



2023 CCF CHINASOFT

# 中国软件大会

智能化软件创新推动数字经济与社会发展

NASAC

第22届全国软件与应用学术会议

The 22nd National Software and Application Conference

FMAC

第8届全国形式化方法与应用会议

The 8th National Conference on Formal Method and Application

中国·上海 2023年12月1-3日  
SHANGHAI·CHINA December 1-3, 2023





2023 CCF CHINASOFT  
中国软件大会

# 兴趣驱动的操作系统教学探索

报告人：石亮

报告日期：2023.12.01



- 操作系统教学情况
- 理论教学的兴趣驱动
- 实验教学的兴趣驱动
- 翻转课堂-让科研走进教室
- 课程思政-用事实激励学生
- 总结

理论核心  
教师提供  
学生来学

核心问题  
提供什么，如何提供  
学什么，怎么学

## 操作系统课程安排

- 课时：90学时（54理论，36实验）/4学分
- 拔尖班（大二上）、计算机科学与技术学院（大二下）
- 目标：掌握操作系统基础理论、具备操作系统开发能力
- 教材选择：

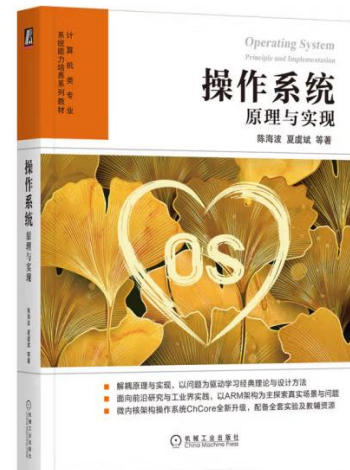


### 课程教材：

书名：《现代操作系统（原书第4版）》  
书号：ISBN978-7-111-58165-9  
作者：[荷] Andrew S. Tanenbaum / [荷] Herbert Bos  
原作名：Modern Operating Systems (4th Edition)  
译者：陈向群 / 马洪兵 等  
出版社：机械工业出版社  
出版时间：2017年10月

### 实验教材：

书名：《现代操作系统：原理与实现》  
书号：ISBN978-7-111-72248-9  
作者：陈海波、夏虞斌等  
出版社：机械工业出版社  
出版时间：2022年12月



## 我们的学生对操作系统学习的看法是什么？

### 如何满足这些需求？

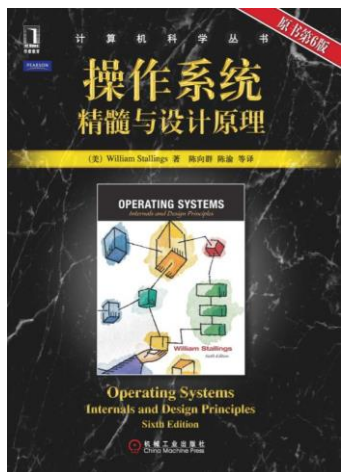
第1排学生： 自我驱动型	第2排-第N-1排学生： 需要引导型	第N排学生： 毕业需求型
<p>此类学生并不需要老师的教授，能够自我搞定。</p> <p>此类学生需要老师给他们平台和机会，让他们发挥能量。</p>	<p>此类学生对操作系统比较模糊，没有明确的方向感。 <b>理论上：</b>需要对操作系统进行深入学习 <b>实践上：</b>需要对操作系统进行动手锻炼</p> <p>此类学生需要教师引导他们对操作系统进行深入教学、掌握操作系统基本理论。 同时，在实践方面，鼓励学生参与操作系统实验的代码、开发等能力锻炼。 我们需要全力提高他们对操作系统的兴趣，从而推动基础软件人才的储备。</p>	<p>此类学生对操作系统的兴趣度较低，很大程度上是因为他们有其他的志向。</p> <p>此类学生需要老师激发他们掌握理解操作系统内容对其他课程的重要性，起到一个<b>纽带作用</b>。最终为其他学科的学习提供基础。</p>

