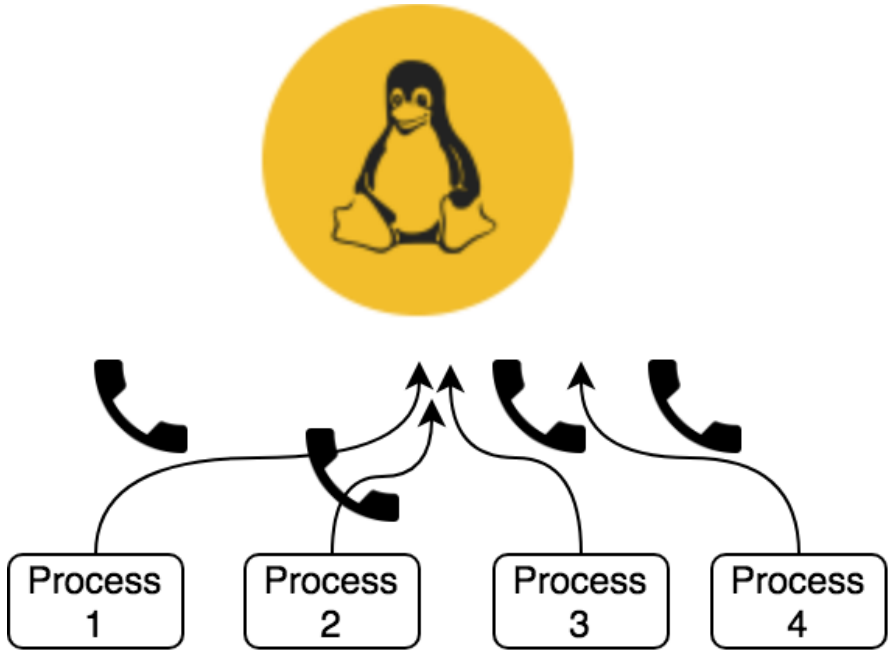
What is System Call



操作系统作为软件，也需要设计软件接口。

What is System Call

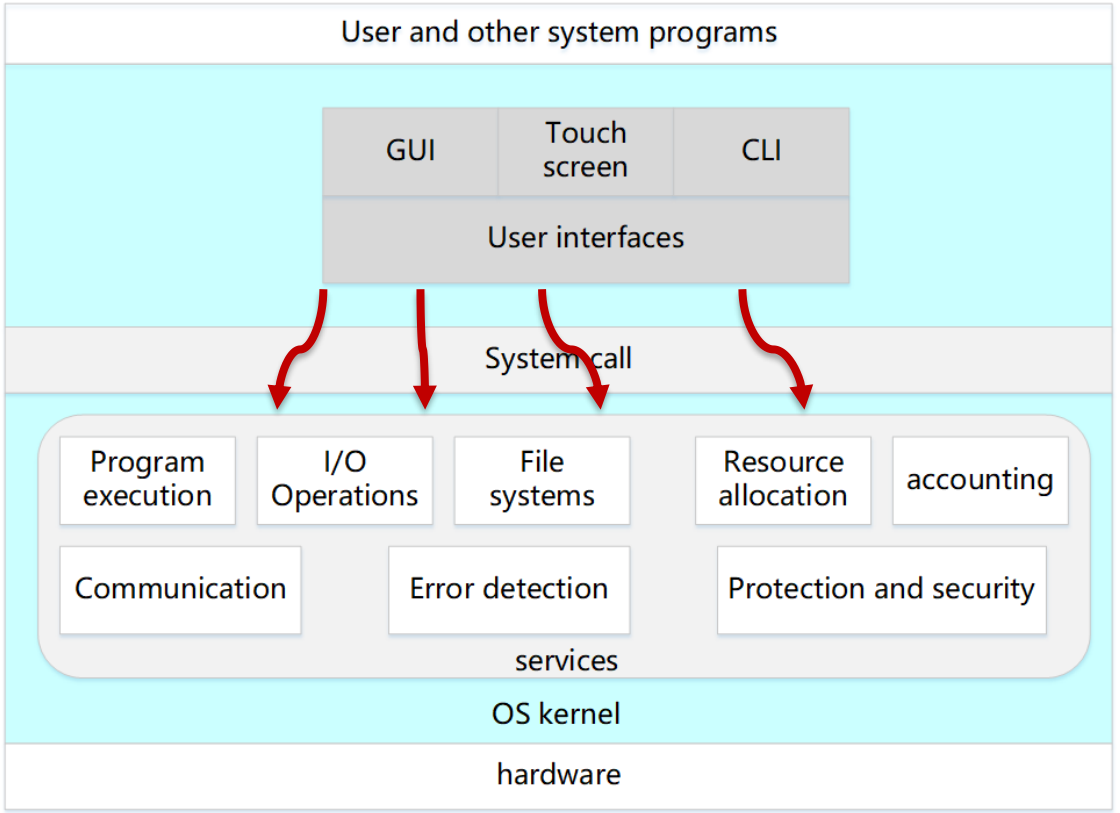
操作系统向用户态环境提供内核服务的最原始接口。



系统调用是操作系统在内核里的一些内建函数，通过系统调用接口层提供给应用层使用。

What is System Call

操作系统想要申请系统服务，必须通过系统调用层。



What is System Call

为什么需要系统调用。

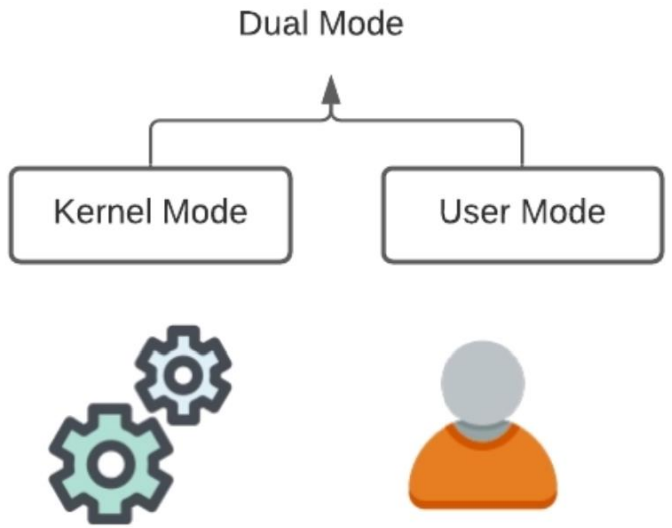
函数调用 v.s. 系统调用

OS系统的核心服务，特别是I/O操作，需要专门的保护，不可以被随意调用。

因此，操作系统的核心服务代码都会被保护在特殊的执行模式下。

应用层发起系统调用，需要经历从用户态到核心态的切换

核心态 = 受保护的特殊执行模式

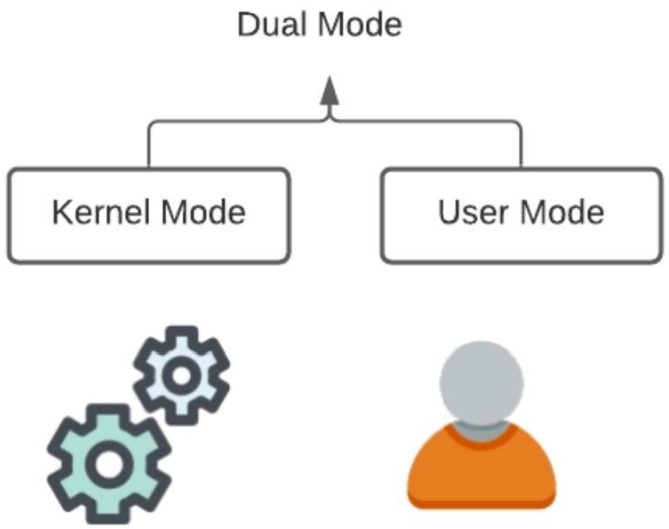


图片来源：https://www.codingninjas.com/studio/library/dual-mode-in-os

双模式

