极客星球分享-5期

Alex|荣哥

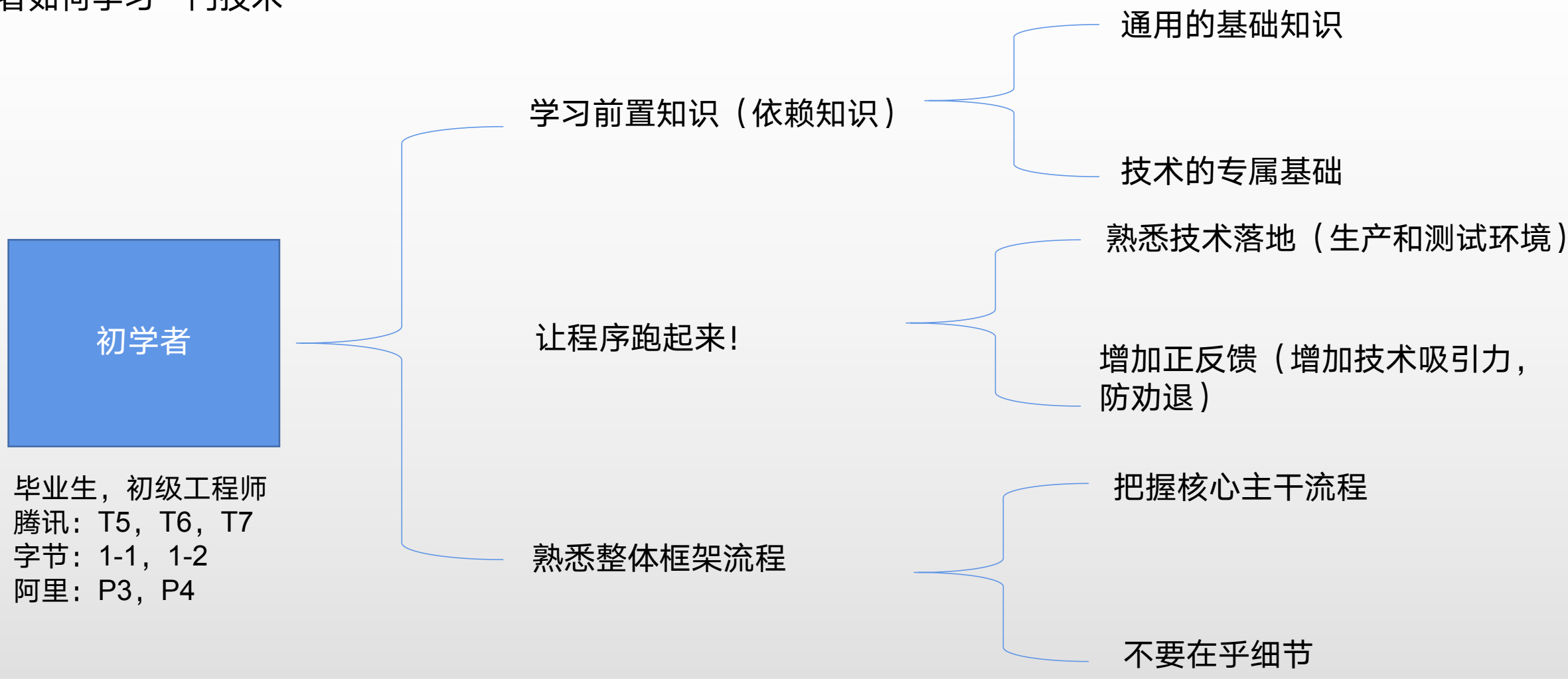
目录

- 如何掌握和精通一门技术
  - 如何学习一门技术：**入门，掌握，精通**
  - 面试中1-4轮技术面要求
  - 不同职级面试的技术要求

- 如何学习和攻破操作系统
  - OS理论原理
  - 深入理解Linux内核总体架构
  - 掌握各个子系统核心知识
  - 常见问题解决经验分享（内核调试，crash问题，性能问题等）

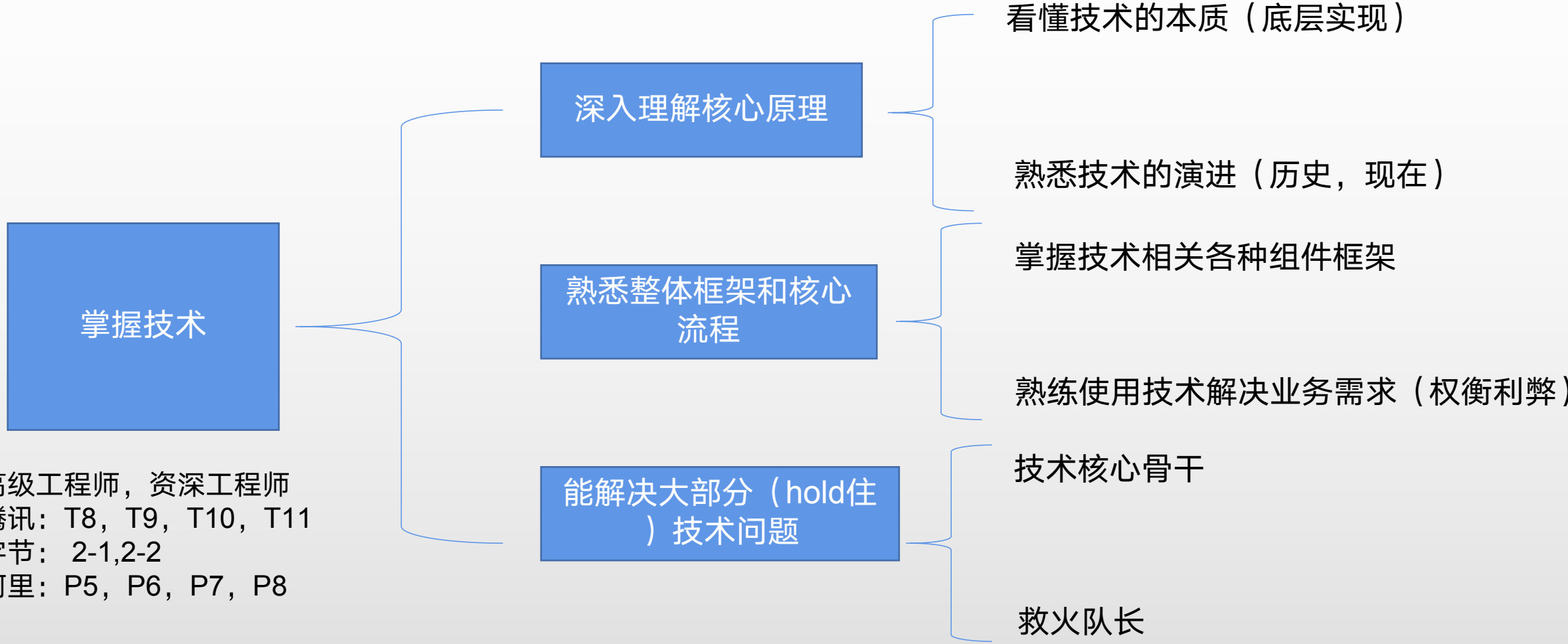
如何掌握和精通一门技术

初学者如何学习一门技术



如何掌握和精通一门技术

如何掌握一门技术



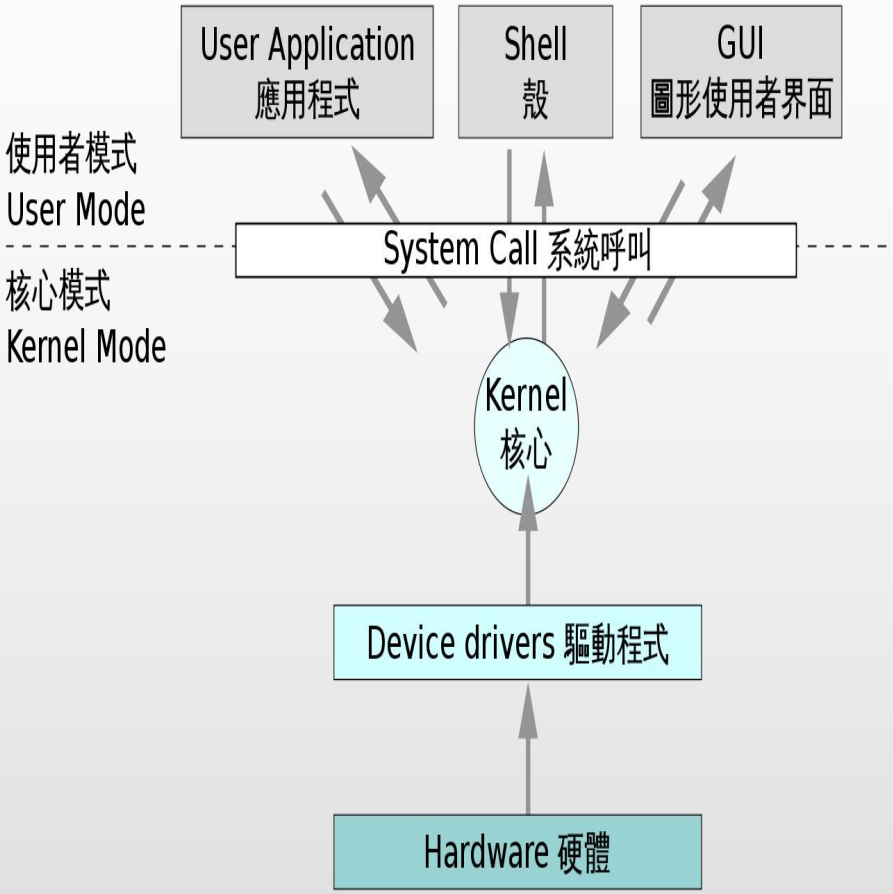
高级工程师，资深工程师
腾讯：T8，T9，T10，T11
字节： 2-1,2-2
阿里：P5，P6，P7，P8

如何掌握和精通一门技术

如何精通一门技术

如何学习和攻破操作系统

OS理论原理



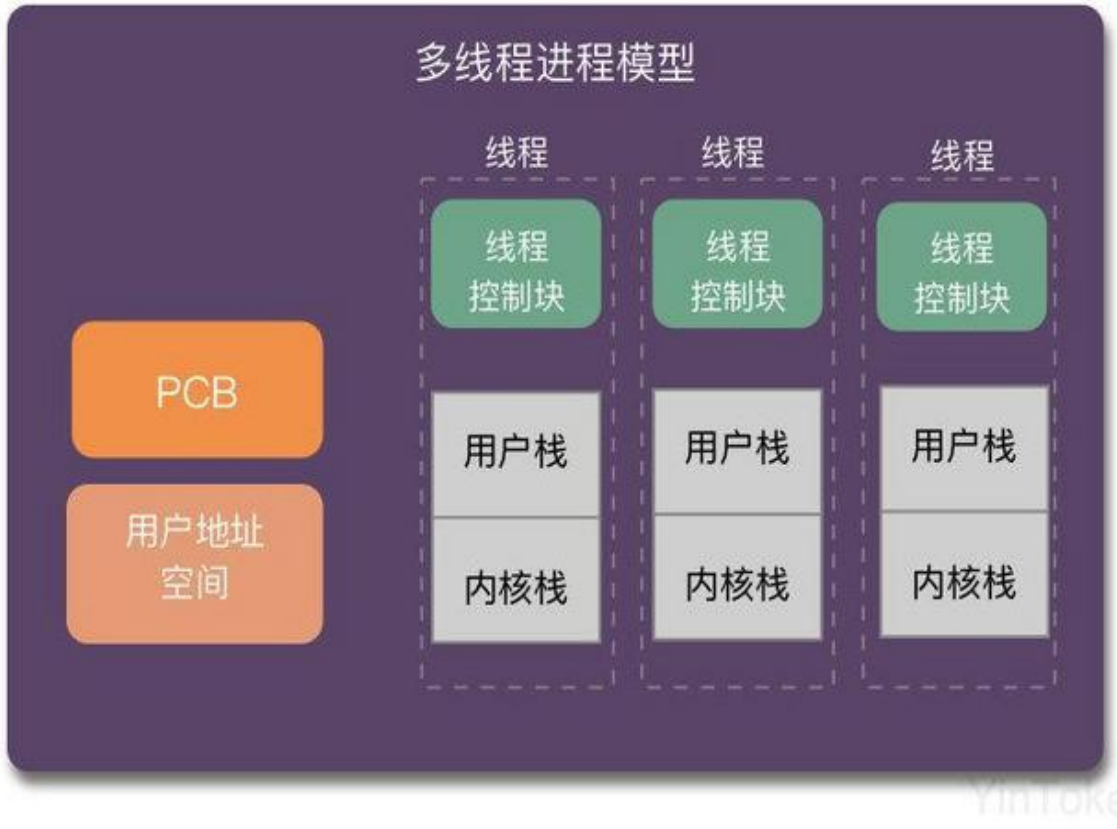
操作系统

- 用户界面
  - GUI
  - Shell
  - 编程接口
- 进程管理
  - 进程和线程原理
  - CPU调度原理
  - 进程状态机
  - 进程间通信
- 内存管理
  - 物理内存管理
  - 虚拟内存管理
- 设备管理
  - I/O设备
  - 硬件
  - 驱动程序
  - 总线
- 存储管理
  - 文件系统
  - 磁盘管理