# 教材和实验方案的选定

## 常见的操作系统教材

教材一：《操作系统精髓与设计原理》（2013-2014） 教材二：《操作系统概念》（2015-2018）

William Stallings

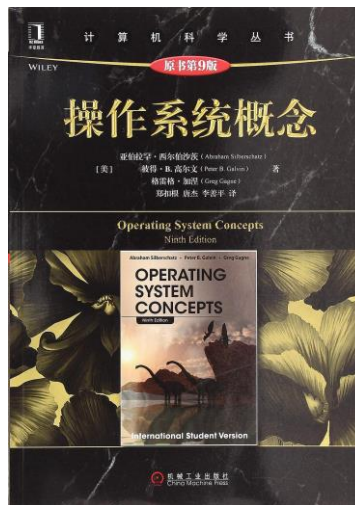


名称	修改日期
操作系统课件	2023/7/31 21:10
课件PPT	2020/11/26 15:54
[BOOK] Linux环境编程.pdf	2020/11/26 15:54
[BOOK] 自己动手写操作系统完全版.pdf	2020/11/26 15:54
0.pptx	2020/11/26 15:54
2014操作系统考题B卷.doc	2020/11/26 15:54
NWPU计算机学院OS课内实验指导一.pdf	2020/11/26 15:54
操作系统材料.rar	2020/11/26 15:54
操作系统课程计划.docx	2020/11/26 15:54
点名册.pdf	2020/11/26 15:54
教学大纲.docx	2020/11/26 15:54
教学日历.docx	2020/11/26 15:54
实验方案--表格版.docx	2020/11/26 15:54
实验指导.pdf	2020/11/26 15:54

### □ 特点：

- 清晰地解释了操作系统的基本概念
- 脉络清晰，循序渐进，极好地表现了知识点之间的承接关系

Abraham Silberschatz



名称	修改日期
ch0.pptx	2020/11/26 15:57
ch1-2.pptx	2020/11/26 15:57
ch3.pptx	2020/11/26 15:57
ch4.pptx	2020/11/26 15:57
ch5.pdf	2020/11/26 15:57
ch5.pptx	2020/11/26 15:57
ch6.pdf	2020/11/26 15:57
ch6.pptx	2020/11/26 15:57
ch7.pdf	2020/11/26 15:57
ch7.pptx	2020/11/26 15:57
ch8.pdf	2020/11/26 15:57
ch8.pptx	2020/11/26 15:57
ch9.pdf	2020/11/26 15:57
ch9.pptx	2020/11/26 15:57
ch10.pdf	2020/11/26 15:57
ch10.pptx	2020/11/26 15:57
ch11.pdf	2020/11/26 15:57
ch11.pptx	2020/11/26 15:57
ch12.pdf	2020/11/26 15:57
ch12.pptx	2020/11/26 15:57
ch13.pdf	2020/11/26 15:57
ch13.pptx	2020/11/26 15:57
ch14 - final.pptx	2020/11/26 15:57
课件.rar	2020/11/26 15:55