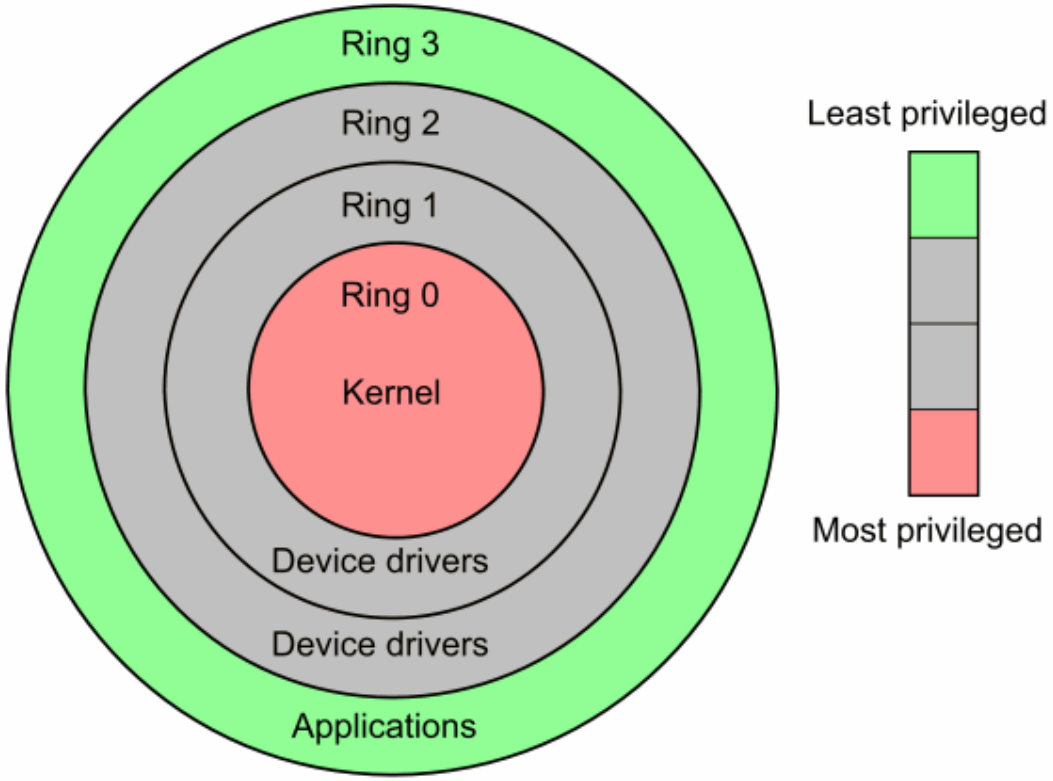
双模式。

用户模式 v.s. 核心模式



X86的4个protection ring

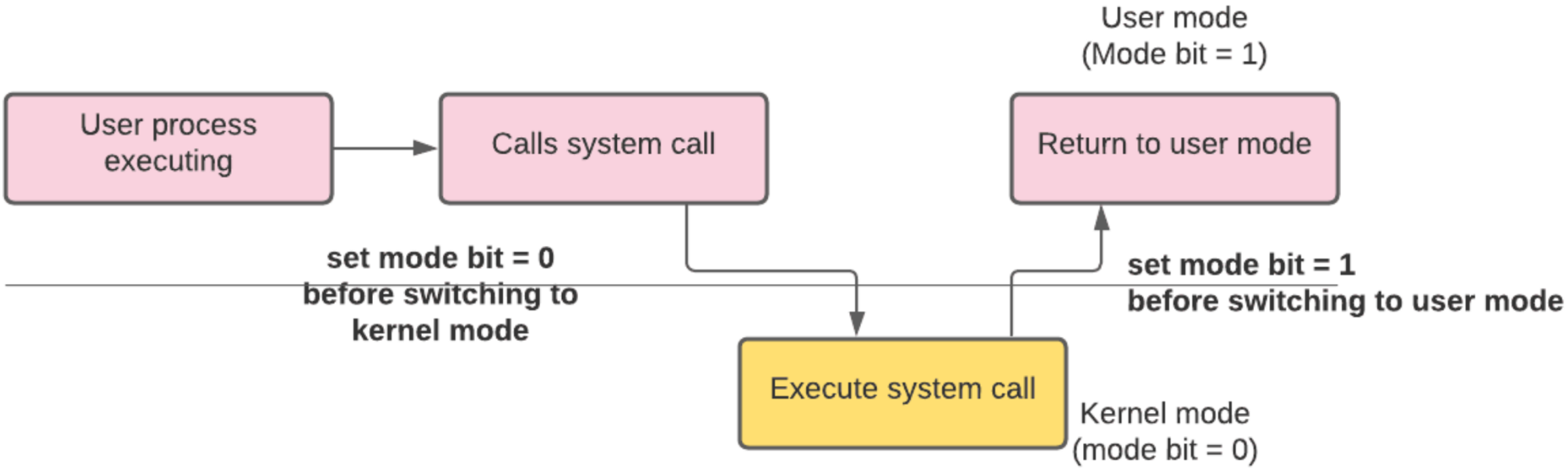
Only rings 0 (Kernel) and 3 (User) are typically used.



图片来源：https://blog.codinghorror.com/understanding-user-and-kernel-mode/

系统调用执行流程

Workflow of System Call



System Call参数传递

系统调用的参数传递：

系统调用过程中会发生执行模式从用户态到内核态的转变，
参数传递方面需要考虑这方面因素

寄存器传参

- 优先选择寄存器传参，效率高
- 问题：可用寄存器数量有限

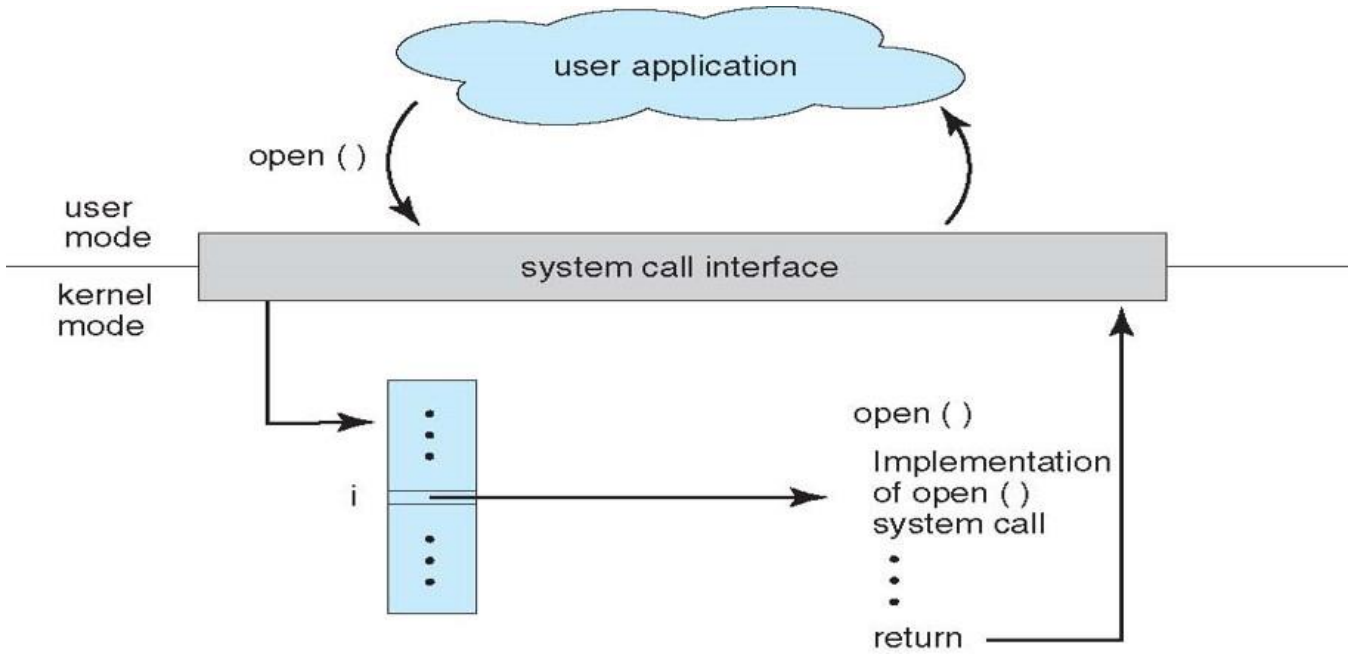
内存块传参

- 分配一块内存，存放系统调用参数，并将参数块指针作为参数传递给内核
- 当参数多于可用寄存器数量时采用

栈传参

- 用户程序中将参数压入栈
- 内核从栈中将参数出栈

系统调用核心流程：



（1）程序执行过程中执行系统调用open()