如何学习和攻破操作系统

OS理论原理

进程管理



深刻理解进程和线程

- 计算资源
- 存储资源
- 进程
- 线程

并发同步

- 临界区资源
- 锁

进程通信

- 管道
- 消息队列
- 信号量
- 共享内存
- 套接字

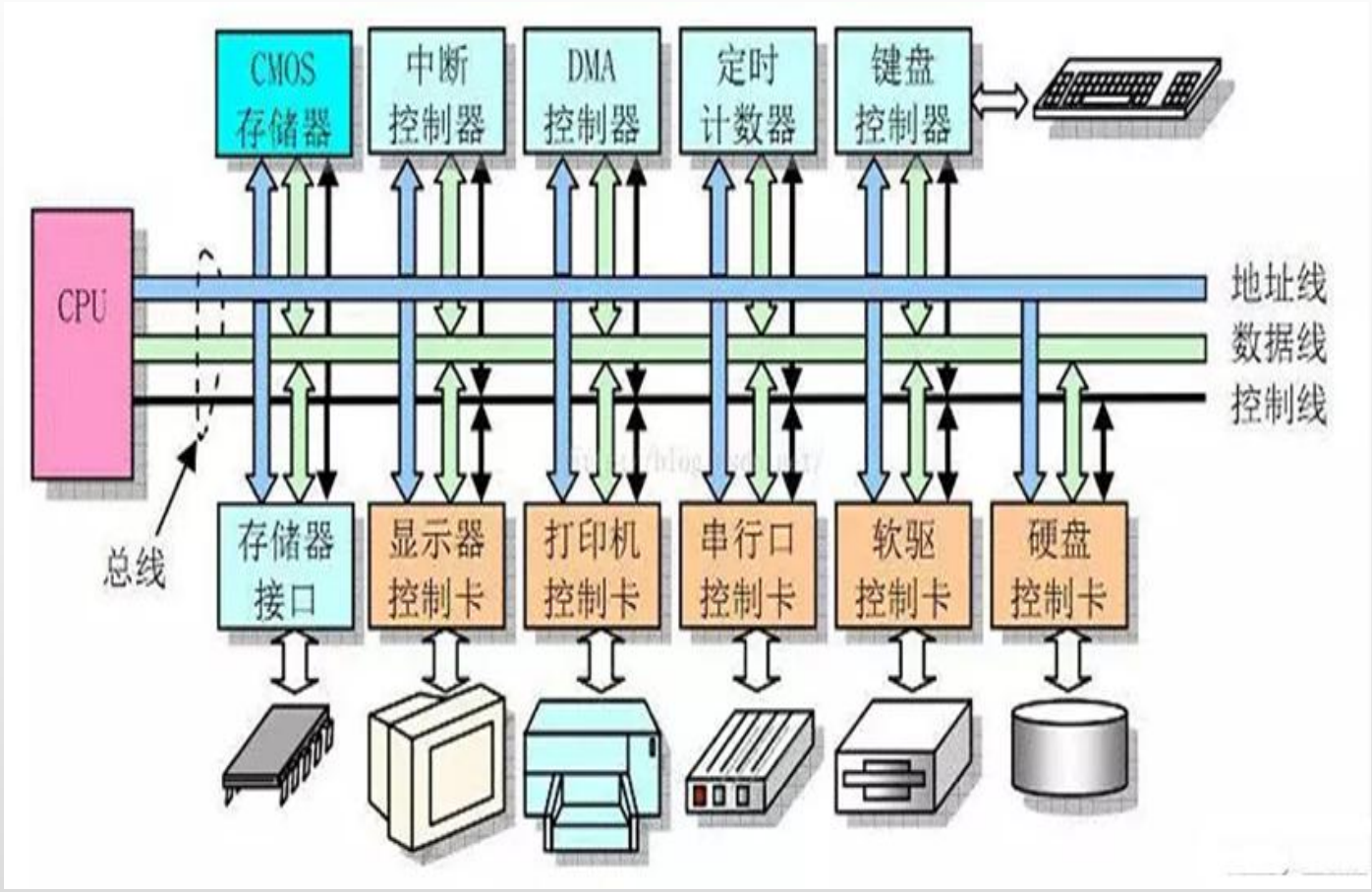
进程调度算法

- 上下文切换
- 调度算法

如何学习和攻破操作系统

OS理论原理

设备管理



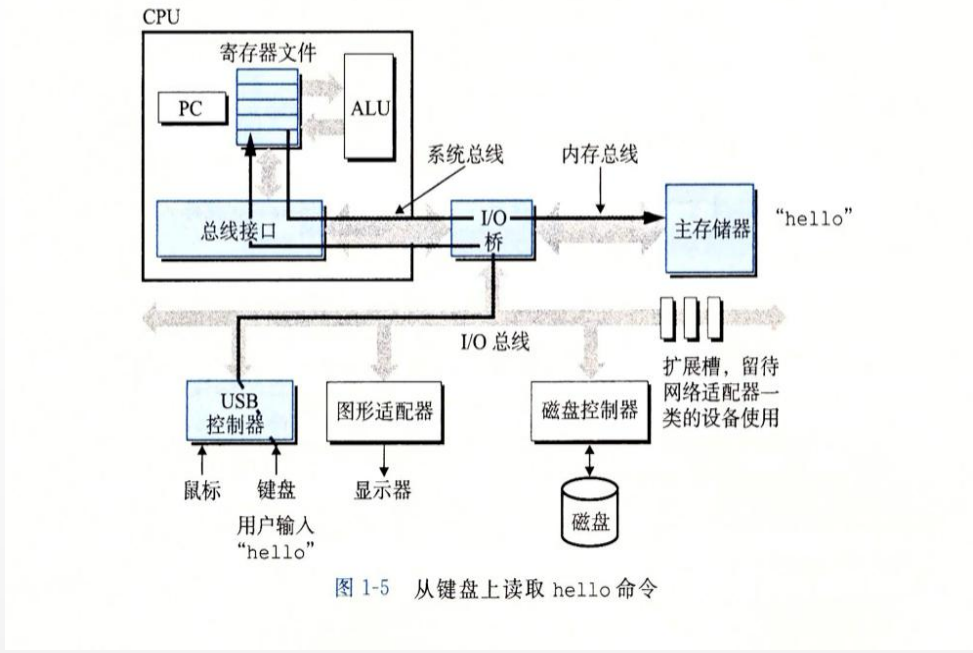


图 1-5 从键盘上读取 hello 命令

驱动程序

设备地址空间

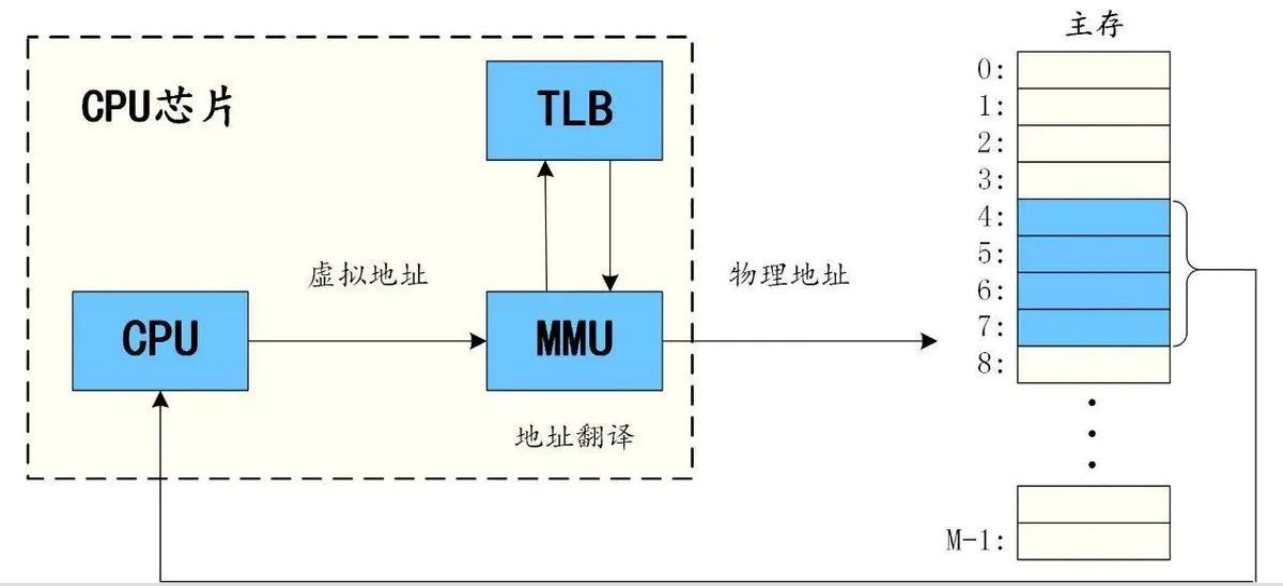
总线

如何学习和攻破操作系统

OS理论原理

内存管理

虚拟内存



内存分配

- 物理内存分配
- 虚拟内存分配
- 非/连续内存分配
- 大内存分配
- 小内存分配
- 内存池

分页和分段

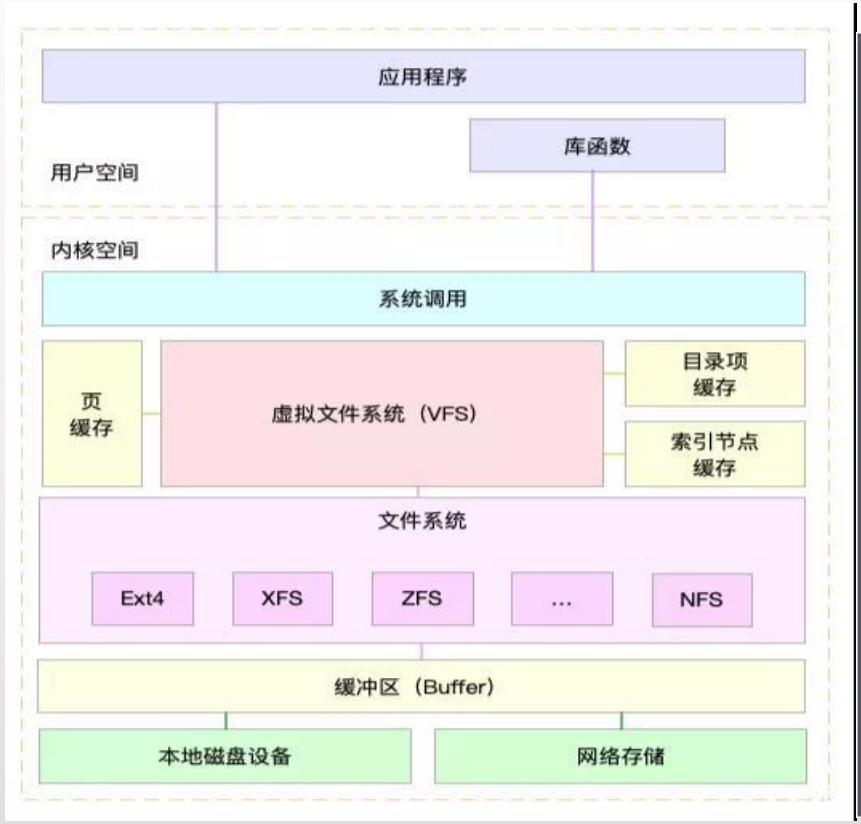
- 分段由逻辑地址到虚拟地址
- 分页由虚拟地址到物理地址

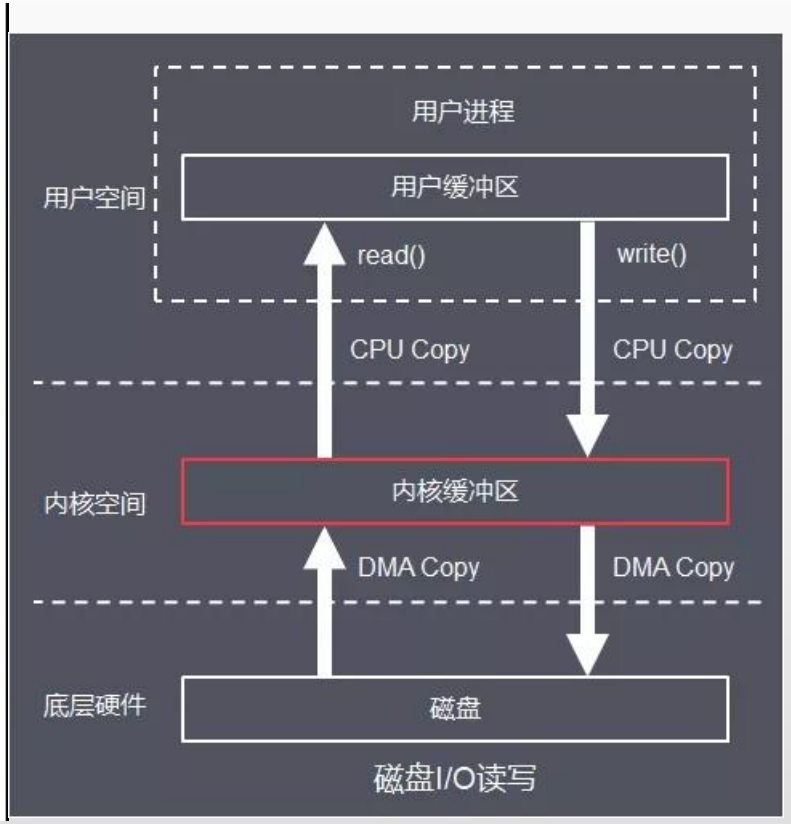
如何学习和攻破操作系统

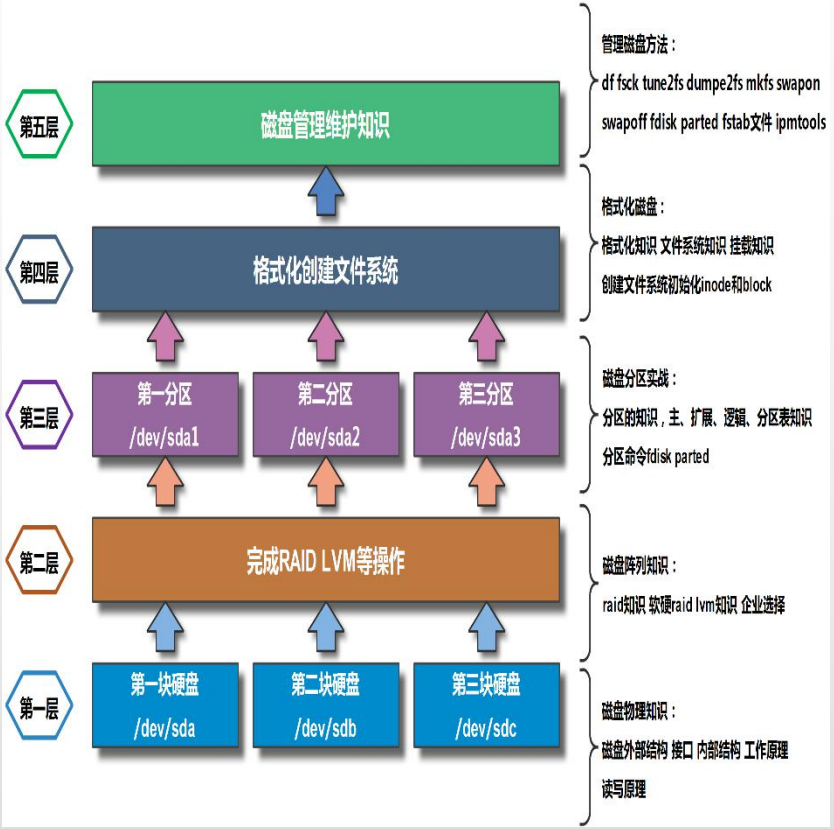
存储管理

OS理论原理

文件系统 ⟷ 文件系统 I/O ⟷ 磁盘管理

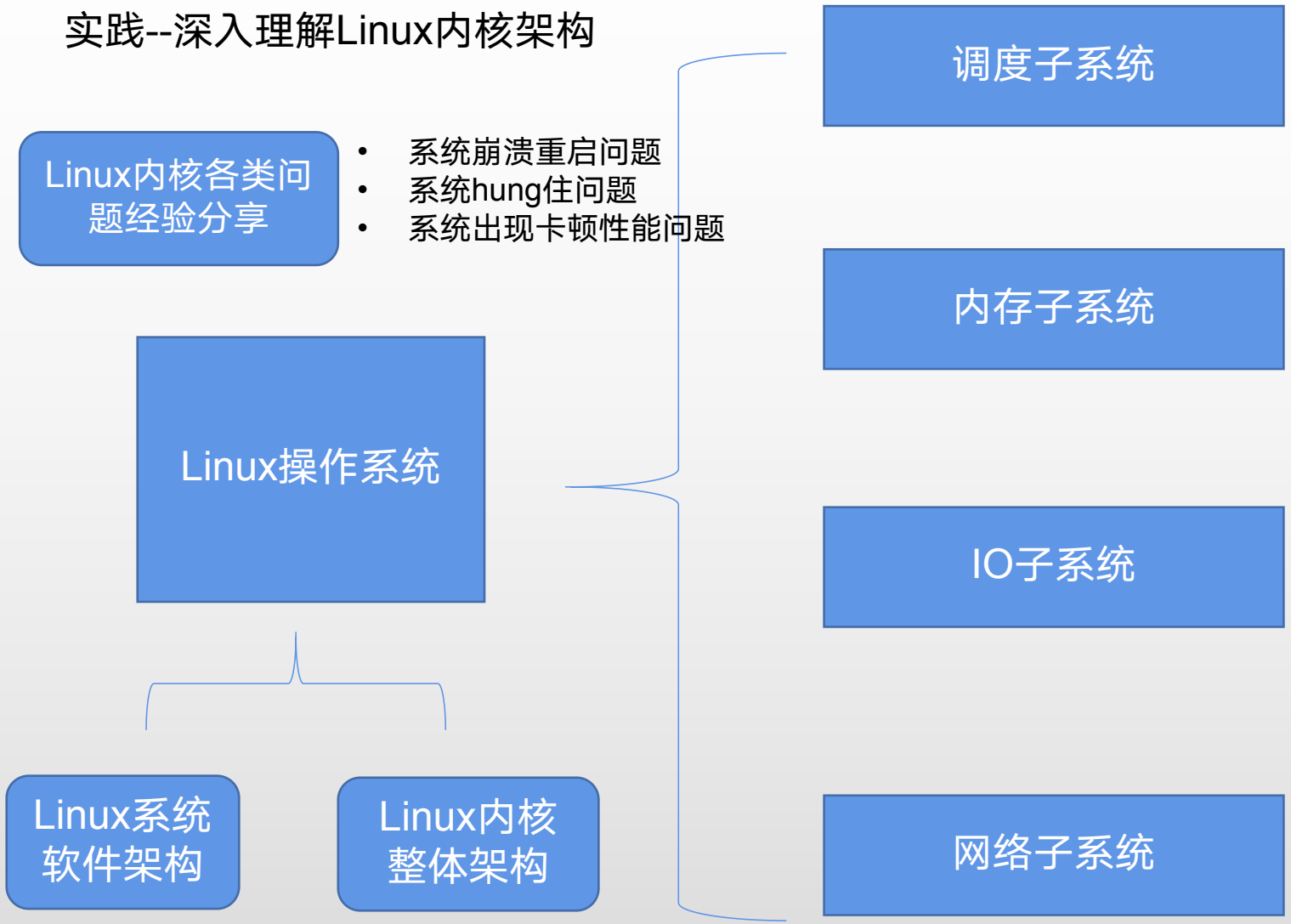






如何学习和攻破操作系统

实践--深入理解Linux内核架构



- CPU
- 抢占
- 中断
- 时钟
- 进程和线程 （任务）
- 多核调度

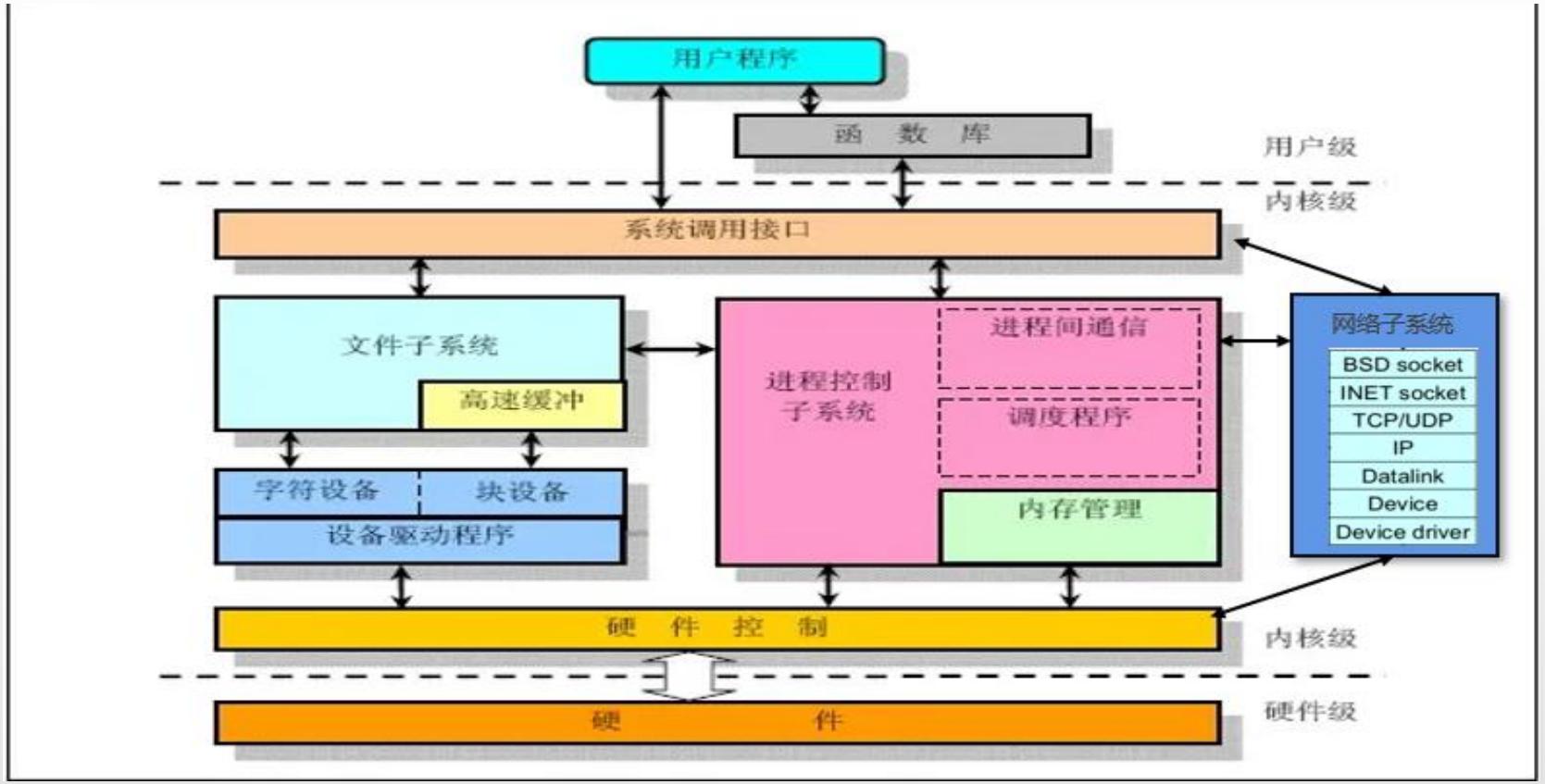
- 物理内存管理
- 虚拟内存管理
- 内存层次结构