### □ 特点：

- 通俗易懂，叙述详细，例子丰富

# 教材和实验方案的选定

## 常见的操作系统教材

### 教材三：《现代操作系统》（2018-至今）

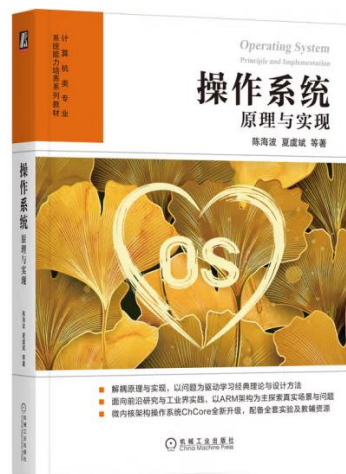
AnderwS.Tanenbaum



名称	修改日期
ucore_lab	2023/7/21 8:45
2018 ASPLOS - Making Huge Pages ...	2020/11/26 15:53
Memory2_4 - Ashish Panwar.pdf	2020/11/26 15:53
Modern OS_1.ppt	2021/3/3 7:20
Modern OS_2 - 习题课.ppt	2021/2/24 9:01
Modern OS_2 (来自DESKTOP-8BMVK...	2021/2/24 9:01
Modern OS_2.ppt	2021/2/23 21:54
Modern OS_3.ppt	2021/3/16 17:43
Modern OS_3_updated.ppt	2023/7/20 14:35
Modern OS_4.ppt	2020/11/26 15:54
Modern OS_5.ppt	2020/11/26 15:54
Modern OS_6.ppt	2021/6/17 11:43
ucore_lab.zip	2021/3/3 7:32
复习提纲.ppt	2022/6/15 8:40
实验0课件.pptx	2021/3/2 22:18

### 教材四：《操作系统：原理与实现》-2023-至今

陈海波、夏虞斌



名称	修改日期
Linux内核完全注释：基于0.11内核.pdf	2022/5/11 7:38
oslab.zip	2022/2/22 16:01
深入理解Linux内核(第三版)中文版.pdf	2022/5/11 7:38
实验1：熟悉实验环境.pdf	2022/2/22 16:00
实验2.zip	2022/5/11 7:39
实验2：操作系统的引导.pdf	2022/5/11 7:37
实验3：系统调用.pdf	2022/5/11 7:37
实验4,5：复盘.mp4	2022/5/11 7:39
实验4：进程运行轨迹的跟踪和统计.pdf	2022/5/11 7:40
实验4：资料文件.zip	2022/5/11 7:39
实验5：基于内核栈切换的进程切换.pdf	2022/5/11 7:40
实验6：信号量的实现和应用.pdf	2022/5/11 7:38
实验报告模板.doc	2022/5/11 7:37

#### □ 特点：

- 强调计算机领域中没有过时的思想
- 涉猎的内容时间跨度大

#### □ 特点：

- 结合代码进行讲解，相关概念不再是空中楼阁
- 关注到较新的操作系统技术

## 计划

### 9大模块：

1 操作系统基础知识

2 操作系统运行环境和机制

3 进程/线程模型

4 进程/线程调度

5 进程同步机制

6 死锁

7 内存管理

8 文件系统

9 设备管理





## 方法:

- 1 **逻辑思路**驱动兴趣：通过构建逻辑思路，一步步打造完整的系统；
- 2 **编程**驱动兴趣：通过设计小的编程题目，让学生感受操作系统的魅力；
- 3 **统计的方法**驱动兴趣：通过对一些问题的统计，带动学生积极思考。