（2）CPU执行模式从用户态切换到核心态

（3）系统根据open系统调用的编号i，在系统调用表中查找，得到指向内核中系统调用open具体实现的代码入口地址

（4）转去执行系统调用open的实现代码

（5）从系统调用代码返回后，CPU执行模式从核心态切换回用户态

（6）应用继续执行后续代码

OS结构设计

OS Design Structure

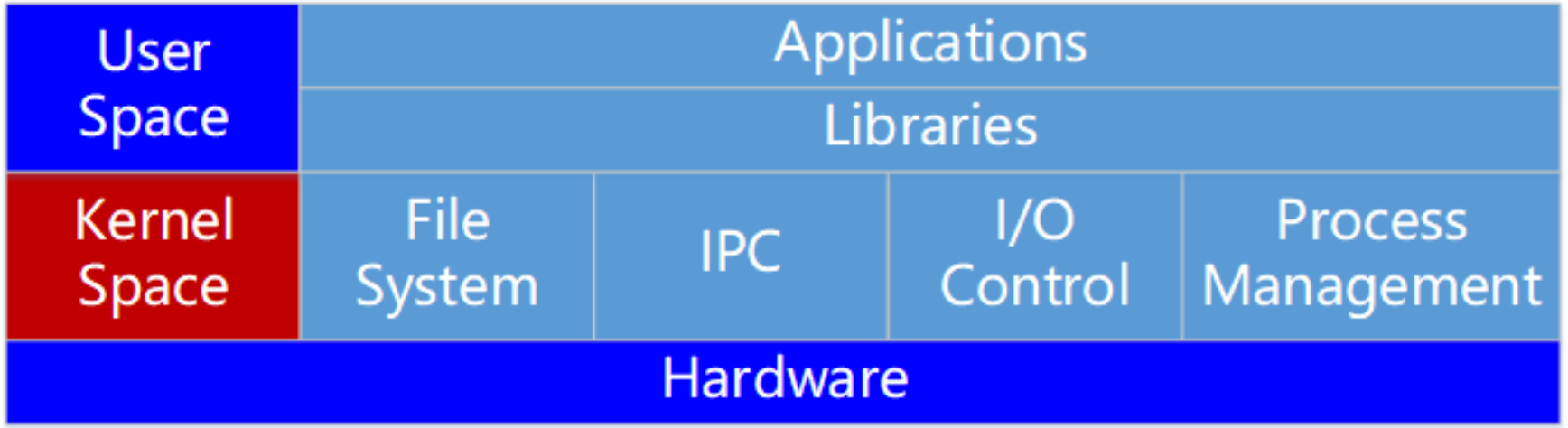
03

操作系统结构分类

- Monolithic Kernel
- 所有OS功能模块均在内核态工作

单内核设计结构

- 单内核 （宏内核）

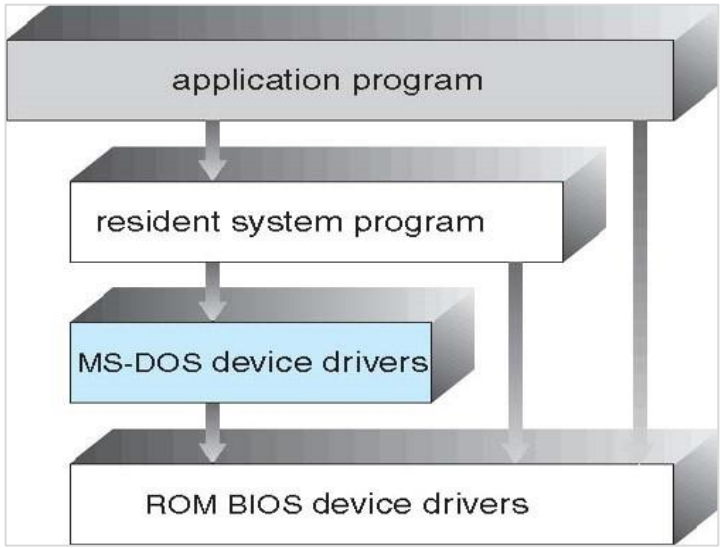


典型代表:
DOS,
UNIX

单内核设计结构

- 单内核实例1：MS-DOS

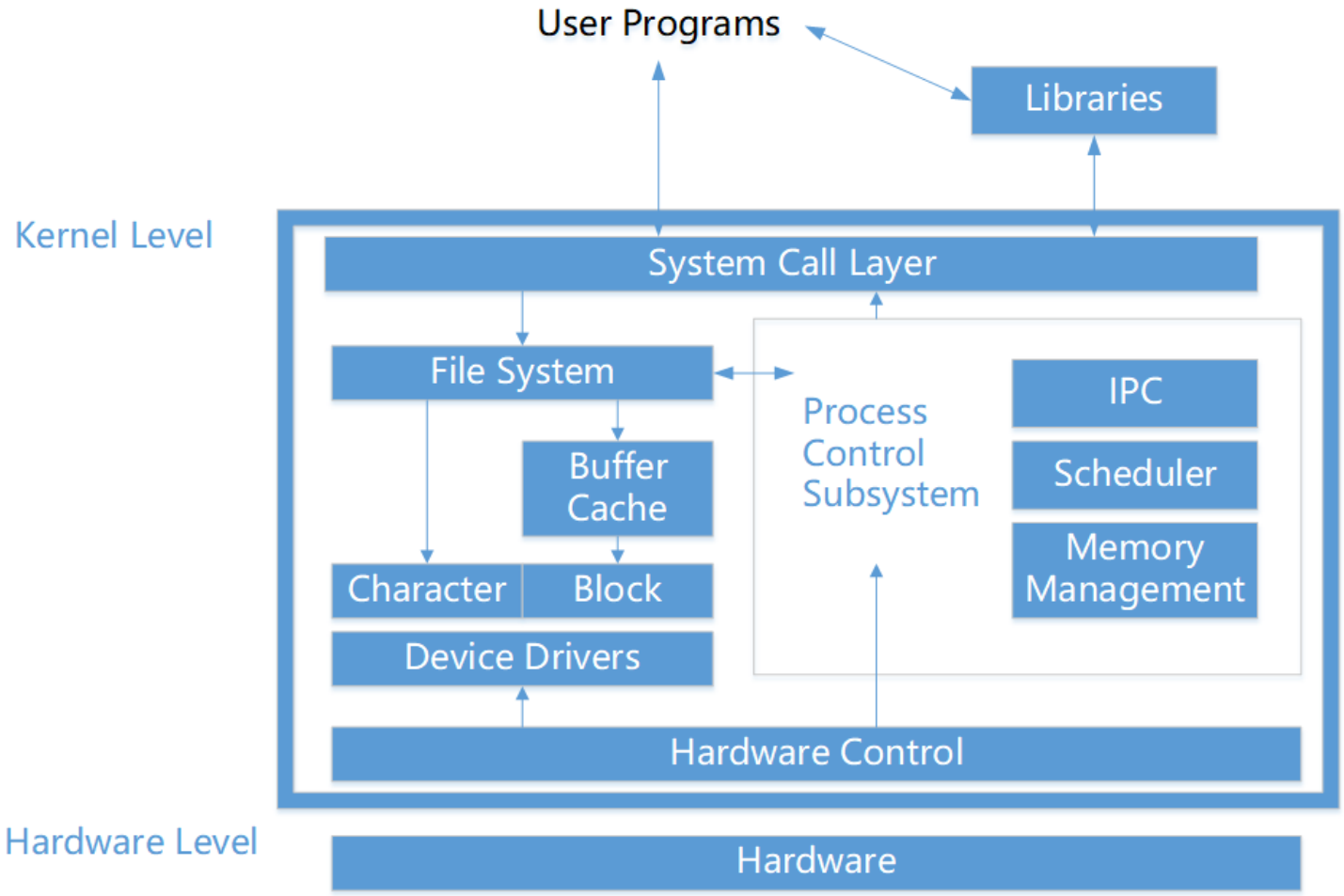
简单结构



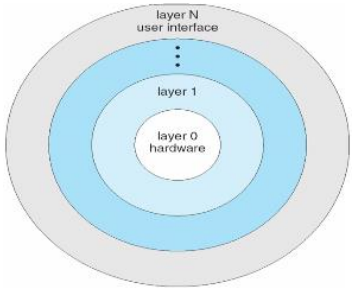
特点：
- 系统没有进行清晰的系统模块划分
- 应用程序与OS内核之间缺少隔离保护