- 内存空间

- 文件系统和磁盘
- 存储IO栈
- 缓存buffer
- 高性能 I/O

- 网卡驱动
- 内核协议栈
- 高性能网络
- 网络优化

如何学习和攻破操作系统

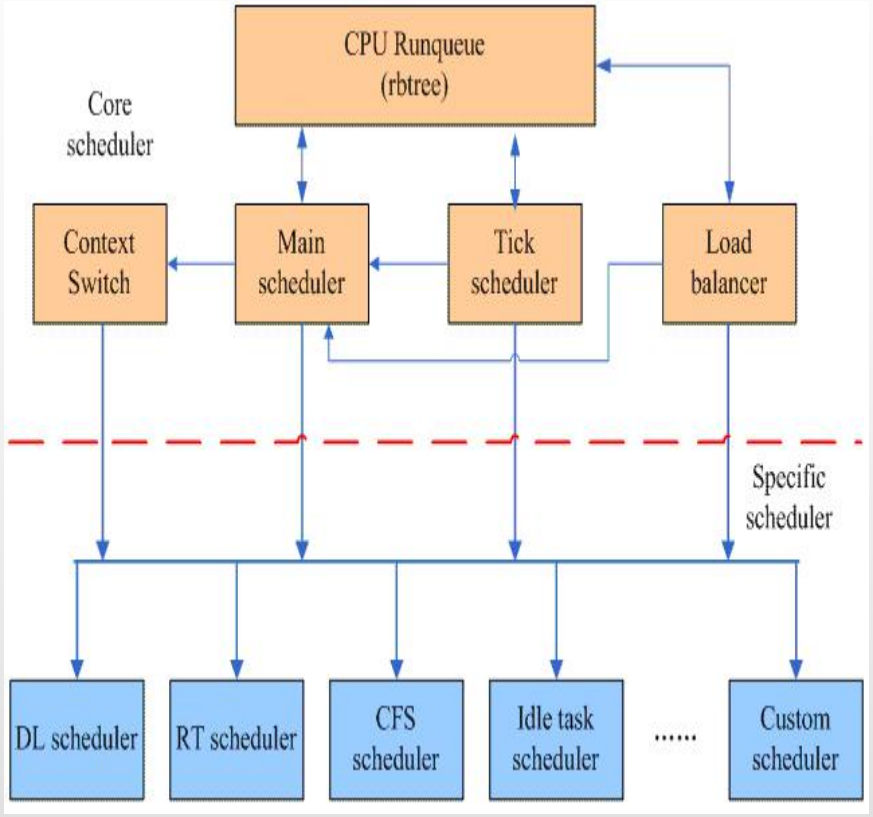
深入理解Linux内核架构-总体认识

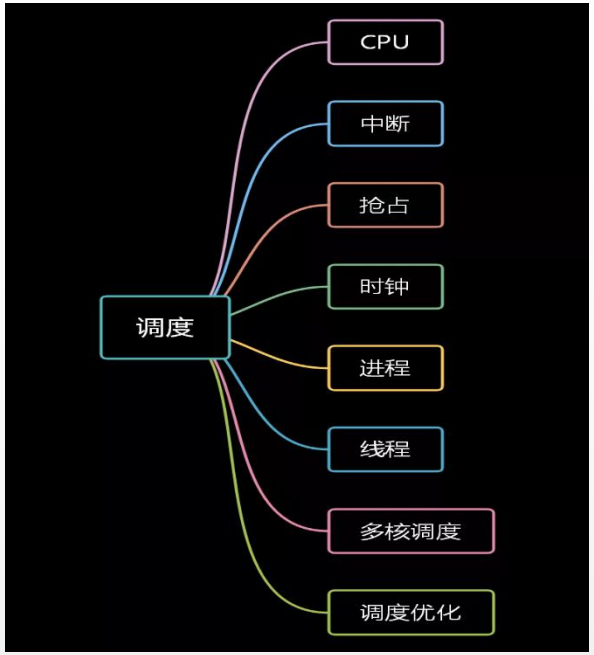


如何学习和攻破操作系统

深入理解Linux内核架构-调度子系统

调度框架

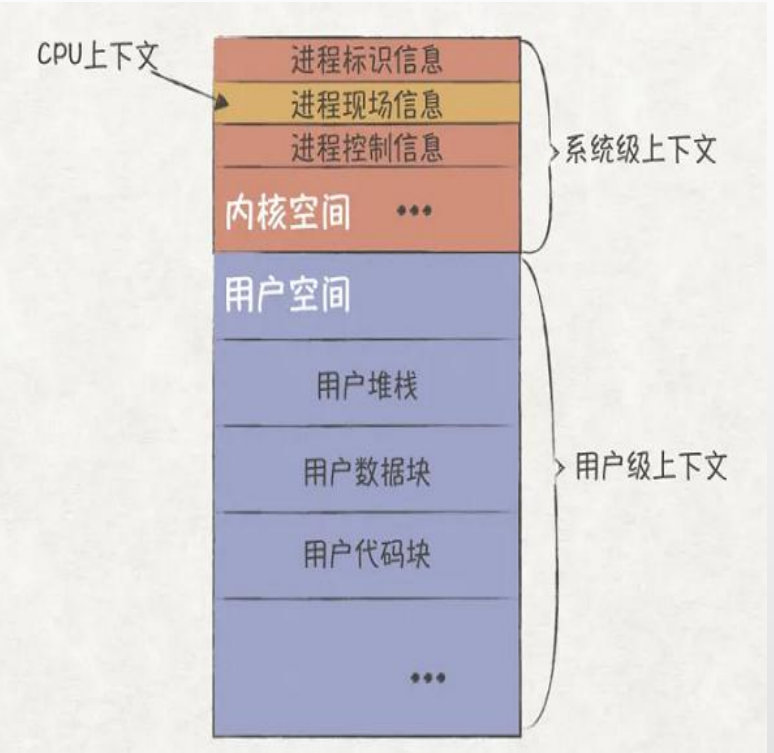




调度上下文

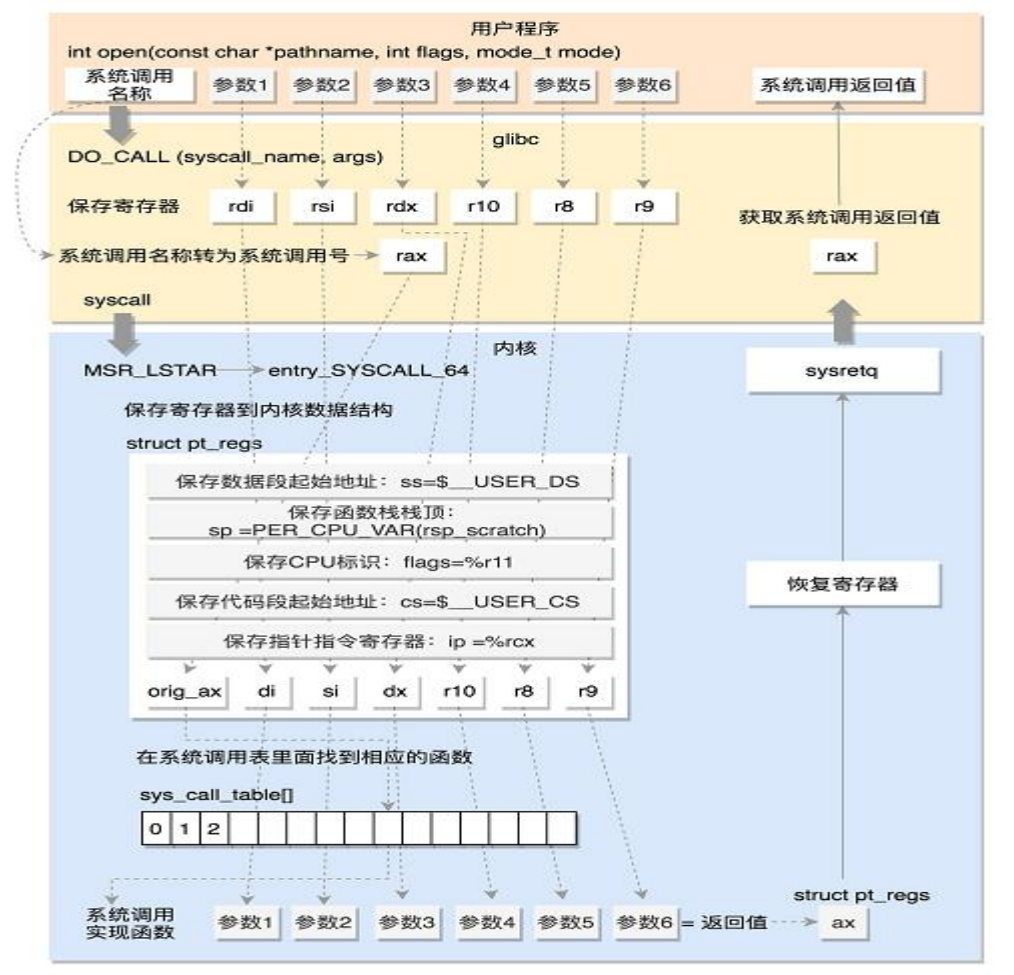
- 中断上下文
- 进程上下文
- 系统调用上下文
- 协程上下文（非内核）

CPU调度原理

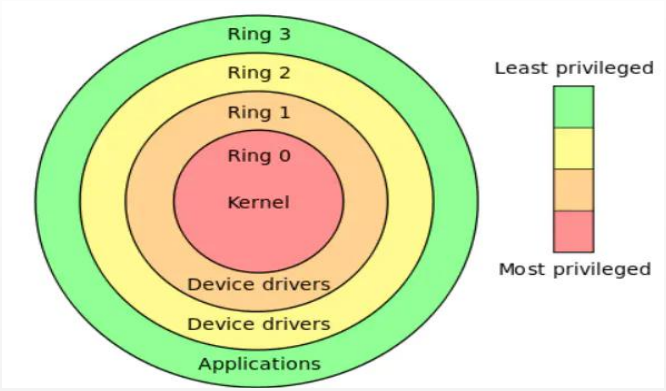


如何学习和攻破操作系统

调度子系统-系统调用原理



模式切换



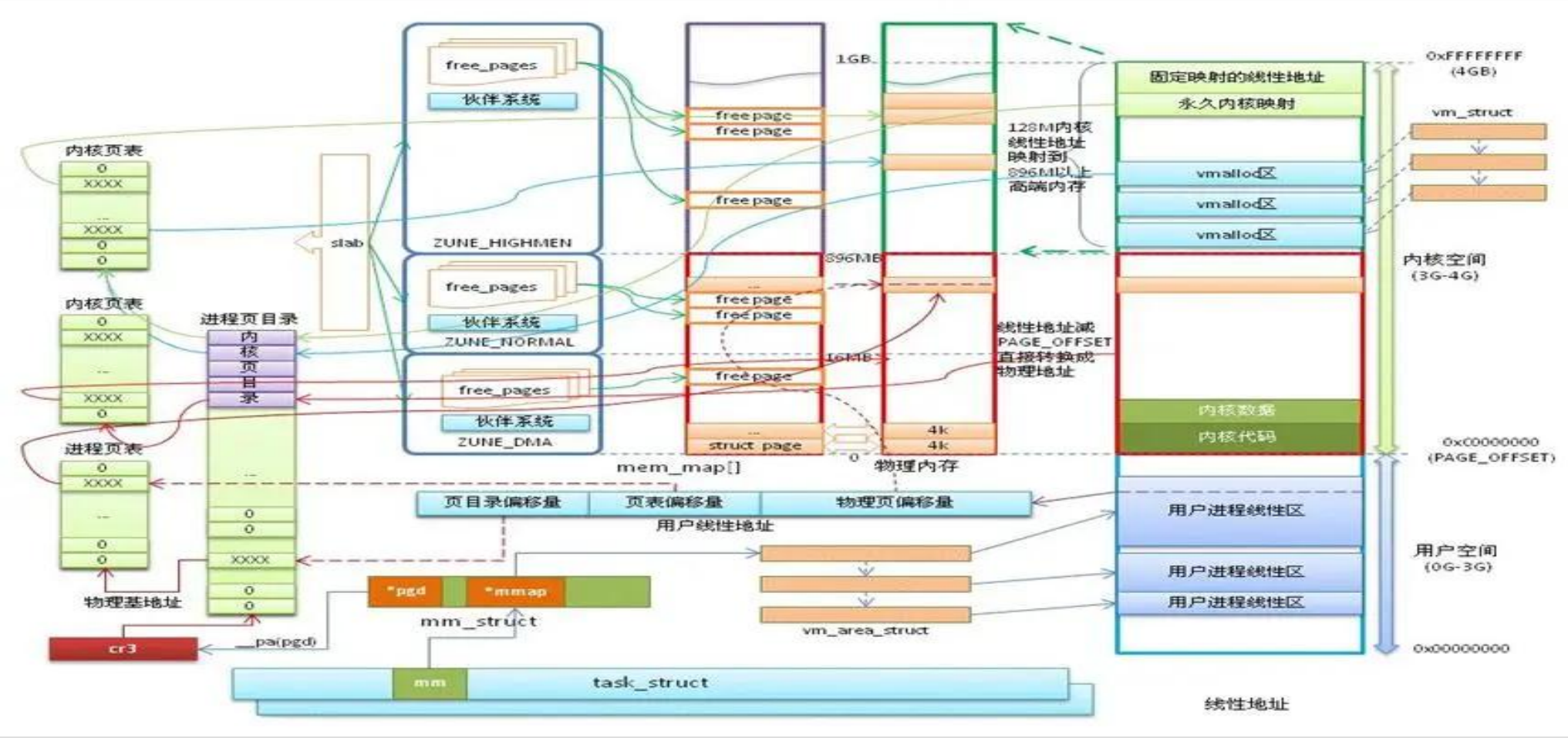
- 内核空间（Ring 0）
- 用户空间（Ring 3）

CPU模式切换

- 切换时先保存CPU寄存器中的用户态指令
- 再重新更新内核态指令位置
- 最后跳转到内核态运行内核任务
- 当系统调用结束后需要恢复原来保存的用户态

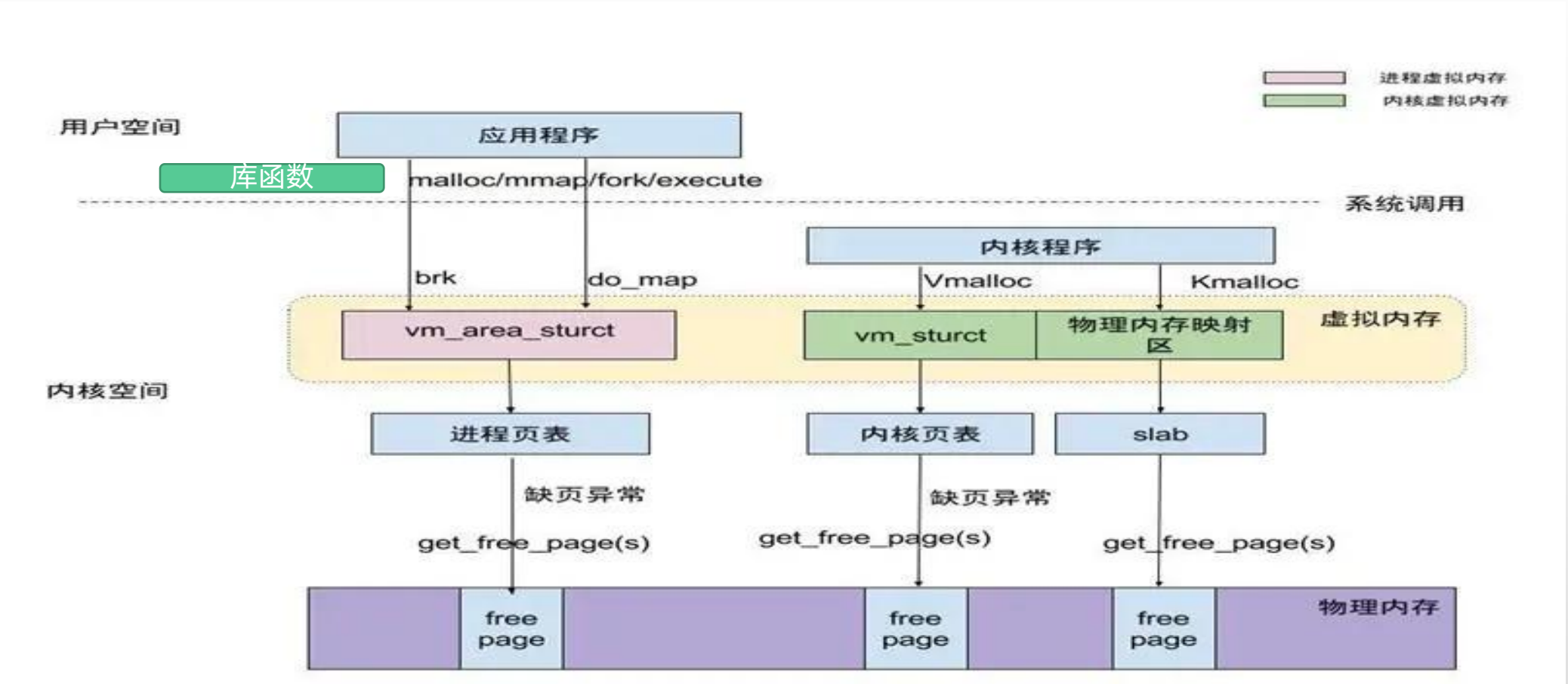
如何学习和攻破操作系统

深入理解Linux内核架构-内存子系统



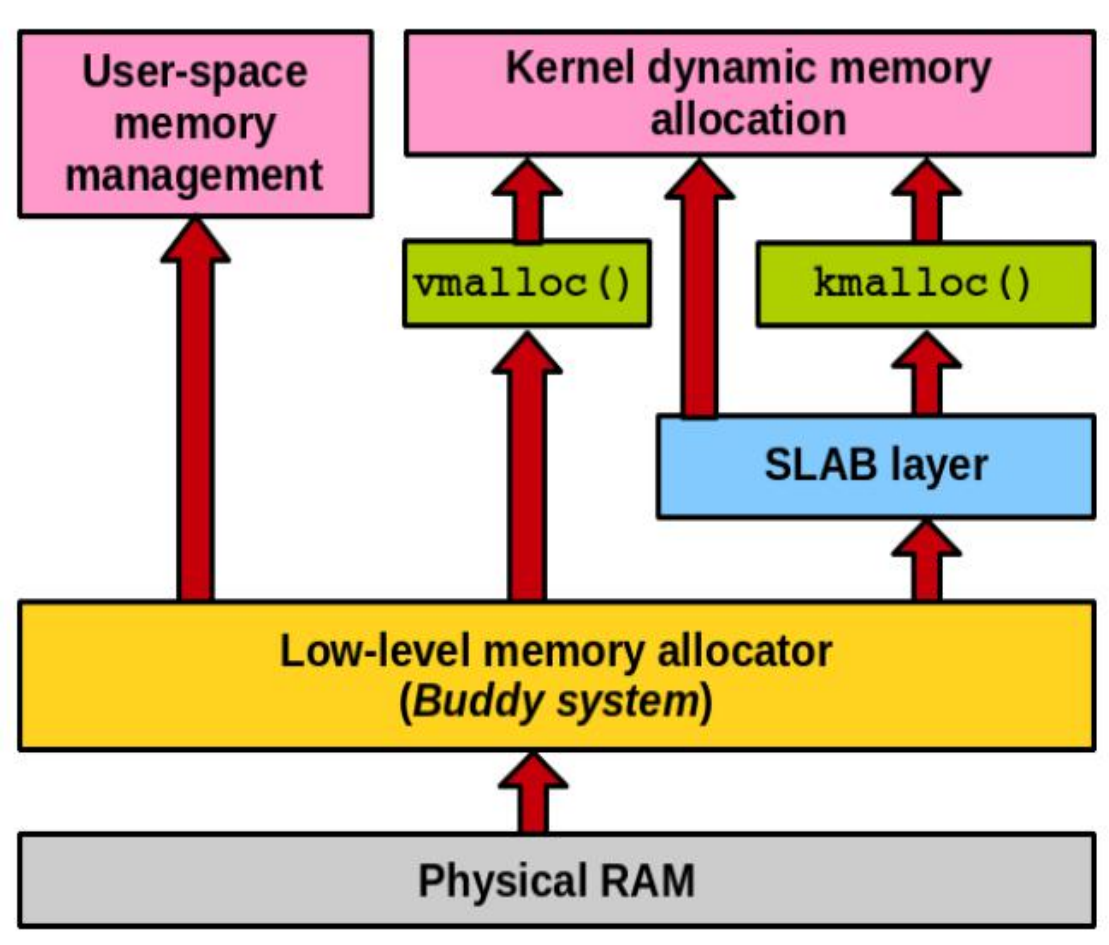
如何学习和攻破操作系统

深入理解Linux内核架构- Linux内存整体架构

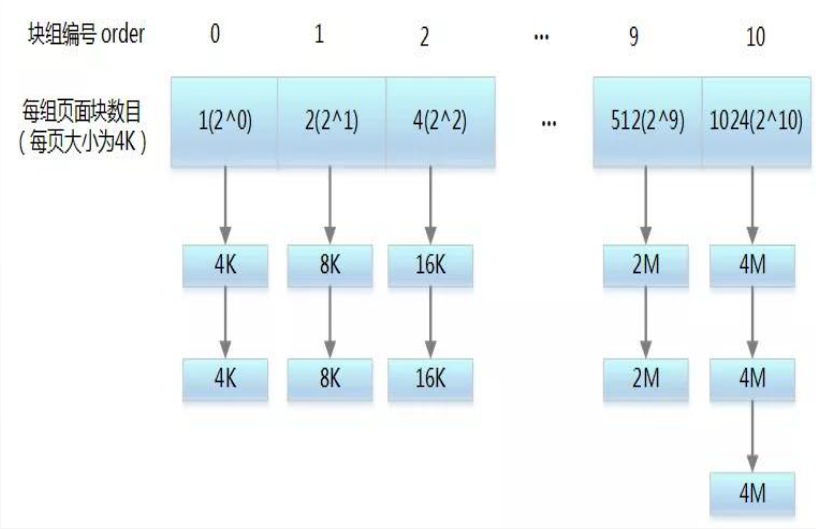


如何学习和攻破操作系统