**逻辑思路**驱动兴趣：通过构建逻辑思路，一步步打造完整的系统；

物理内存  
管理

导入

页表设计

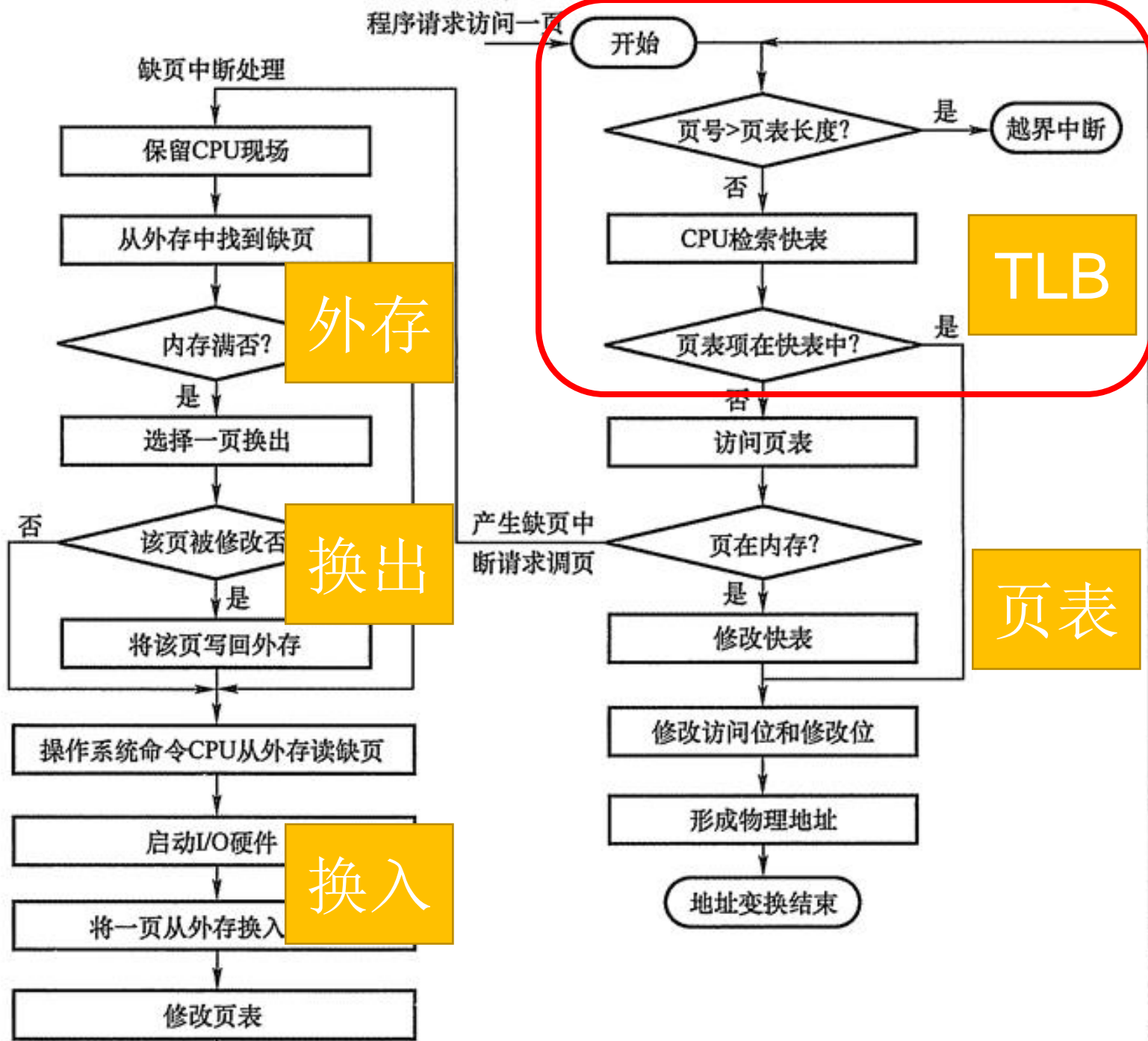
虚拟地址  
到物理地  
址转换

多级页表  
解决页表  
大小问题

TLB加速  
页表换页  
过程

缺页中断  
过程查找  
内存

整体系统



外存

换出

换入

TLB

页表

产生缺页中断请求调页

**编程驱动兴趣**：通过设计小的编程题目，让学生感受操作系统的魅力；

增加一个系统调用  
解读PCB结构体  
编写fork系统调用程序  
编写基于pthread的线程代码  
用Pthread库编写多线程互斥和同步的小程序

解读页表结构体  
编写程序实现bitmap和Buddy系统

解读inode结构体  
写一个程序遍历文件系统所有文件

**设计的原则**：短小，一般能够在1个小时左右完成的小练习。针对课堂的关键知识点展开练习。

**统计的方法驱动兴趣（抖包袱）**：通过对一些问题的统计，带动学生积极思考；

下面统计一下：有多少同学对自己的手机不满意，分别是什么方面不满意？

下面统计一下：你的手机内存多大，够用吗？你今天学了操作系统，你打算怎么解决这个问题？

下面统计一下：你平时开启多少应用程序？你觉得这些程序操作系统怎么管理？

下面统计一下：你的手机存储空间有多大，够用不？不够用怎么办？

下面统计一下：有哪些同学认为磁盘是顺序访问，还是随机访问设备？为什么？

## 预期效果:

- 1 **逻辑思路**驱动兴趣: 能够将一系列离散的技术串联, 提高系统设计能力;
- 2 **编程**驱动兴趣: 能够使用操作系统, 激发学生的学习热情;
- 3 **统计的方法**驱动兴趣: 让大家在笑声中理解技术的力量, 上课不走神;