单内核设计结构

- 单内核实例2：UNIX



层次结构



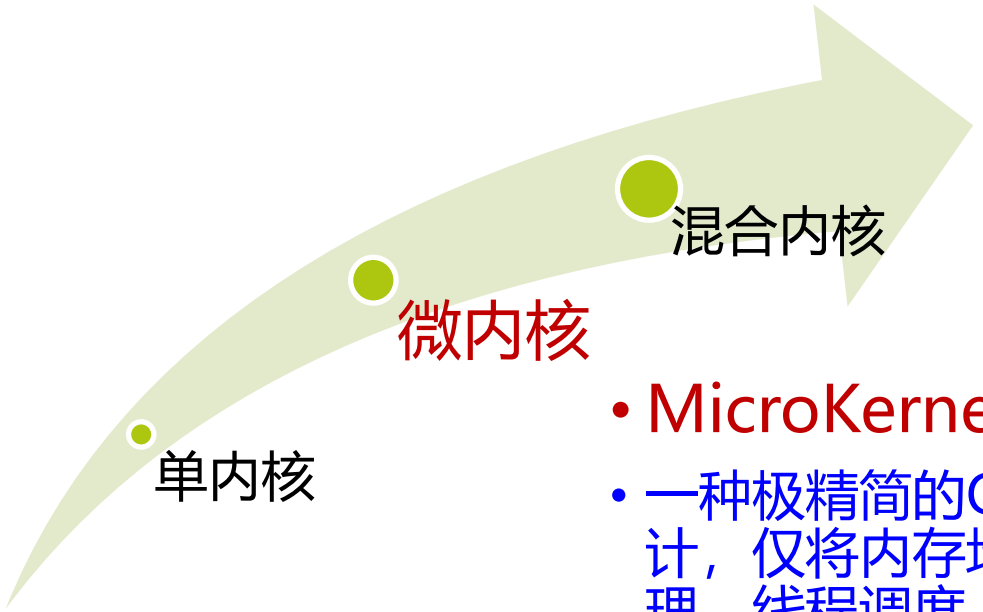
单内核设计结构

- 单内核设计的优劣

宏内核，也被称为单体内核，是一种把所有的服务都集中在一起的内核设计。它的优点是性能高，因为所有服务都在内核中运行，调用过程简单，效率高。

但是，这种设计也有缺点，如果内核中的一个服务出现问题，可能会影响到整个系统的稳定性。

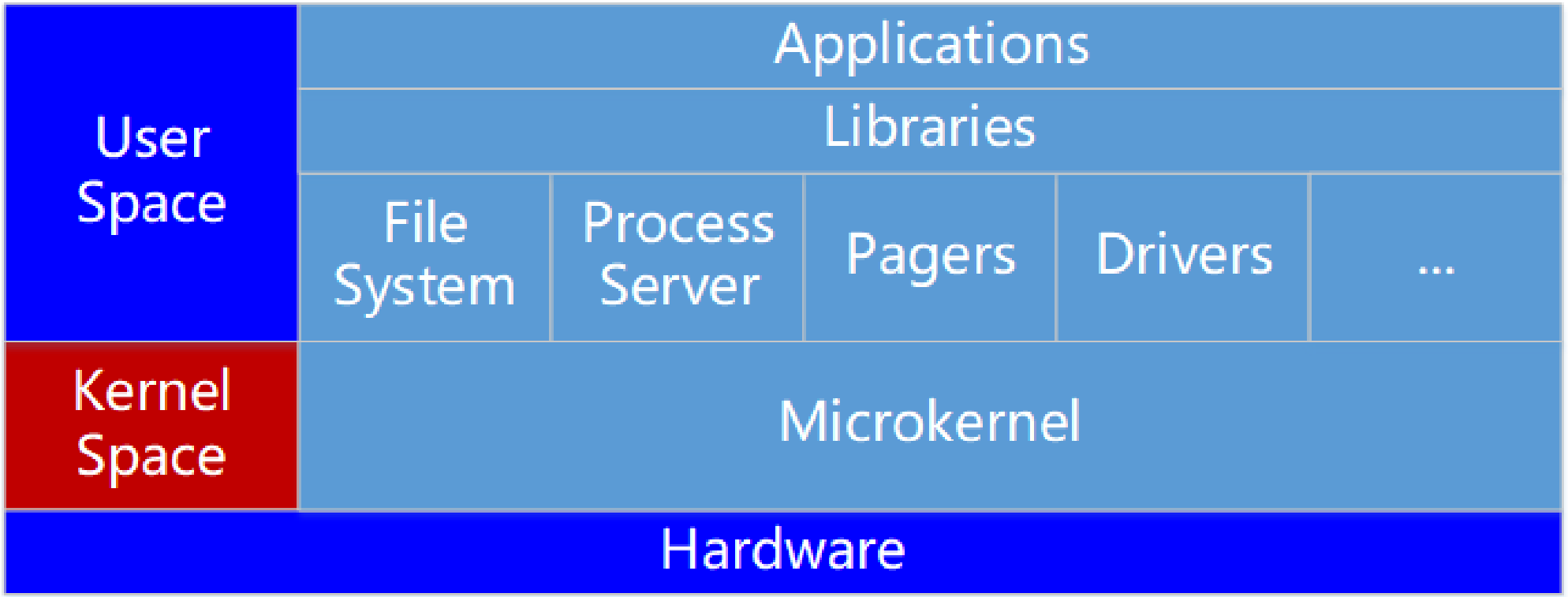
操作系统结构分类



- MicroKernel
- 一种极精简的OS内核设计，仅将内存地址空间管理、线程调度、进程间通信纳入内核，而将文件系统等模块置入用户空间

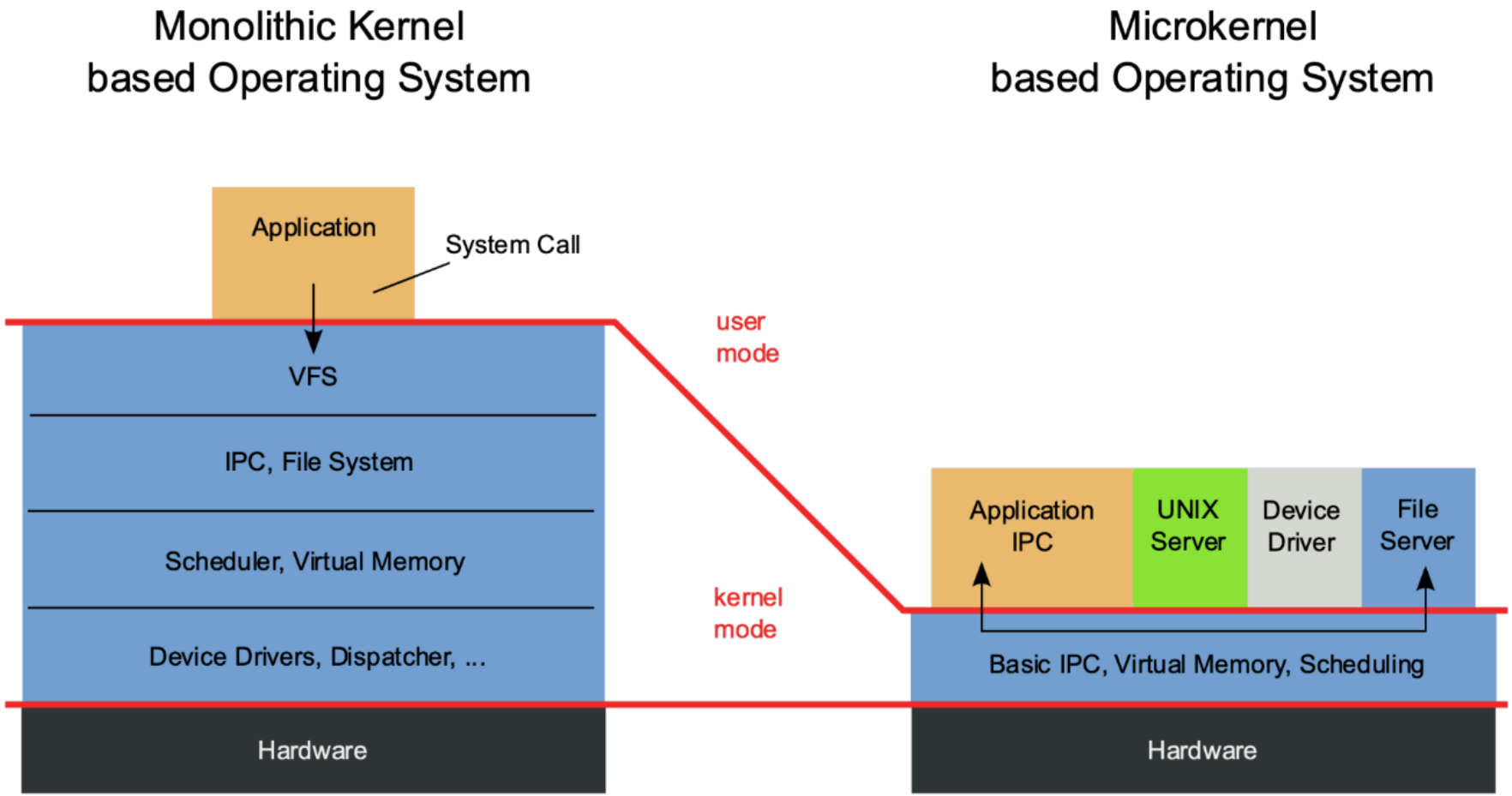
微内核设计结构

典型代表：
Mach



微内核设计结构

- 单内核与微内核对比图



微内核设计结构

• 微内核的优点

微内核，只提供最基本的服务，如进程调度、内存管理等，其他的服务，如文件系统、网络协议等，都在内核之外的用户空间中运行。这种设计的优点是结构简单，容易理解和修改，如果一个服务出现问题，也不会影响到其他服务。但是，这种设计的缺点是性能较低，因为服务之间的调用需要在内核和用户空间之间进行切换，效率较低。