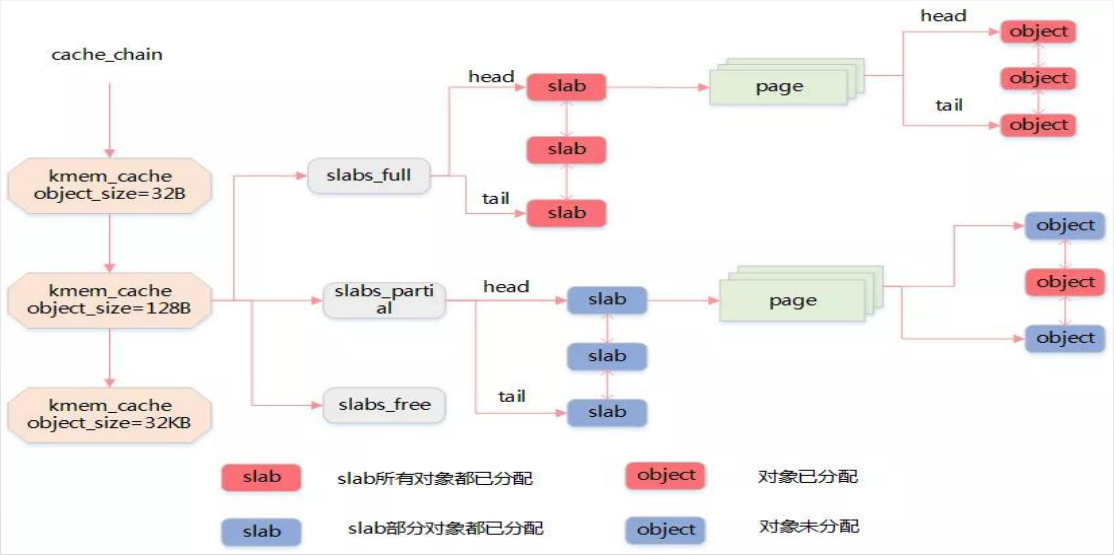
深入理解Linux内核架构-内存子系统



Buddy系统

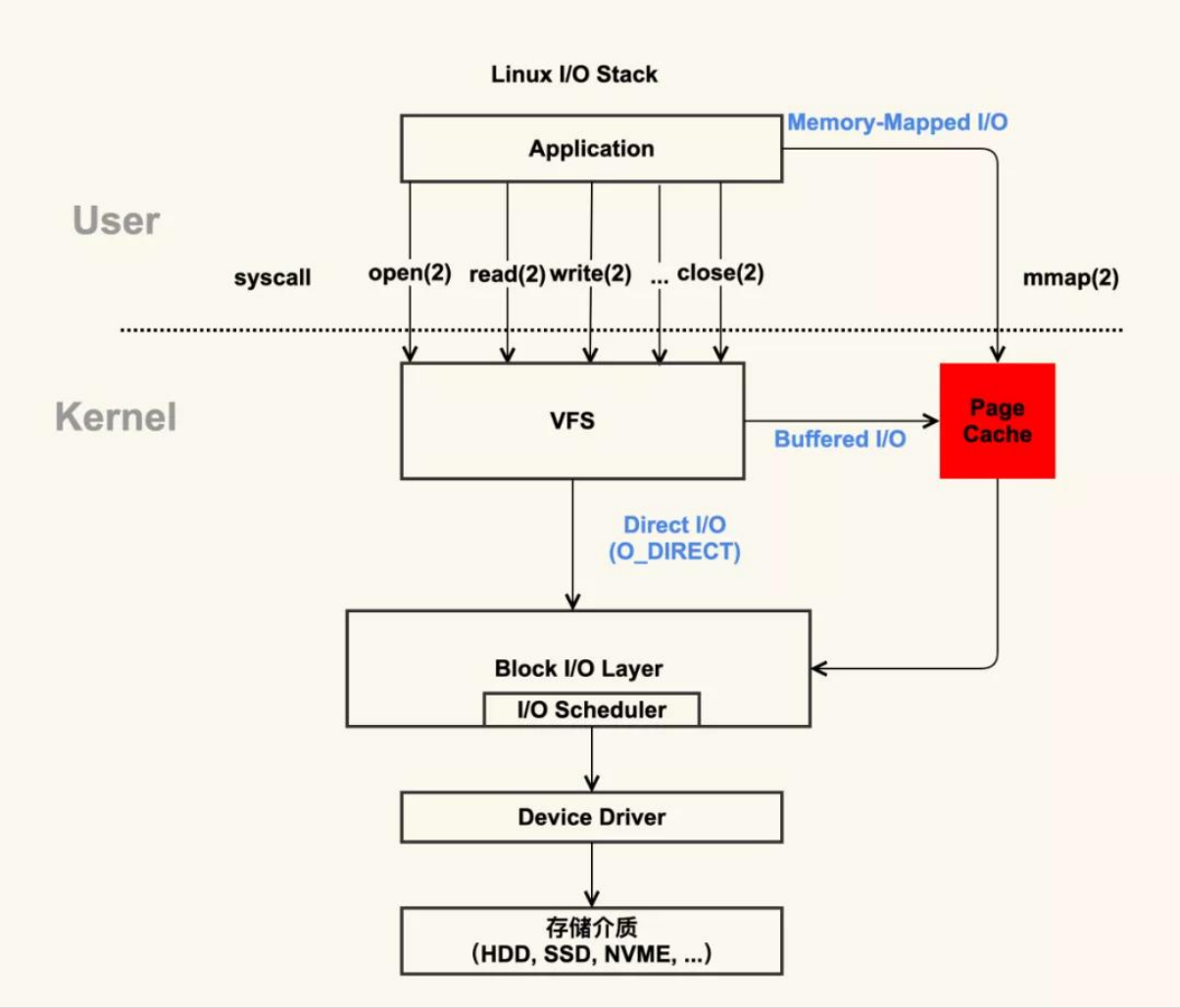


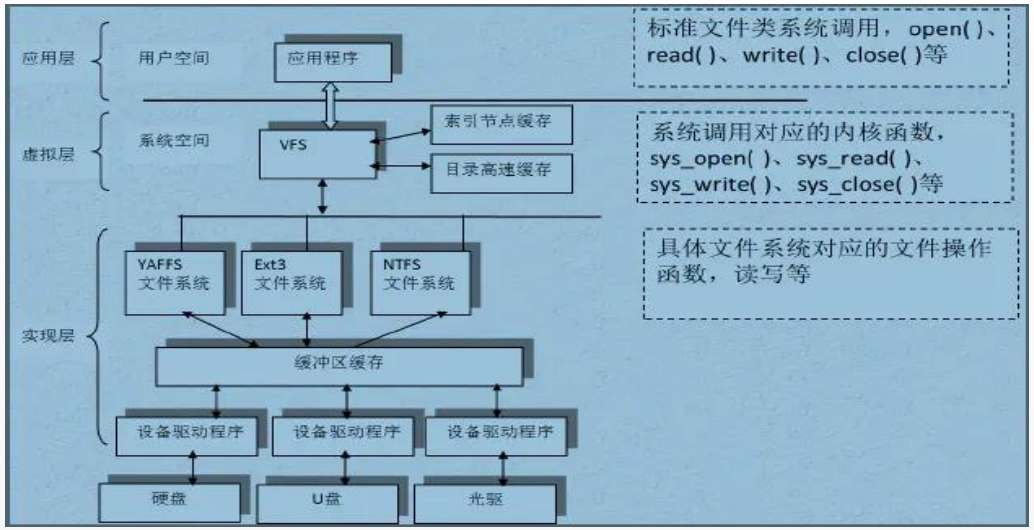
Slab算法

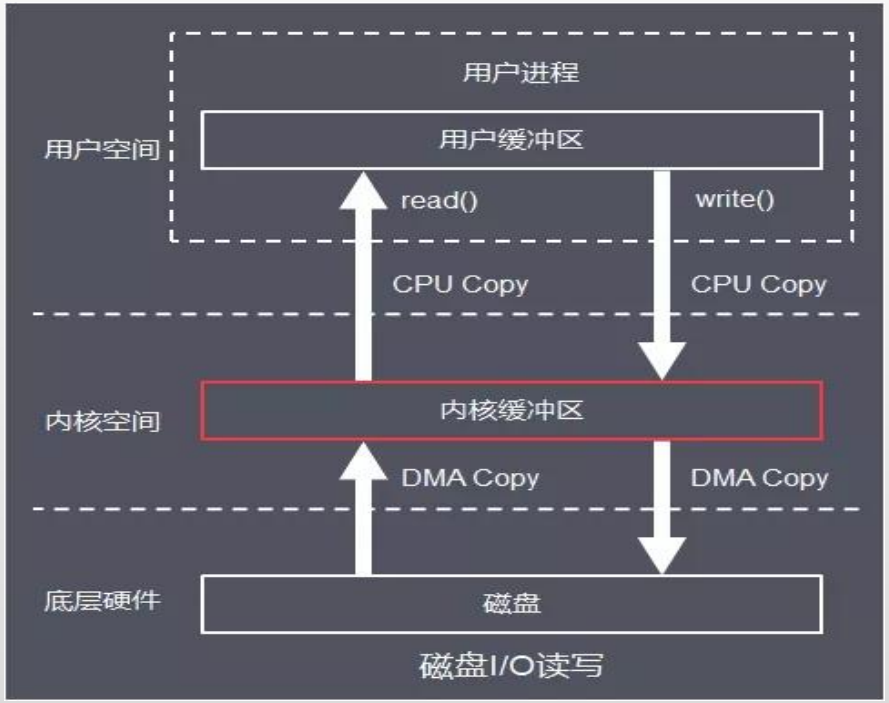


如何学习和攻破操作系统

深入理解Linux内核架构-IO子系统

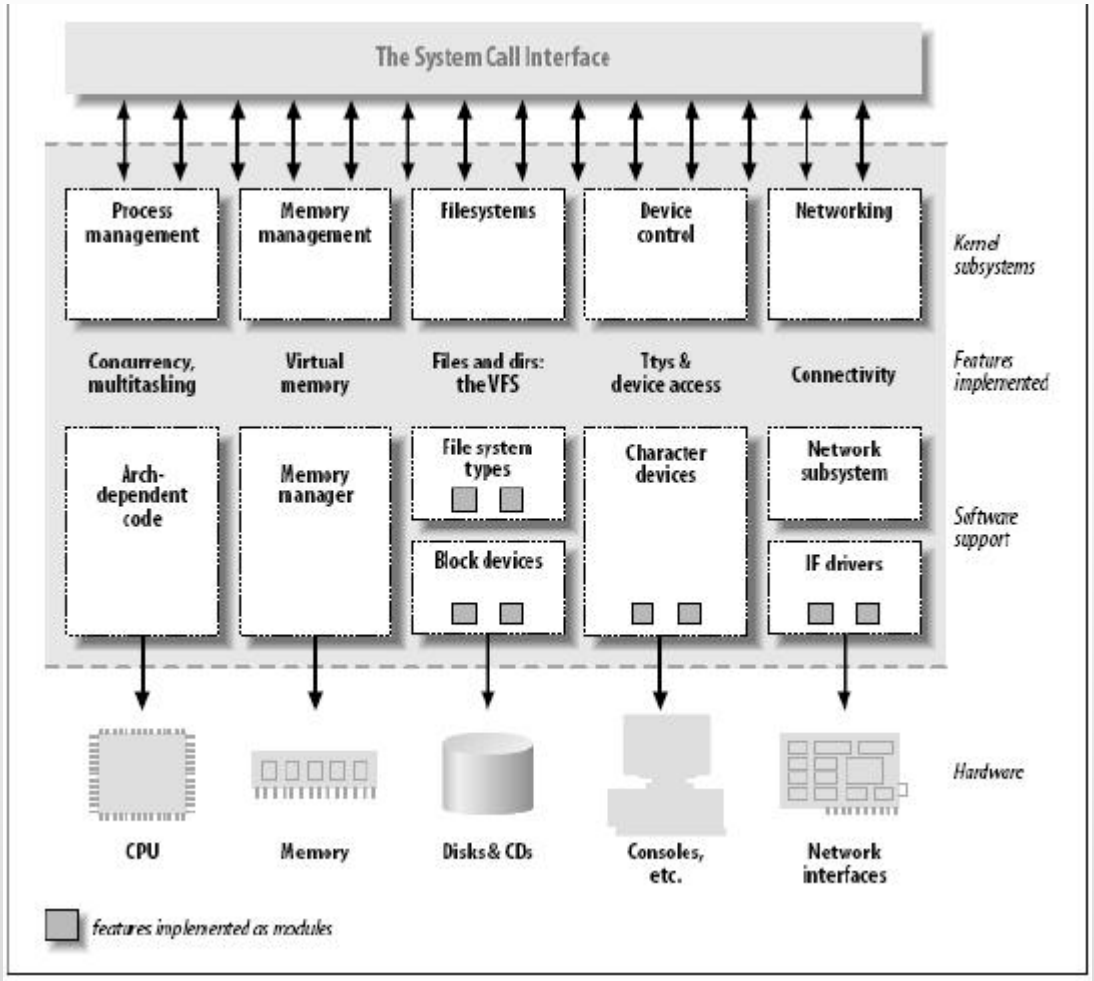


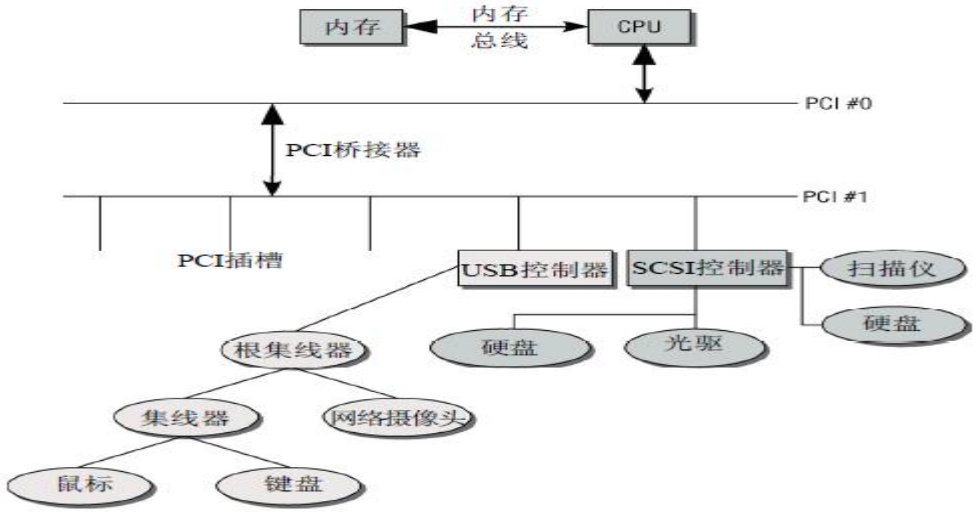


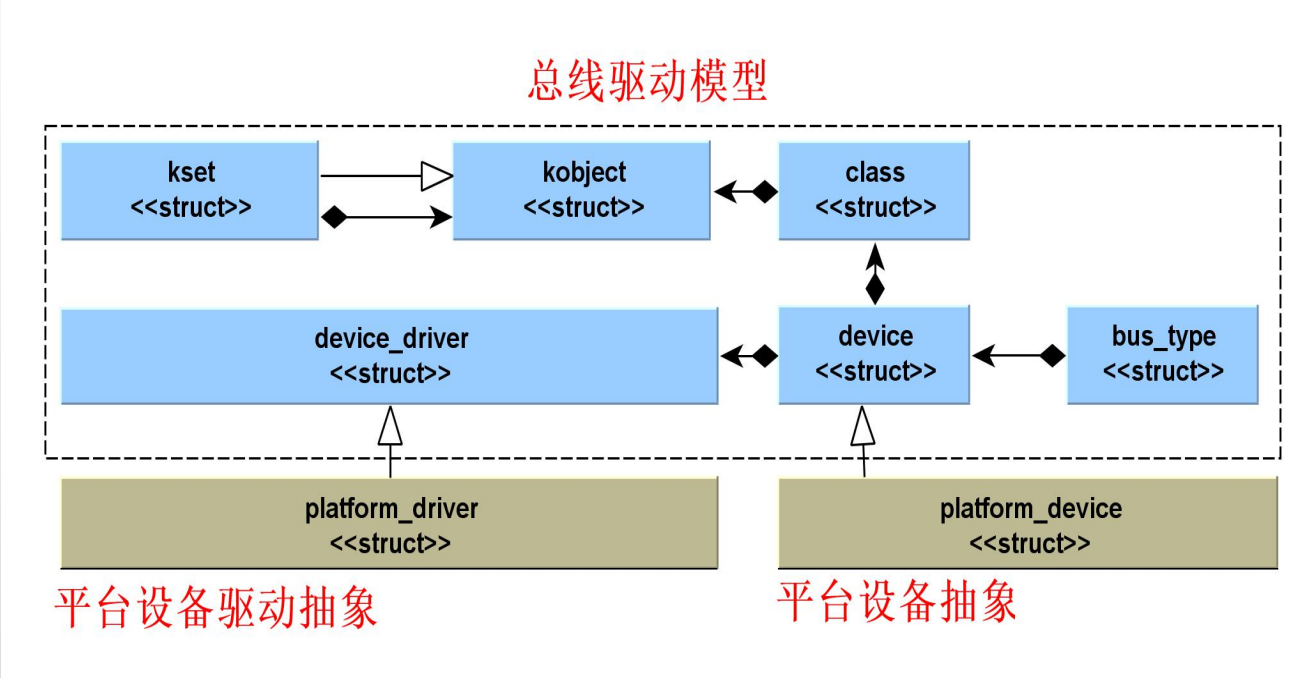


如何学习和攻破操作系统

深入理解Linux内核架构-驱动子系统

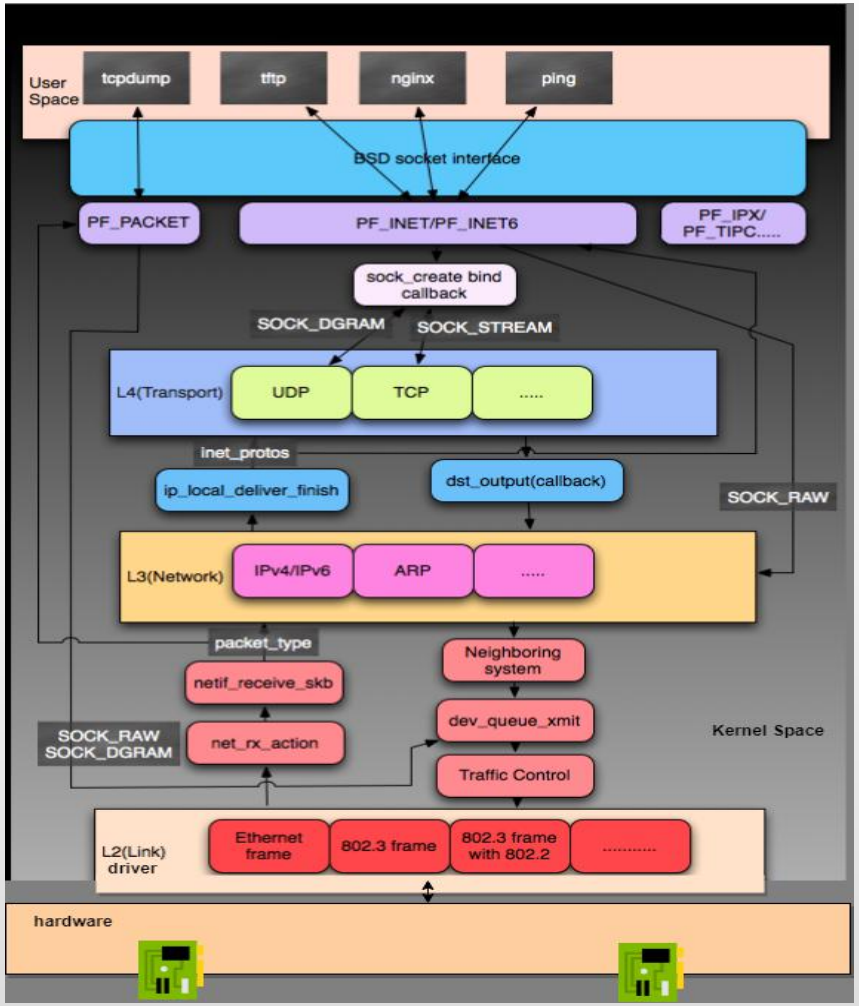


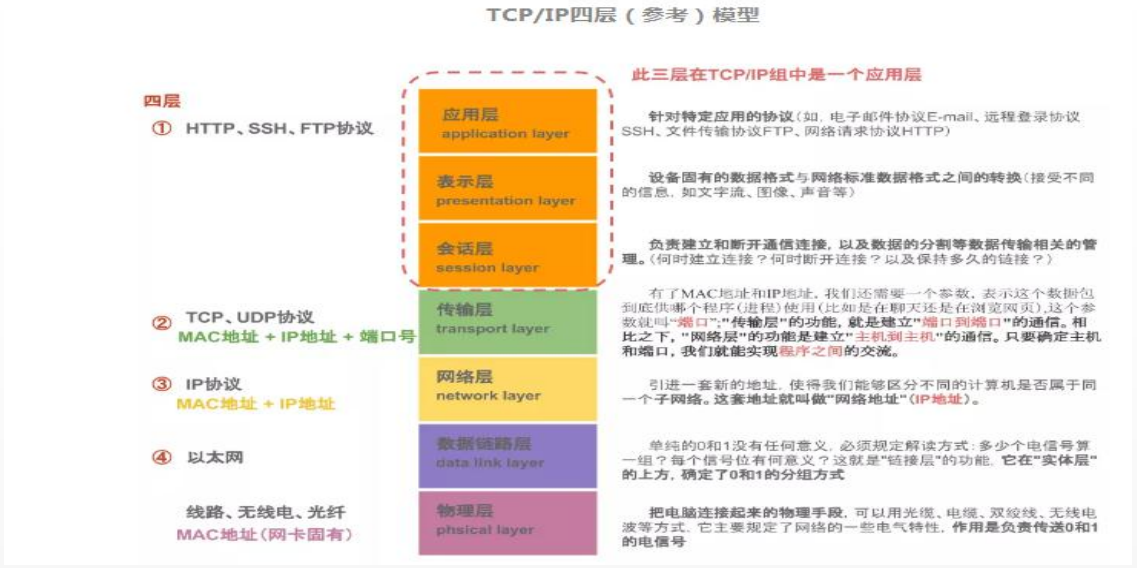


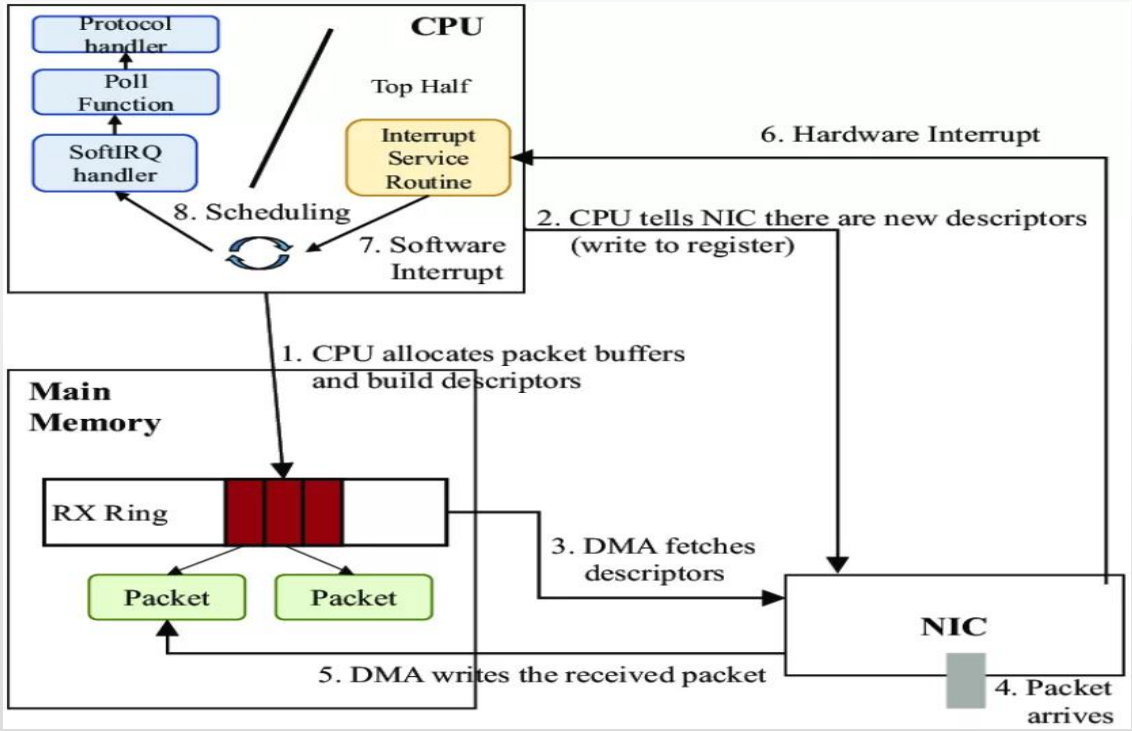


如何学习和攻破操作系统

深入理解Linux内核架构-网络子系统

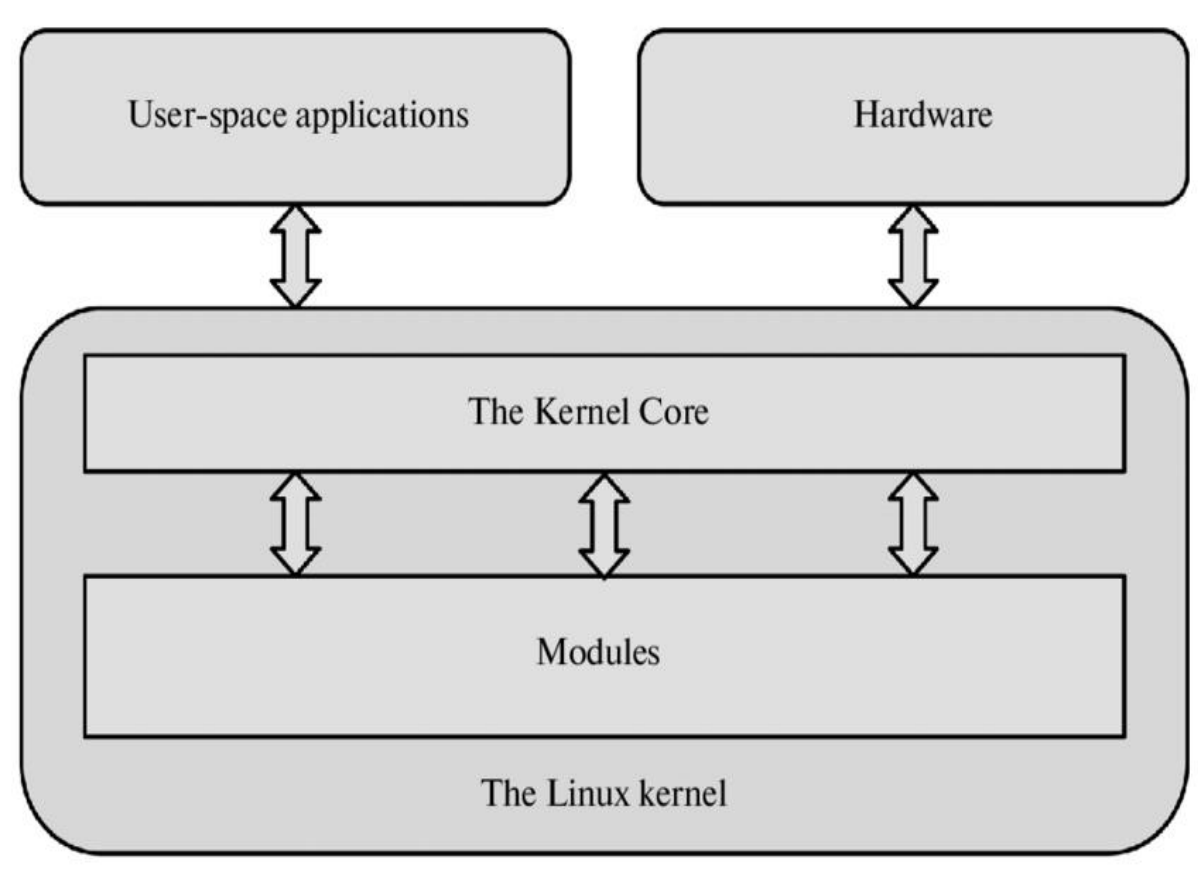


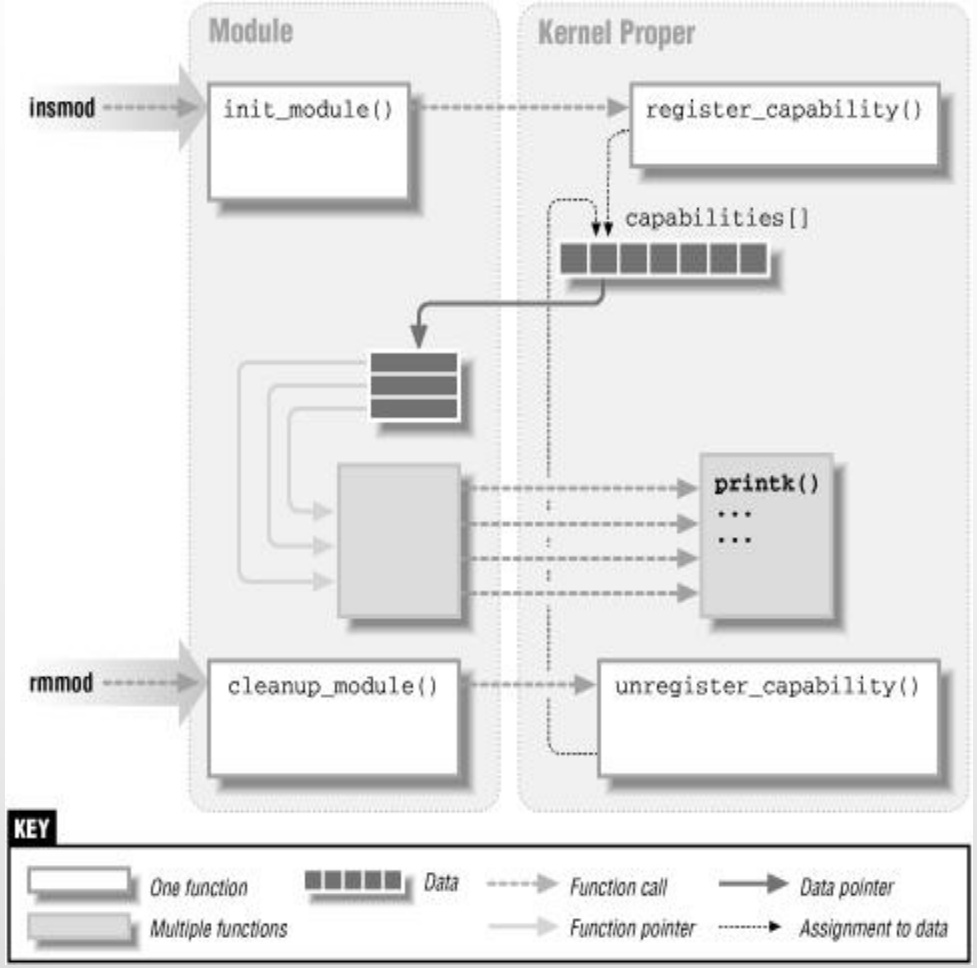




如何学习和攻破操作系统

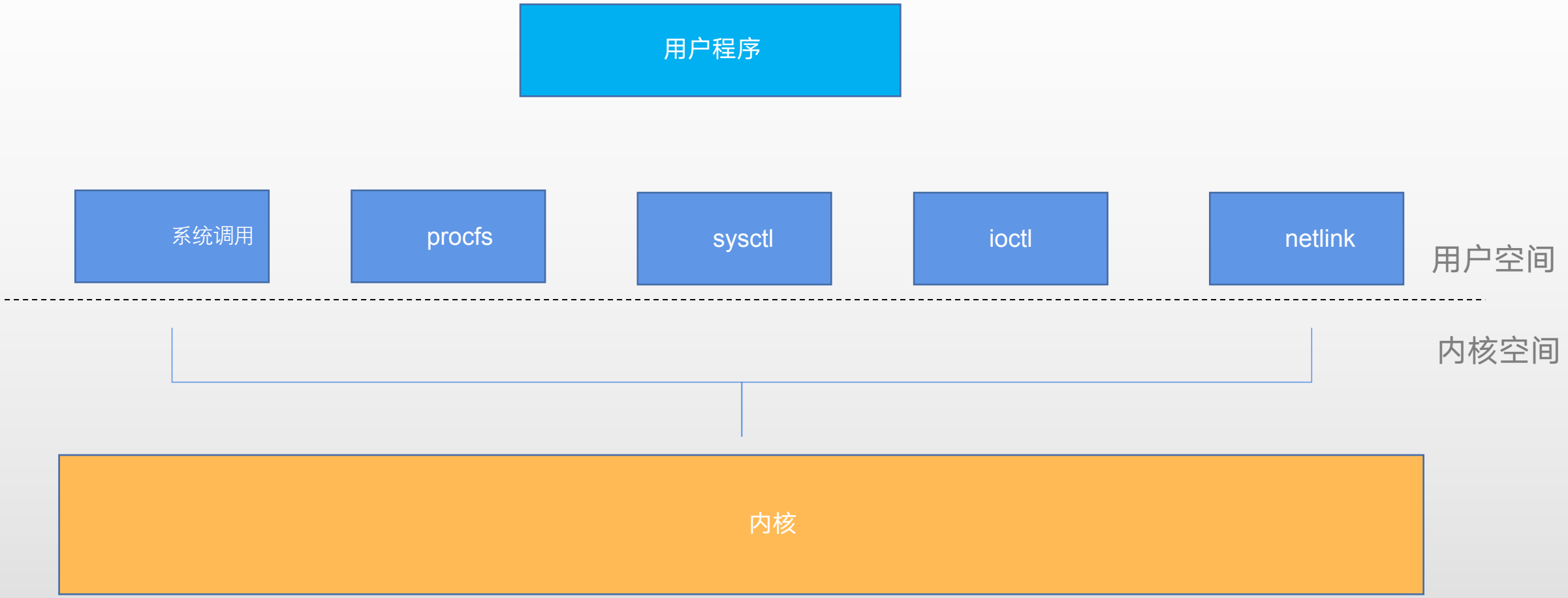
深入理解Linux内核架构-模块机制





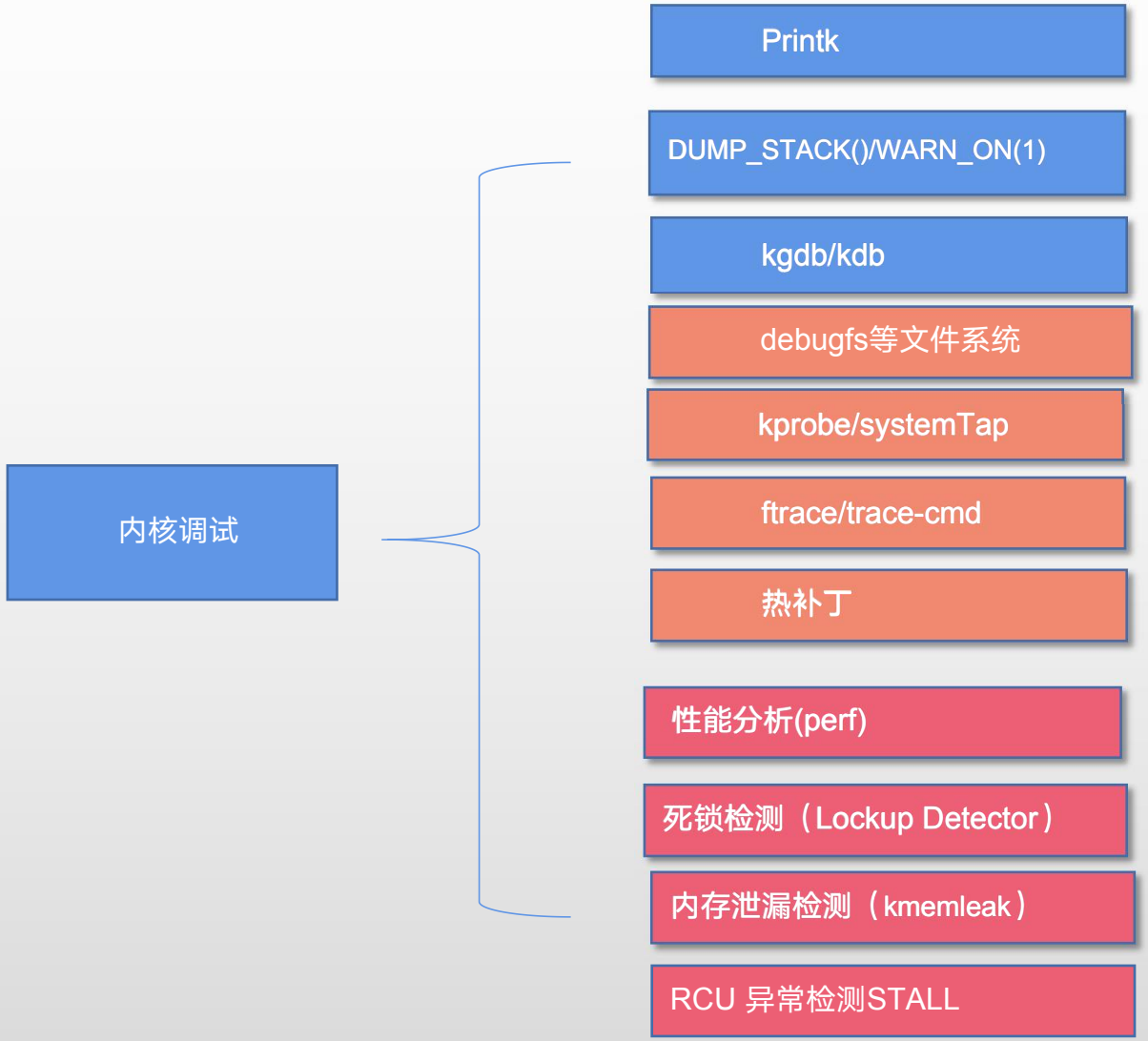
如何学习和攻破操作系统

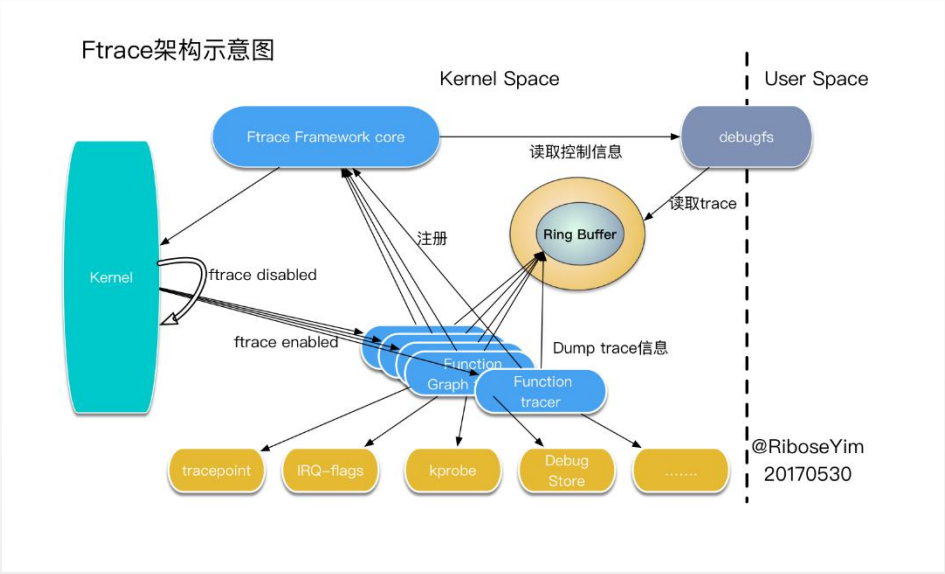
深入理解Linux内核架构-**内核通信方式**