**达成目标:** 在理论学习中, 驱动学生的思考和参与程度

# 教材和实验方案的选定

## 常见的操作系统实验

### 案例一：清华大学的uCore实验（2020-2021）

**xv6**  
a simple, Unix-like teaching operating system

Russ Cox  
Frans Kaashoek  
Robert Morris

xv6-book@pdos.csail.mit.edu

Draft as of September 4, 2018

#### □ xv6的缺点：

- 缺乏保护模式页机制
- 缺乏虚拟内存管理机制

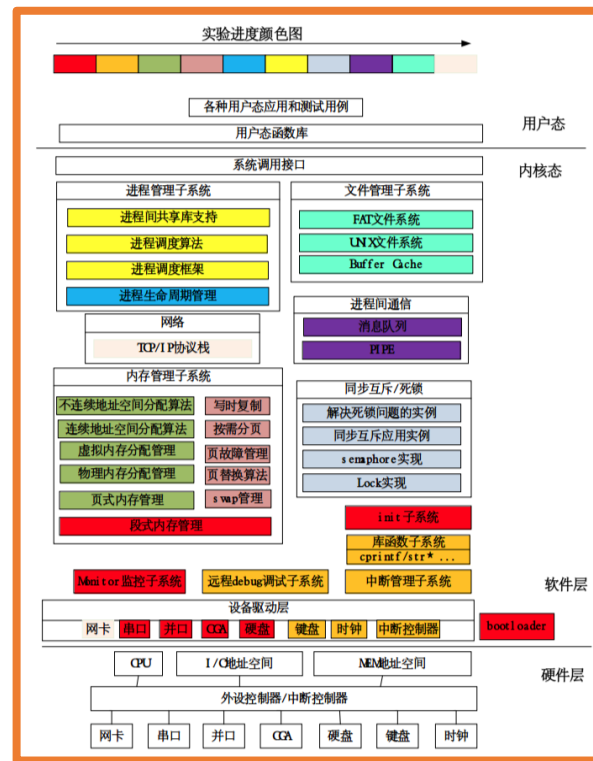
#### □ JOS的缺点：

- 不是传统UNIX内核架构

#### □ Linux的缺点：

- 代码量过大，过于复杂

==



2006年，MIT的Frans Kaashoek等开发了xv6操作系统并将其用于教学。

2008年，清华大学的陈渝老师鼓励其硕士研究生仿照xv6开发小型OS。

uCore实验引导学生完成一个从“空空如也”到“五脏俱全”的小型操作系统。

# 教材和实验方案的选定

常见的操作系统实验的实践分析

## 案例一：清华大学的uCore实验 (2020-2021)

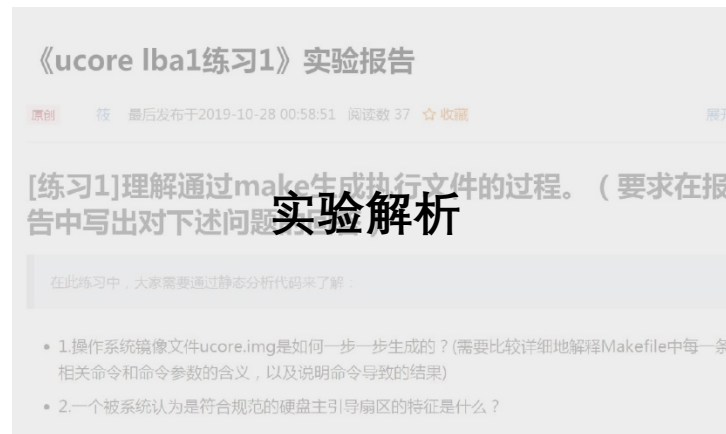


```
tmu.h (-/ucore_lab/labcodes/lab1/kern/mm) - VIM
#define STS_IG      0x5      // Task Gate / Coum Transmissions
#define STS_IG16   0x6      // 16-bit Interrupt Gate
#define STS_TG16   0x7      // 16-bit Trap Gate
#define STS_T32A   0x9      // Available 32-bit TSS
#define STS_T32B   0xB      // Busy 32-bit TSS
#define STS_CG32   0xC      // 32-bit Call Gate
#define STS_IG32   0xE      // 32-bit Interrupt Gate
#define STS_TG32   0xF      // 32-bit Trap Gate

/* Gate descriptors for interrupts and traps */
struct gatedesc {
    unsigned gd_off_15_0 : 16; // (2 bytes) low 16 bits of offset i