系统服务模块化，可移植性高

可以多套系统服务共存，相当于同时运行多个操作系统，可扩展性强

稳定统一的接口（可以独立维护私有驱动以及服务，不需要与内核源码绑定）

在商业上，微内核可以避免代码受到一些开源协议的影响，如GPL

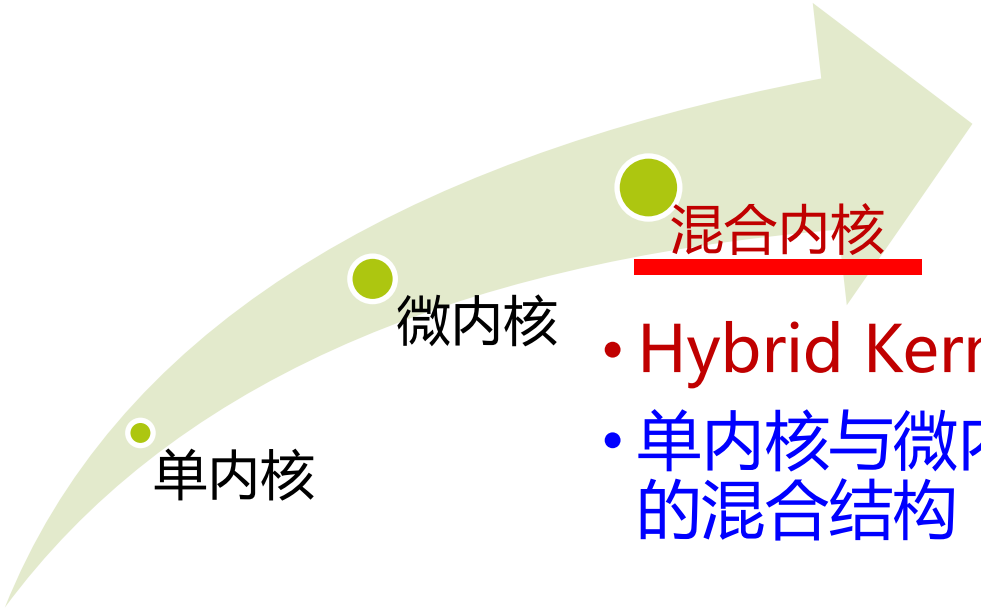
内核精简，便于进行形式化验证，利用数学证明内核的安全性、可靠性、实时性

微内核中的通信通常基于消息传递形式，从而能够很好地支持分布式系统

• 微内核的缺点

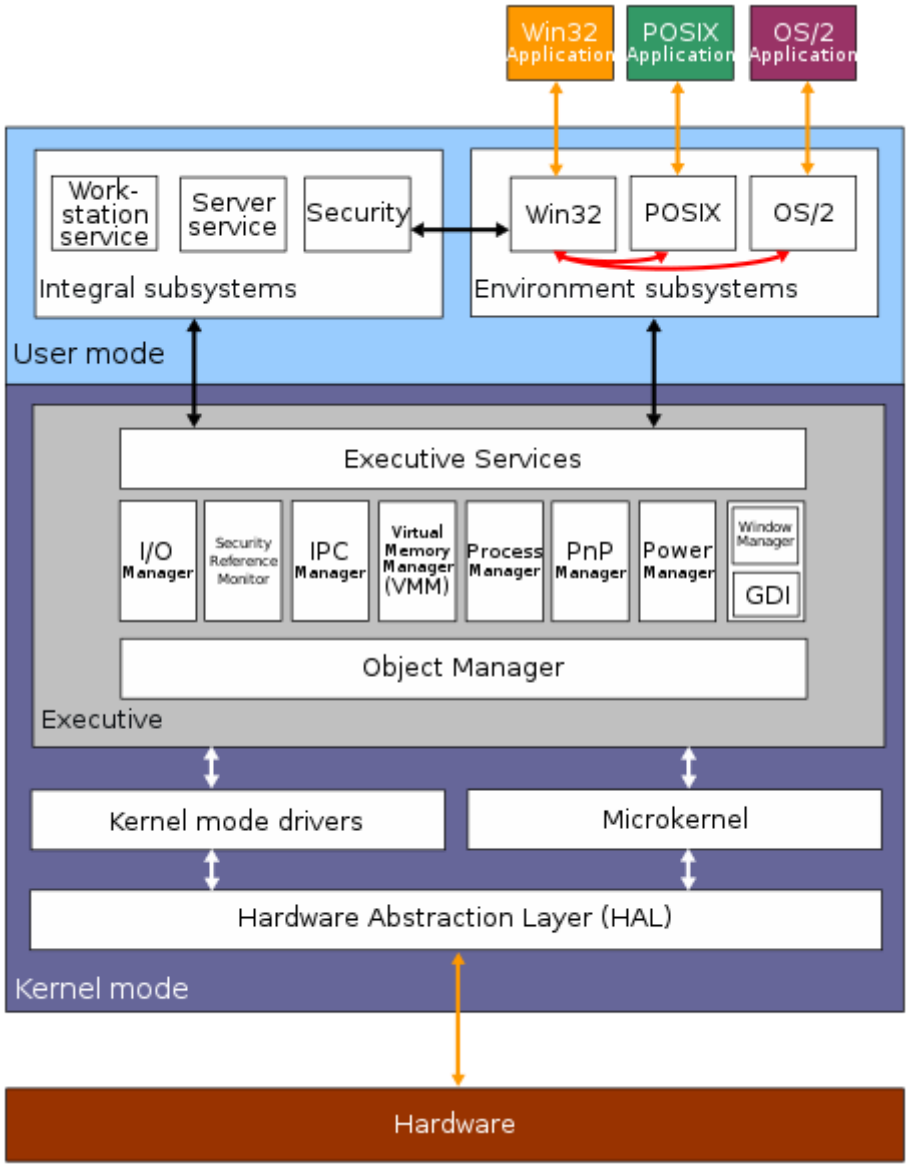
相较于宏内核，效率有所下降

操作系统结构分类



- Hybrid Kernel
- 单内核与微内核的混合结构

混合内核设计结构



- 混合内核示例1：Windows 2000

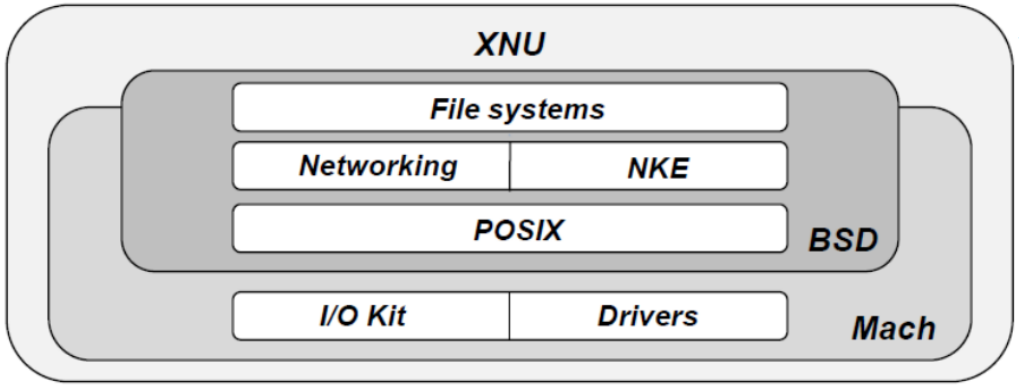
Windows 2000的NT Kernel：
混合内核机制

NT内核组成：
设计上受Mach影响的微内核
包含主要进程管理、内存管理等核心模块的Executive

Emulation Subsystem：放在用户态

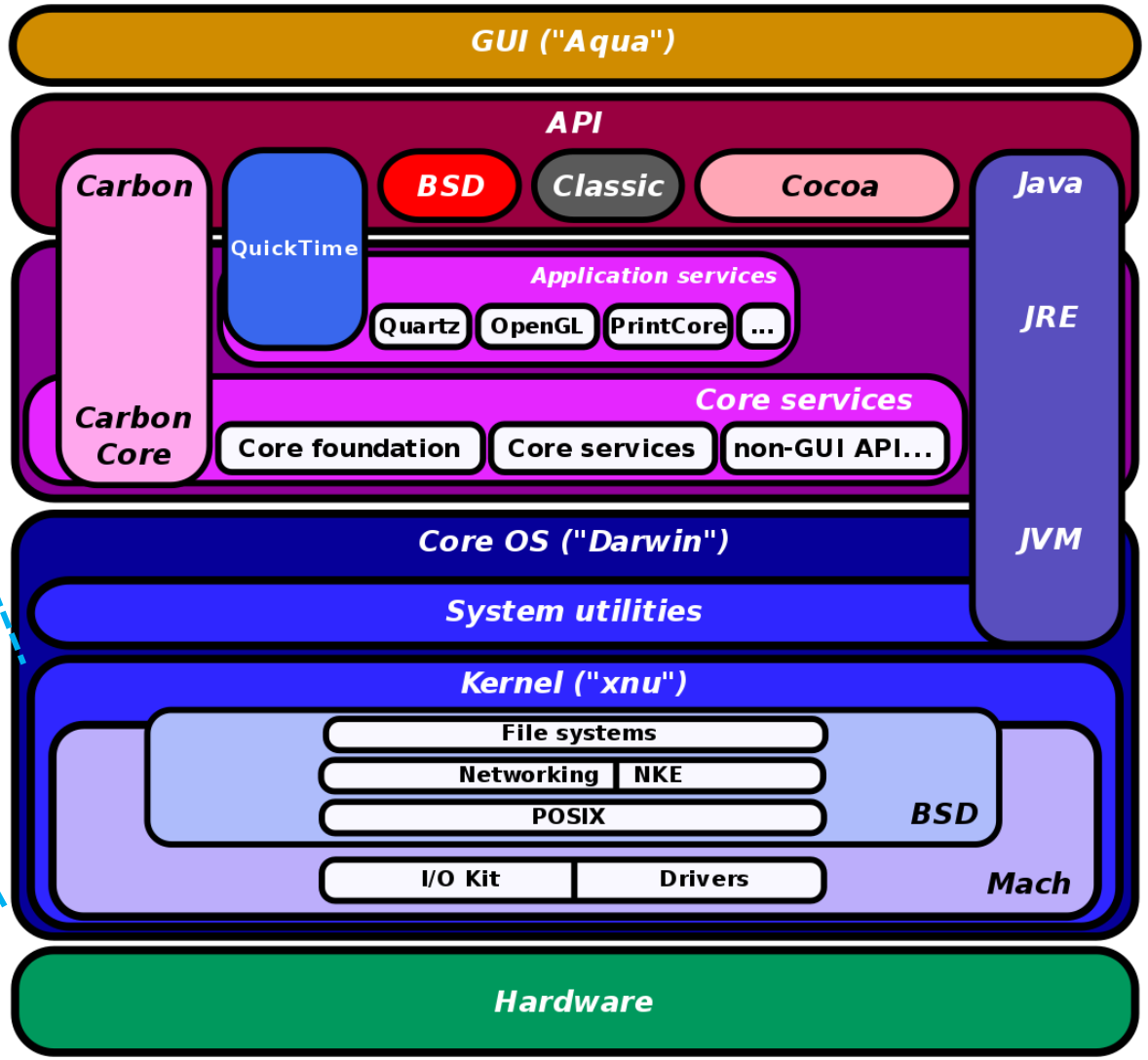
混合内核设计结构

- 混合内核代表2：XNU



Mach微内核：
CPU调度、虚存机制、I/O Kit、设备驱动

BSD内核部分：
实现文件系统、网络层、POSIX接口层



MacOSX Architecture

图片来源：https://en.wikipedia.org/wiki/Darwin_%28operating_system%29

混合内核设计结构

- 混合内核设计的优势

混合内核，基于微内核的架构设计，把一些性能要求高的服务放在内核中，比如设备驱动、应用进程间通信等，而其他的服务则放在用户空间中。

这种设计既有宏内核的性能优势，又有微内核的稳定性优势。

但是，这种设计的缺点是（比纯粹的微内核设计）复杂性高，需要仔细地选择哪些服务放在内核中，哪些服务放在用户空间中。

Summary

小结：

- 操作系统服务
- 系统调用
- OS结构设计

下列选项中，通过系统调用完成的操作是（ ）。

A. 页置换

B. 进程调度

C. 创建新进程

D. 生成随机整数

简述微内核的主要优点。

答：

（1）提升了系统的可扩展性
（2）增强了系统的可靠性
（3）提升了系统的可移植性
（4）提供对分布式系统的良好支持
（5）融入了面向对象技术

What is System Call

你以为的Helloworld:

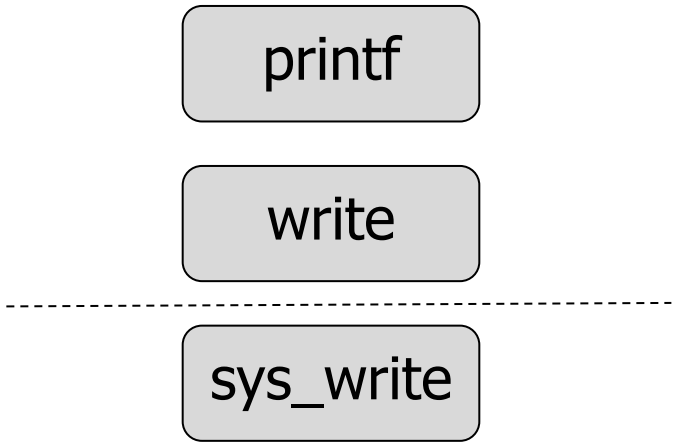
printf(“hello,world!”);

Only a single line of code?

What is System Call

实际的Helloworld：

1.程序loader为helloworld程序准备运行时环境
2.执行printf，跳转到标准C库代码
3.解析格式化字符串
4.解析后的字符串被写到标准输出
5.通过驱动程序将字符串发送给TTY设备