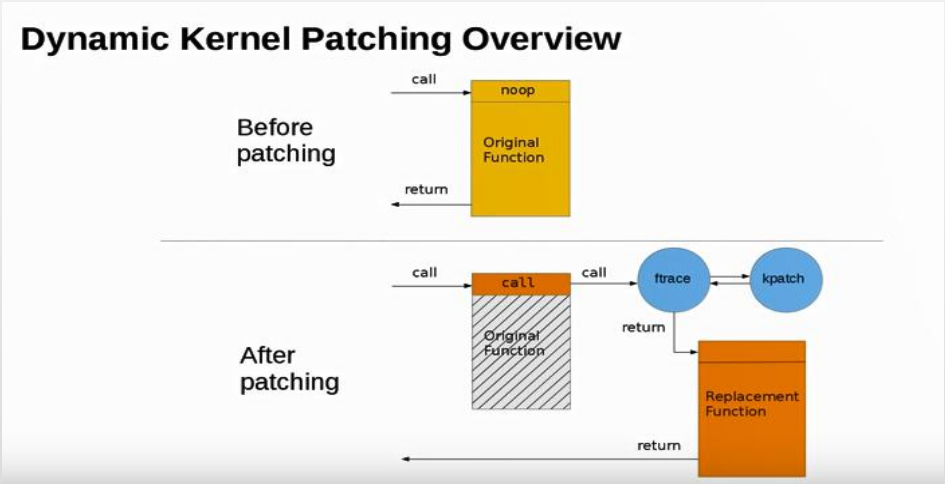


如何学习和攻破操作系统

Linux内核各类问题经验分享—内核调试







如何学习和攻破操作系统

Linux内核各类问题经验分享

系统崩溃重启问题

- **软件bug**
- **soft lockup**
- **hard lockup**
- **指令异常abort**
- **system error**

系统hung住问题

- **死锁**
- **长时间D状态**

系统卡顿/性能问题

Linux内核-系统崩溃重启问题

常见问题

- CPU-hung软锁问题（softlockup: hung tasks）
- 内存非法访问（空指针，非法页访问）
- 触发内核BUGON（内核主动crash，代码执行状态不符合预期）
- CPU指令跳变（非法访问内存，堆栈逻辑错误）

核心思想：

大胆猜测+搜索，细心推导参数和当前函数里面变量的值，然后结合反汇编和异常指令等，综合推导出代码逻辑错误或者非软件问题等

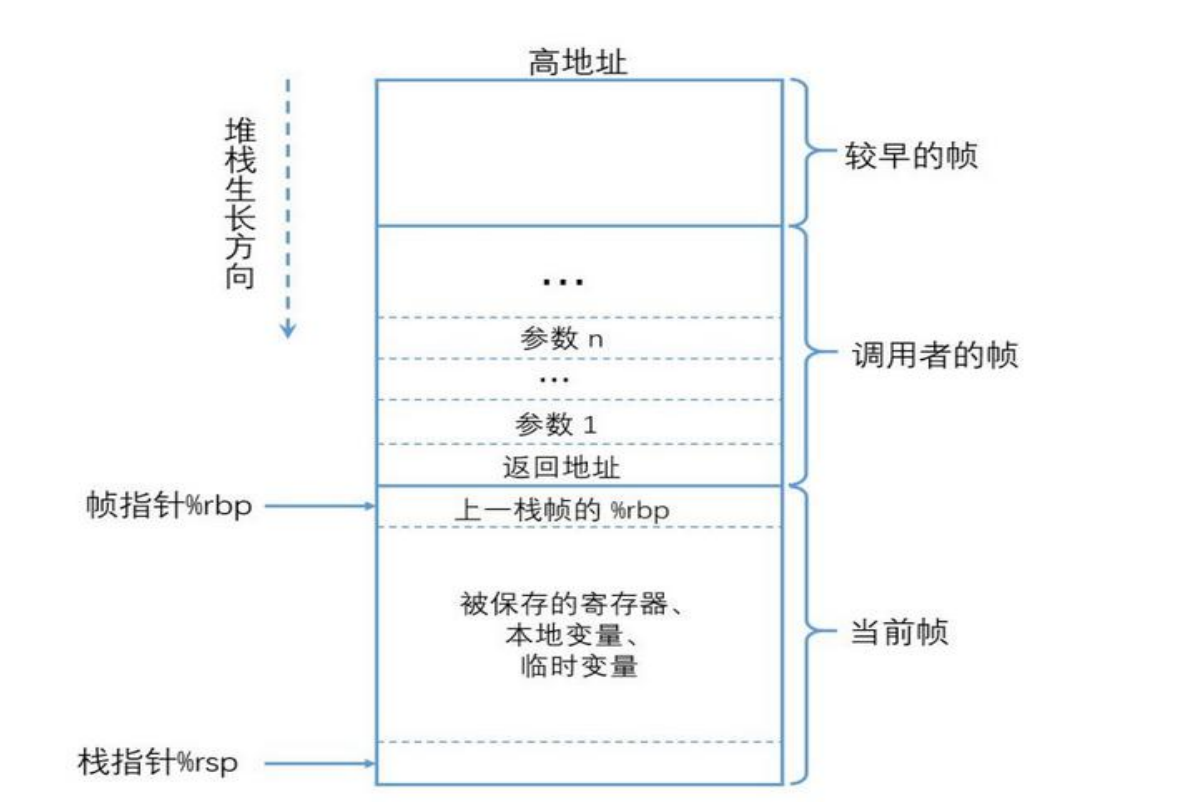
- 了解堆栈原理, bt 查看堆栈，bt –ff/-FF 查看详细堆栈信息,bt –a 查看所有CPU堆栈信息；
- 了解x86_64汇编原理，各个寄存器作用，有些寄存器可以被刷掉，经常出现问题寄存器已经被刷，需要去推导堆栈信息；
- 参数推到：整数参数（包括指针）按顺序放在寄存器%rdi，%rsi，%rdx，%rcx，%r8和%r9中，函数的返回值存储在%eax中；
- 熟悉Crash工具，struct /list 等 命令打印数据结构信息和字段偏移等，配合内存地址可以打印当前结构内存数据（非常重要），方便对照汇编代码进行推理逻辑；
- 对照代码（内核代码）尝试查看全局变量或者per CPU数据得到一些系统运行信息，方便定位问题，比如时钟中断统计，watchdog信息，CPU运行队列等；