    n segment
    unsigned gd_ss : 1; // segment selector
    unsigned gd_args : 1; // args, 0 for interrupt/trap gate

    unsigned gd_rsv1 : 3; // reserved (should be zero I guess)
    unsigned gd_type : 4; // type (STS_{TG,IG32,TG32})
    unsigned gd_s : 1; // must be 0 (system)
    unsigned gd_dpl : 2; // descriptor (meaning new) privilege
    level
    unsigned gd_p : 1; // Present
    unsigned gd_off_31_16 : 16; // (2 bytes) high bits of offset i
    n segment
};
```

视频讲解



实验解析

### □ 特点:

- 基于基本框架，往上添砖加瓦
- 代码量适中

### □ 缺点:

- 难度偏高，学生难以独立完成
- 部分学生反映配环境困难

适合挑战性学生



## 常见的操作系统实验

### 案例二：哈尔滨工业大学的HIT-OSLAB实验（2022）

- 学生：我们要成为掌握计算机关键技术  
的工程师

绝知此事要躬行！



#### □ Windows的缺点：

- 内核代码不开源
- 相关文档资料较少

#### □ 最新的Linux的缺点：

- 代码量过大，过于复杂

=



哈尔滨工业大学的李治军老师强调在可实际运行的操作系统上的实践。

需要考虑到简单性、参考资料易获取和将来可以进一步挖掘。

选择在Linux 0.11上完成操作系统课程实验。

## 常见的操作系统实验的实践分析

### 案例二：哈尔滨工业大学的HIT-OSLAB实验（2022）

#### 实验A内容：

- 实验 1 熟悉实验环境
- 实验 2 操作系统的引导
- 实验 3 系统调用
- 实验 4 进程运行轨迹的跟踪与统计
- 实验 5 基于内核栈切换的进程切换
- 实验 6 信号量的实现和应用
- 实验 7 地址映射与共享
- 实验 8 终端设备的控制
- 实验 9 proc文件系统的实现

#### 实验B内容：

- 实验 1 启动操作系统
- 实验 2 物理内存管理
- 实验 3 虚拟内存管理
- 实验 4 内核线程
- 实验 5 用户进程
- 实验 6 处理器调度
- 实验 7 同步互斥
- 实验 8 文件系统