What is System Call

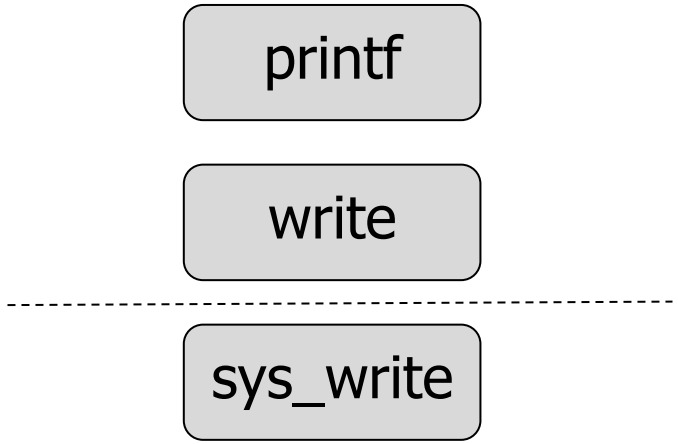
系统调用设计理念：

策略（Policy）

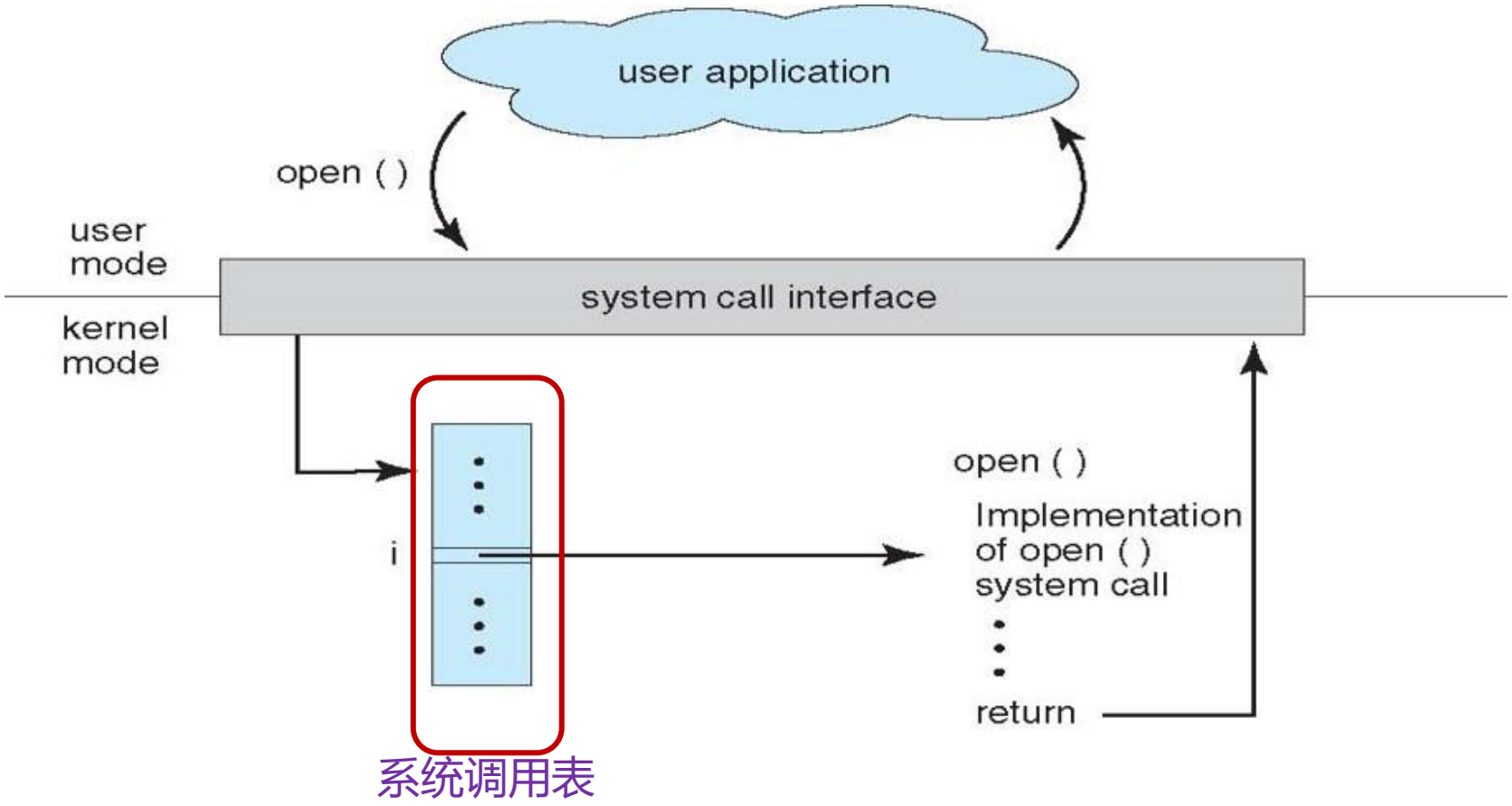
机制（Mechanism）

实际的Helloworld：

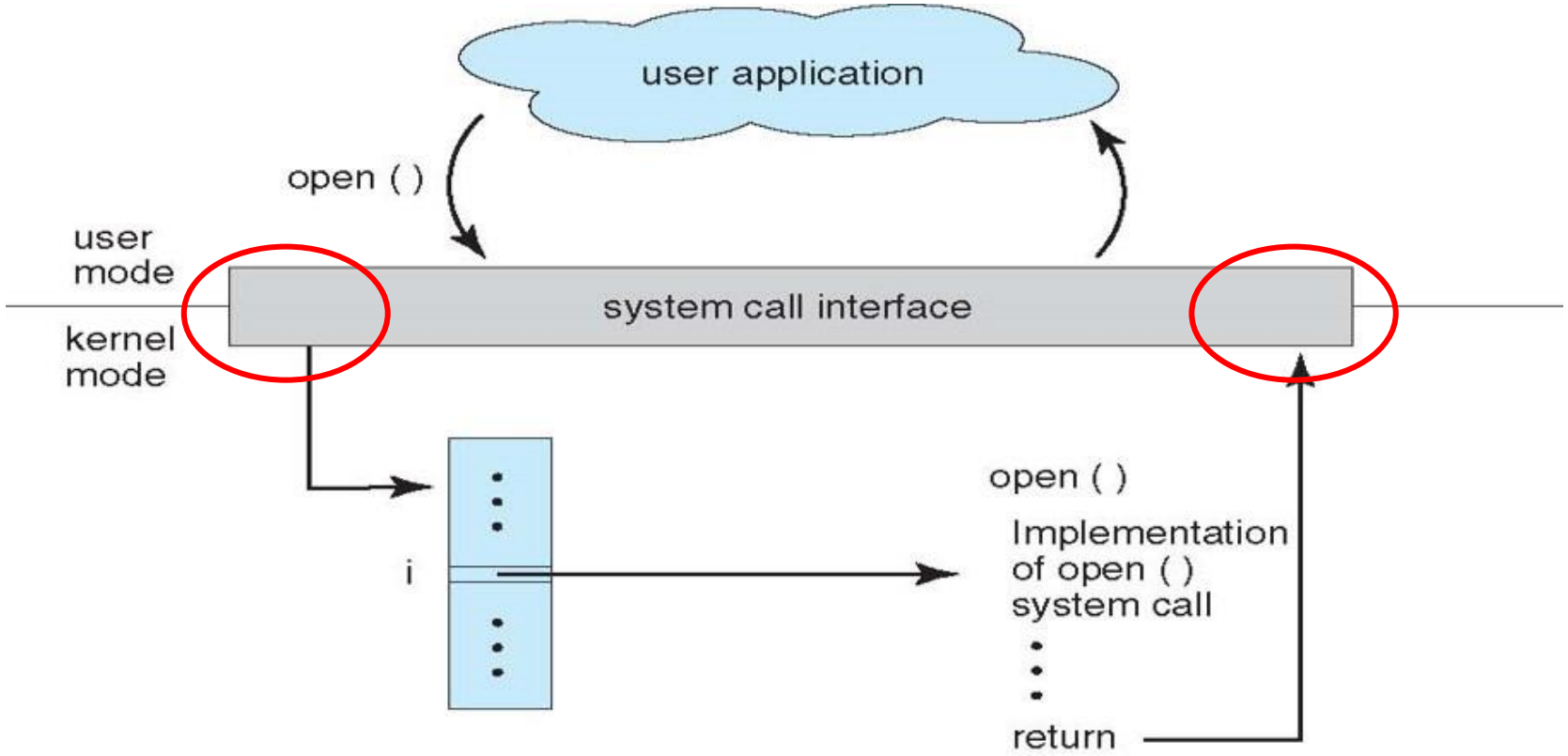
1.程序loader为helloworld程序准备运行时环境
2.执行printf，跳转到标准C库代码
3.解析格式化字符串
4.解析后的字符串被写到标准输出
5.通过驱动程序将字符串发送给TTY设备



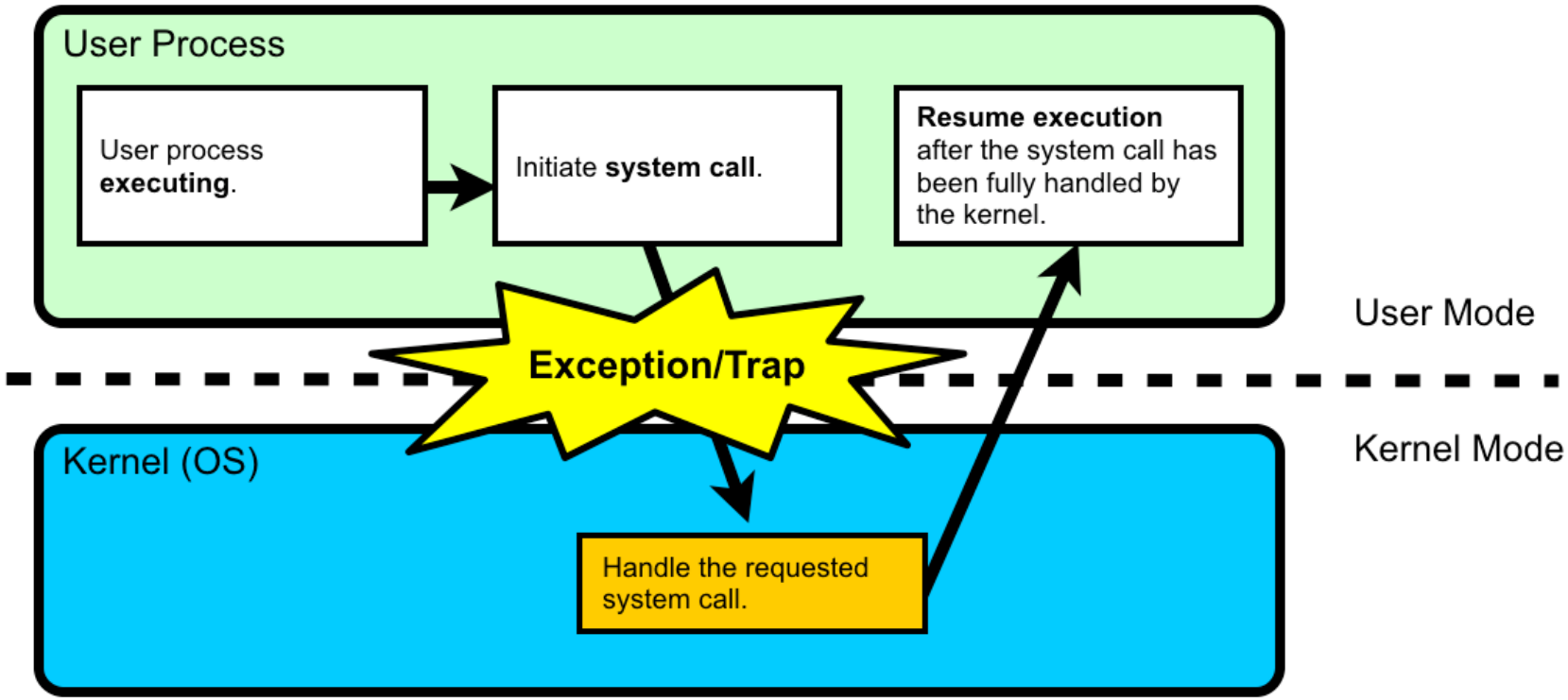
系统调用核心流程：


user application
open ( )
user mode
kernel mode
system call interface
i
open ( )
Implementation of open ( ) system call
return
系统调用表