深入理解堆栈原理

如何从堆栈中查找

- 被压栈的寄存器参数
- 本地变量

进而找到函数调用参数



Register	Usage	Preserved across function calls
%rax	1st return register, number of vector registers used	No
%rbx	callee-saved register; base pointer	Yes
%rcx	pass 4th integer argument to functions	No
%rdx	pass 3rd argument fo functions, 2nd return register	No
%rsp	stack pointer	Yes
%rbp	callee-saved register, frame pointer	Yes
%rsi	used to pass 2nd argument to functions	No
%rdi	used to pass 1st argument to functions	No
%r8	used to pass 5th argunent to functions	No
%r9	used to pass 6th argument to functions	No
%r10	temp register, used for passing a function's static chain ptr	No
%r11	temp register	No
%r12	callee-saved register	Yes
%r13	callee-saved register	Yes
%r14	callee-saved register	Yes
%r15	callee-saved register	Yes

如何学习和攻破操作系统

Linux内核各类问题经验分享--**系统hung住问题**

- 在内核中可以通过配置如下debug选项来对系统hung住的情况进行检测：

  CONFIG_DETECT_HUNG_TASK=y
  CONFIG_DEFAULT_HUNG_TASK_TIMEOUT=120
  CONFIG_BOOTPARAM_HUNG_TASK_PANIC=y

  以上配置的情况会在一个进程进入D状态时间超过120秒后，打印出对应的stack trace信息。如果配置了 CONFIG_BOOTPARAM_HUNG_TASK_PANIC 那么会更进一步使得内核直接panic，此时系统表现就不是hung住了，而是系统崩溃重启。

- 一般在系统hung住时，假如系统没有自动进入panic，那么可以手动触发一个crash，抓出coredump进行分析，查找故障现场的task信息，寻找对应的D状态的进程，并找出该进程的栈回溯，然后找到问题所在。

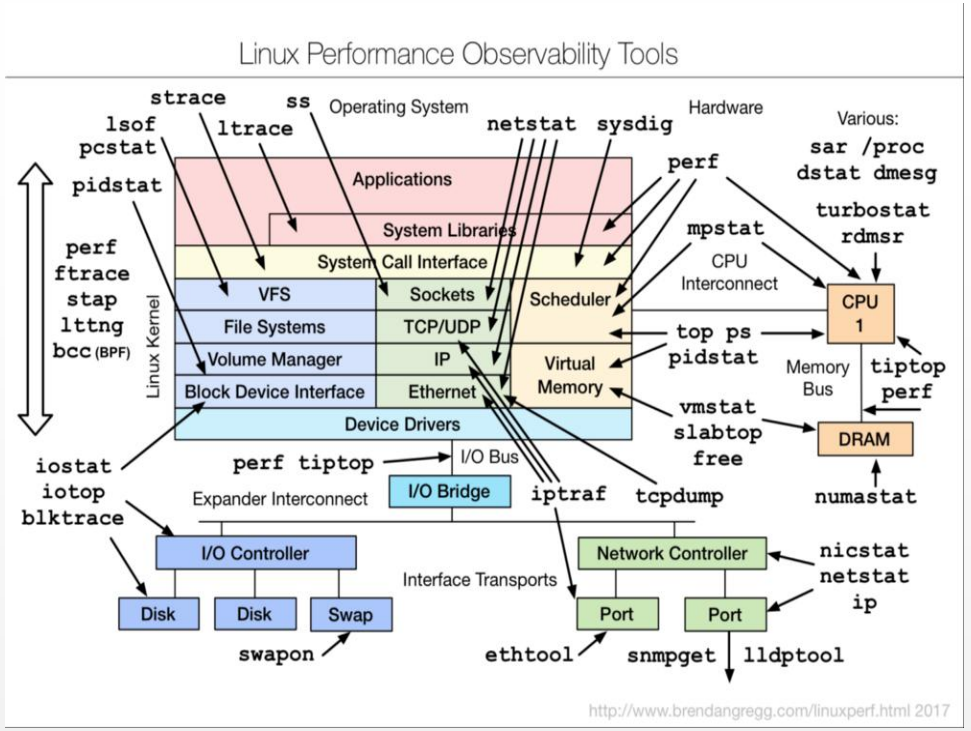
如何学习和攻破操作系统

Linux内核各类问题经验分享-**系统卡顿/性能问题**

系统出现短暂卡顿，响应慢，这实际上属于性能问题，而不是故障问题。

- 可能也是系统中的负载过重，或者很多进程在等待IO，也就是处于不可中断的睡眠状态（D状态）。

- 可以利用top、iostat、mpstat、ps、ftrace、perf等等工具进行进一步的定位。
  - 使用top命令进一步确认具体是原因导致的负载过高
    - CPU这一列wa的数值很高，说明CPU资源在等待I/O，由此可以判断是因为I/O的原因导致了系统负载过高，操作卡顿.
  - 使用iostate命令确认具体I/O异常
    - 如果%iowait的值过高，表示硬盘存在I/O瓶颈
    - 如果%idle值高但系统响应慢时，有可能是CPU等待分配内存，此时应加大内存容量