区分难易的实验内容

#### □ 特点：

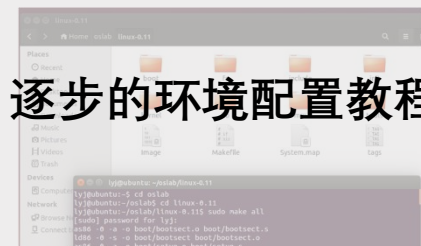
- 既有深度，又有广度
- 代码量适中

#### □ 缺点：

- 难度适中，对于需要挑战的同学缺乏兴趣

#### 编译内核

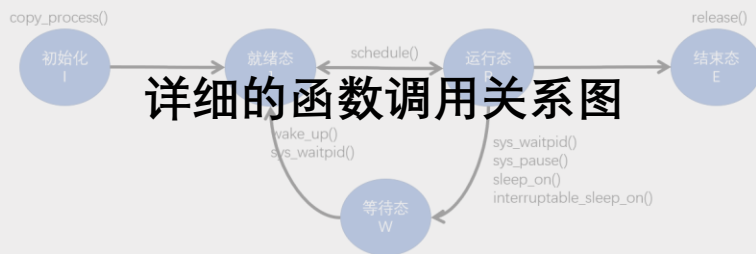
实验环境安装成功后，进入linux-0.11源代码文件夹，可以看到其中的Makefile文件，使用make all命令即可完成内核的编译。如果出现错误可以使用sudo字符使用管理者权限。



#### 逐步的环境配置教程

#### 实验原理

linux中进程的状态转换



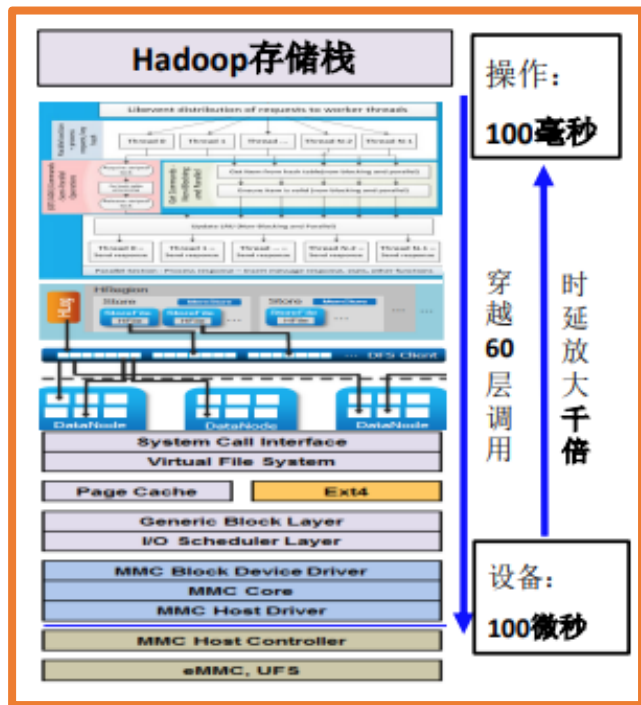
#### 详细的函数调用关系图

适合感兴趣学生

# 教材和实验方案的选定

常见的操作系统实验

## 案例三：上海交通大学的ChCore实验（2023）



### Linux代码庞杂

- Linux代码已超3000万行，仍以每年200万行在增加/更新
- Android涉及Linux代码不超过300万行
- 苹果自研iOS内核的代码在200万行左右