系统调用实现要点：模式切换，参数传递（返回值处理）

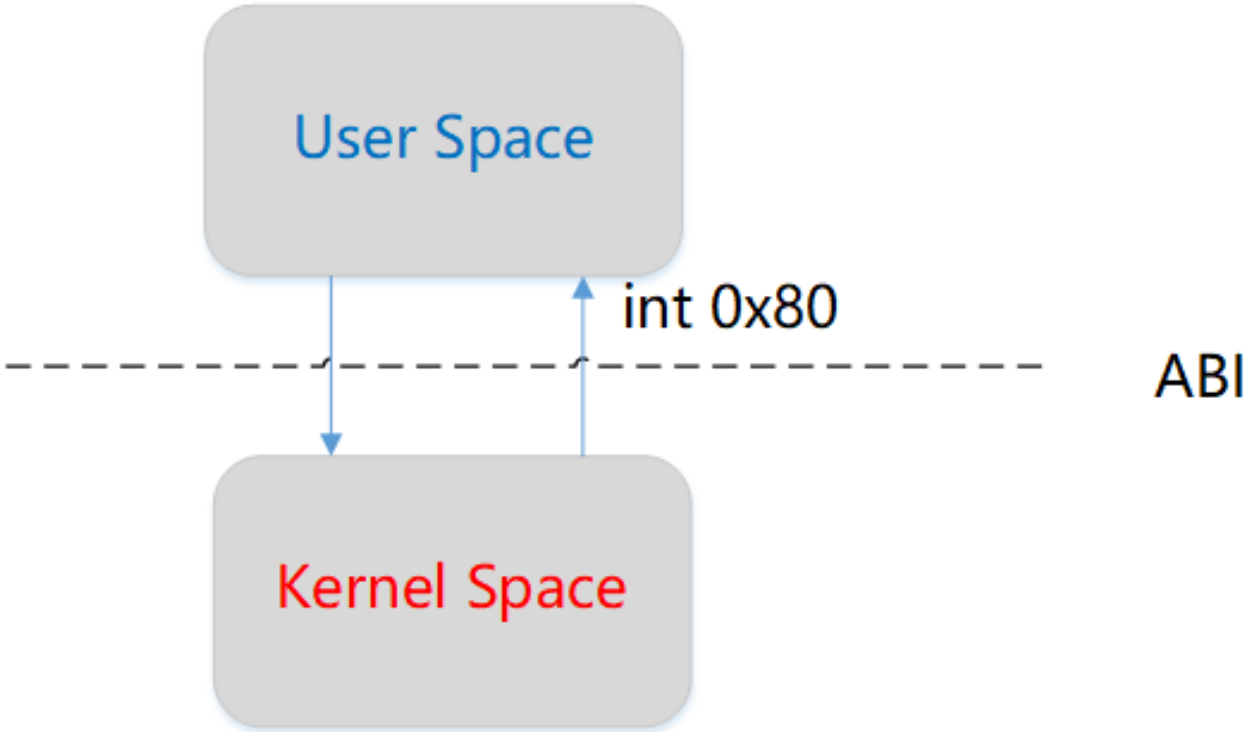


模式切换： User Mode ⇔ Kernel Mode

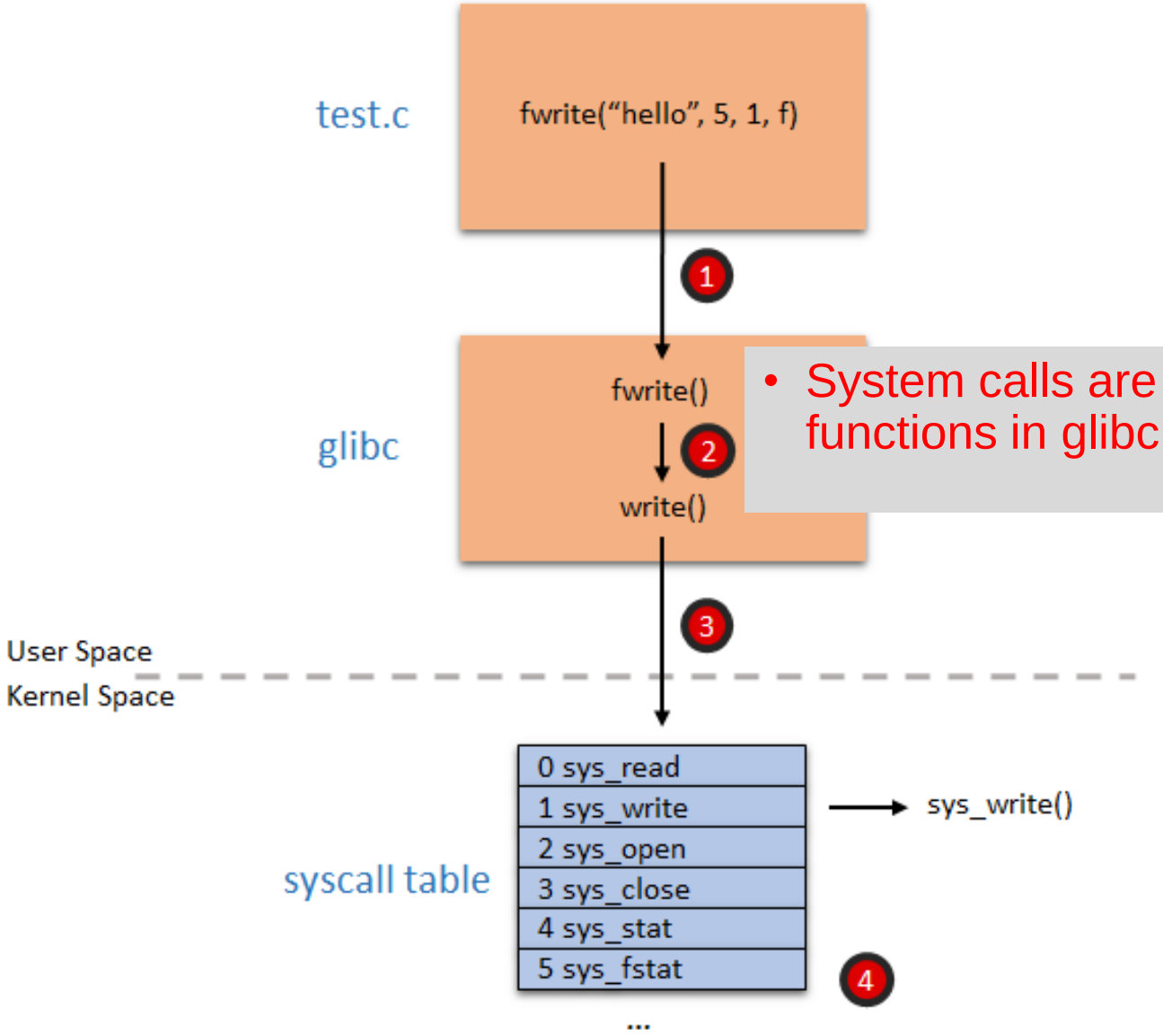


Linux系统调用

- **The system call is the fundamental interface between an application and the Linux kernel.**



Linux系统调用



- System calls are generally invoked via wrapper functions in glibc

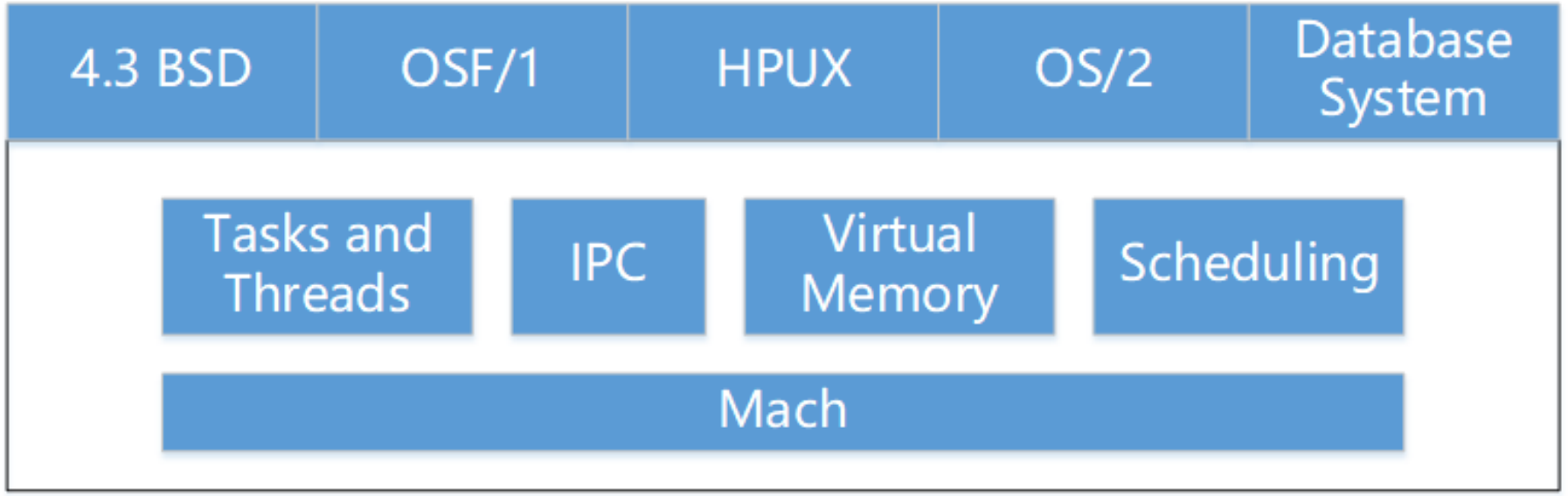
常见OS发行版

多了解些操作系统发行版实例.

UNIX, FreeBSD, Minix
Fedora, CentOS, OpenEuler, Debian, SuSE, ElementaryOS, Kali Linux, …
Windows 95,98, Me, 2000, Win7, Win8, Win10