    - %idle值如果持续低于10，那么系统的CPU处理能力相对较低，表明系统中最需要解决的资源是CPU
  - 使用iotop或者ps查找导致高I/O的进程。
  - mpstat查看多核系统中每个CPU的当前运行状况信息，主要针对软中断打爆问题等。
  - ftrace 的作用是帮助开发人员了解 Linux 内核的运行时行为，以便进行故障调试或性能分析。



- perf是以时间点触发事件采样获取程序运行的时间分布。主要针对以下三种事件

  1、Hardware Event 是由 PMU 硬件产生的事件，比如 cache 命中，当您需要了解程序对硬件特性的使用情况时，便需要对这些事件进行采样；

  2、Software Event 是内核软件产生的事件，比如进程切换，tick 数等；

  3、Tracepoint event 是内核中的静态 tracepoint 所触发的事件，这些 tracepoint 用来判断程序运行期间内核的行为细节，比如 slab 分配器的分配次数等。

如何学习和攻破操作系统--推荐资料

书籍

- 《操作系统导论》
- 《操作系统：设计与实现》
- 《操作系统—精髓与设计原理》
- 《Linux内核设计与实现》
- 《深入Linux内核架构》
- 《Linux 内核情景分析》
- 《深入理解Linux内核》
- 《深入理解计算机系统》

经典文章

虚拟内存精粹
深入理解 Linux的 I/O 系统
深入理解Linux内存子系统
深入理解虚拟化
Linux网络子系统
Linux Kernel TCP/IP Stack
Linux调度系统全景指南(上篇)
Linux问题分析与性能优化
深入理解Linux 的Page Cache

课程

极客时间：

- 趣谈Linux操作系统
- 极客时间-操作系统45讲(实战）

路线

极客星球：

- Linux内核学习路线
- 后台开发基础修炼路线分）

如何学习和攻破操作系统—Q&A

问题答疑