微内核设计结构

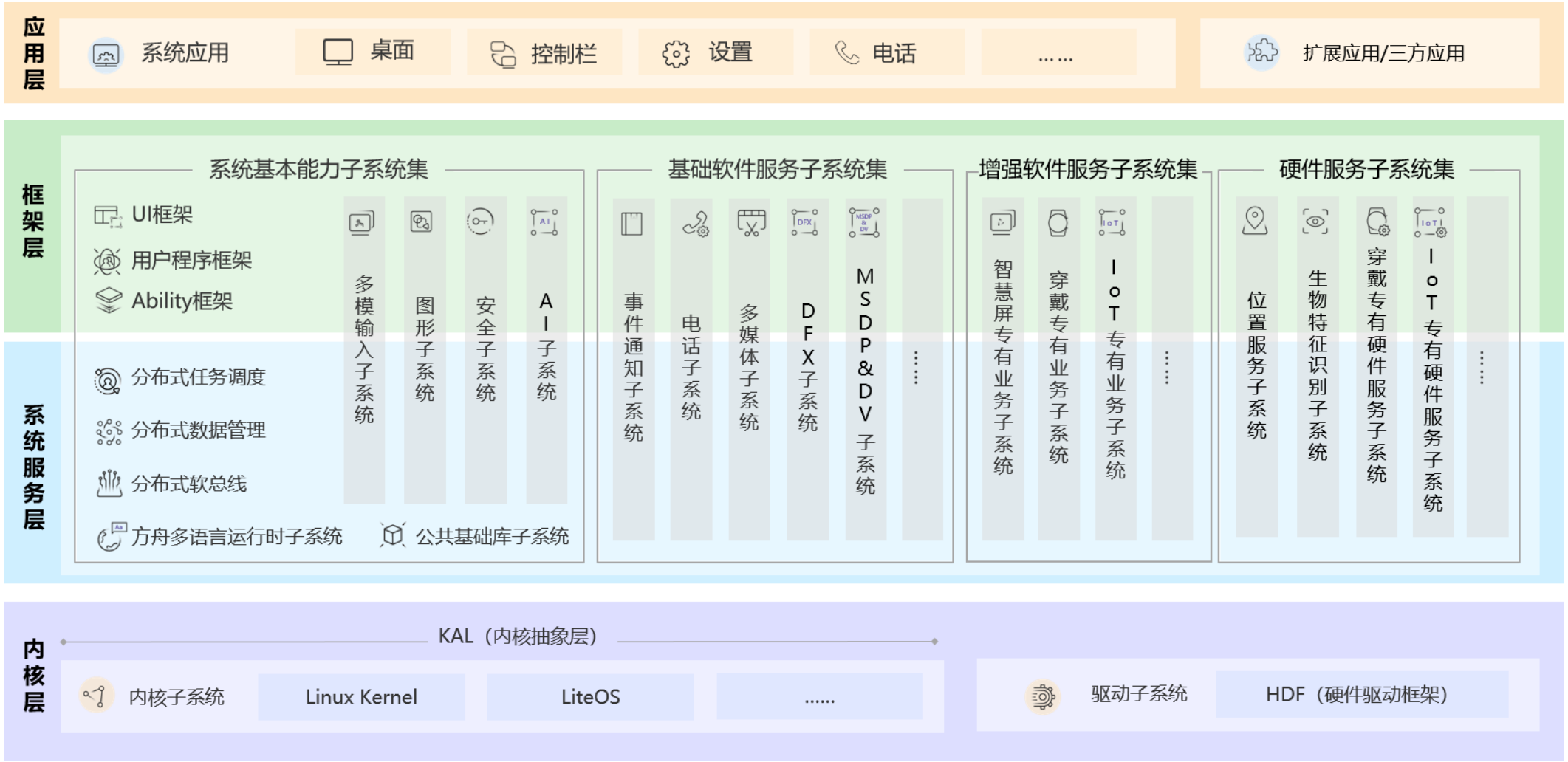
- 实例：Mach



- 只在内核中保留如线程调度、进程间通信等核心功能，而将其他OS服务转移到用户态，以用户态模块的形式实现

国产操作操作系统发展

HarmonyOS



国产操作操作系统发展

OpenEuler

作为最为流行的服务器开源Linux发行版，CentOS占据的市场份额超过70%.

服务器操作系统是企业IT系统的基础架构平台，存储着核心数据，是运行业务的命脉。

企业不能将命运寄托在一个无人维护、不稳定的系统上

国产操作操作系统发展

OpenEuler诞生背景

CentOS

Ali：龙蜥

华为：OpenEuler

国产操作操作系统发展

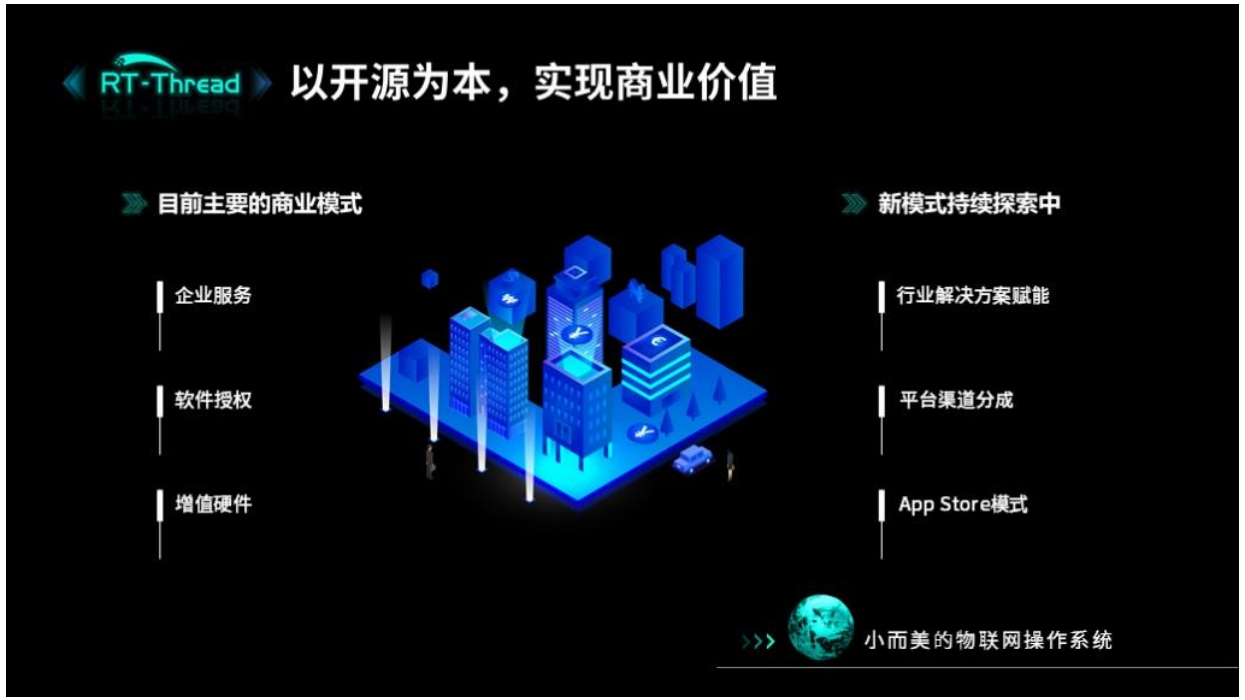
服务端操作系统-发行版

openEuler 是由开放原子开源基金会（OpenAtom Foundation）孵化及运营的开源项目

开放原子开源基金会
OPENATOM FOUNDATION

微内核示例

RTThread Smart

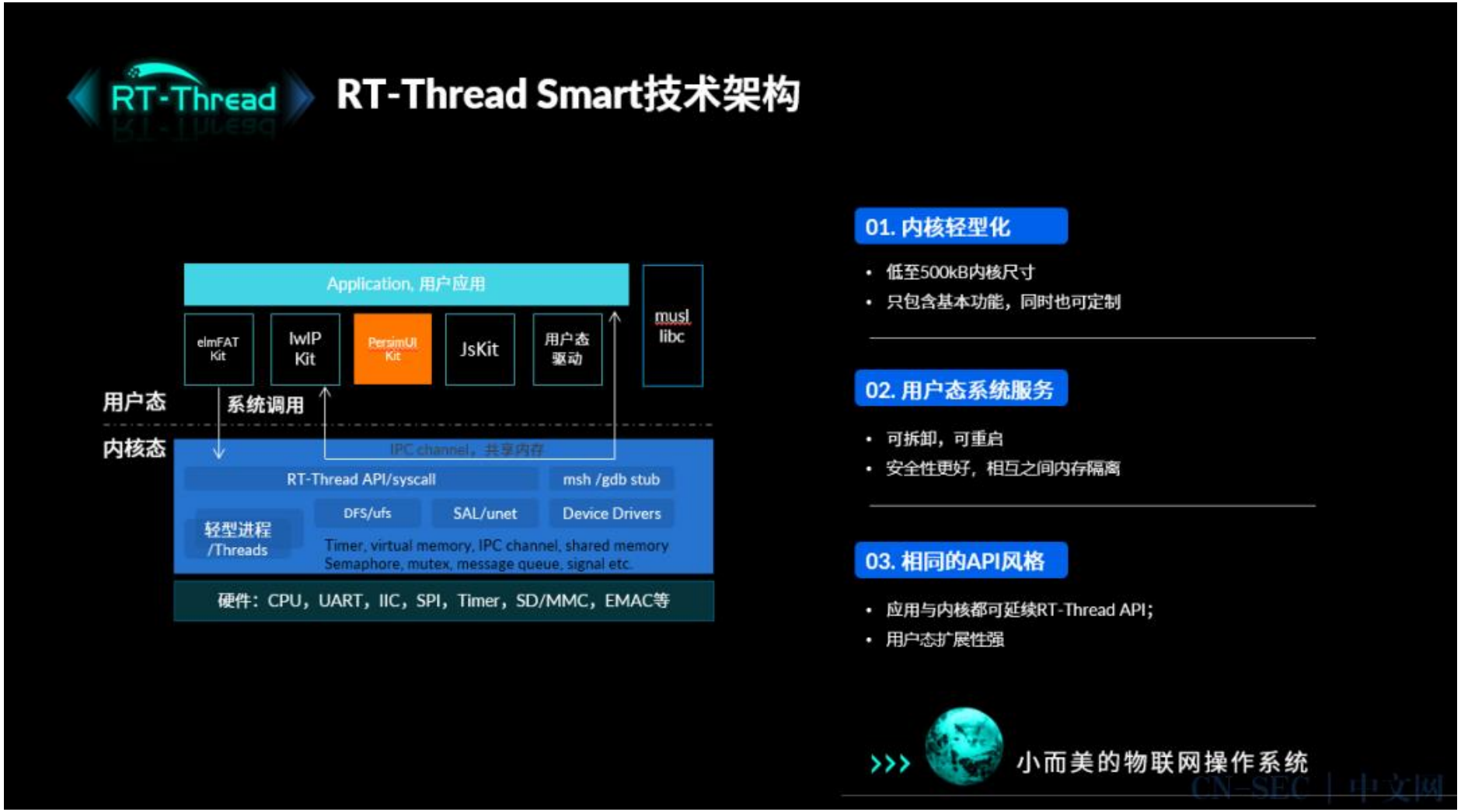


RT-Thread：应用广泛的国产嵌入式实时操作系统

RT-Thread具备一个IoT OS平台所需的所有关键组件，例如GUI、网络协议栈、安全传输、低功耗组件等等。经过12年的累积发展，RT-Thread已经拥有一个国内最大的嵌入式开源社区，同时被广泛应用于能源、车载、医疗、消费电子等多个行业，累积装机量超过 **2亿** 台，是目前国内最火的物联网操作系统。

微内核示例

RTThread Smart



讨论：UI服务在用户态or内核态

Windows在内核态实现核心的GUI服务

Linux在用户态实现GUI服务

谢 谢!
Thank you!