很多场景追随Linux的演进代价太大，生命周期管理困难。

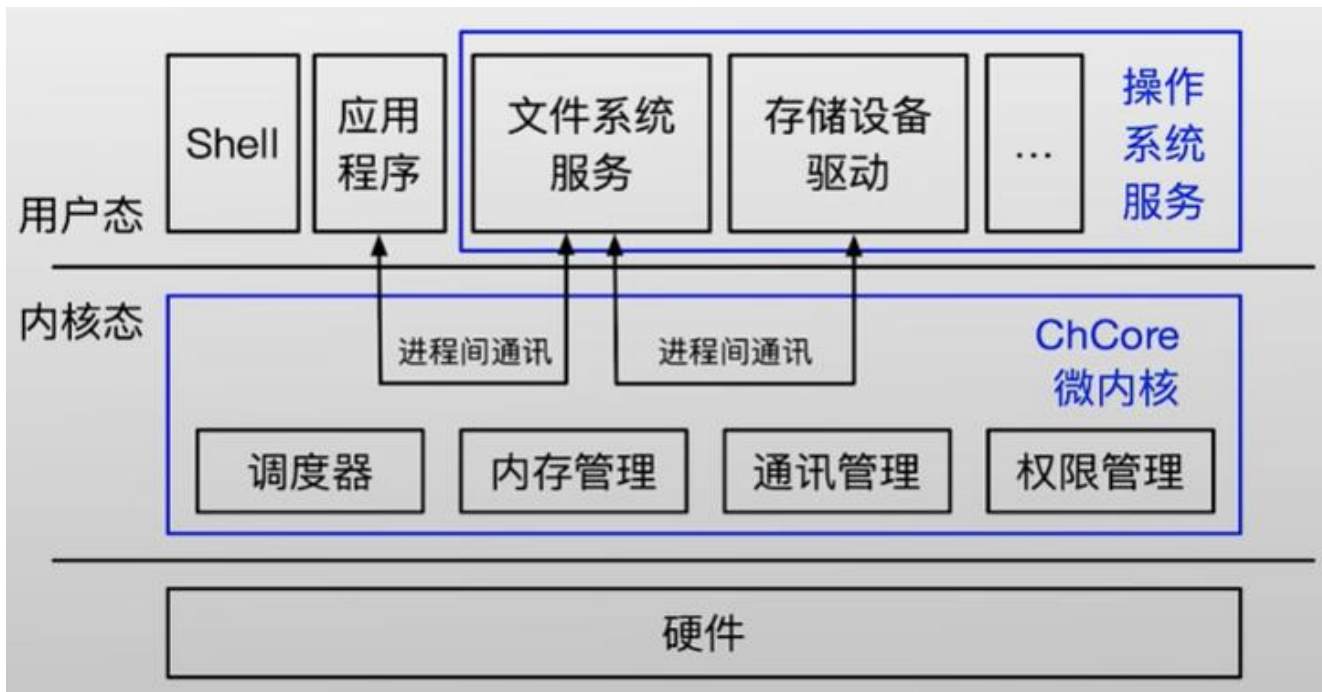


新型硬件发展需要新的操作系统抽象 传统开源模式在特定场景开始走向窘境 智能家庭等新场景需要新的操作系统  
与设计来充分释放算力

# 教材和实验方案的选定

常见的操作系统实验

## 案例三：上海交通大学的ChCore实验（2023）



构建微内核操作系统ChCore，内核仅负责必要的功能：  
启动，内存管理，进程管理，异常处理，IPC等



对 ChCore 代码进行了大量裁剪与简化，并添加了一些练习，从而形成 ChCore 课程实验 (ChCore Labs)

# 教材和实验方案的选定

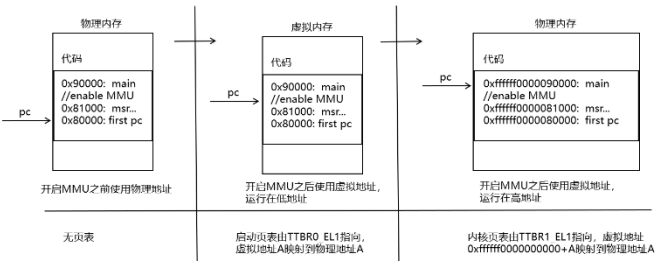
## 常见的操作系统实验的实践分析

### 案例三：上海交通大学的ChCore实验（2023）

基于ChCore Lab v2的实验指导书，删除部分练习，增加了更多提示，下调了实验难度。

#### 内核的启动

- 启动内存管理单元(Memory Management Unit, MMU)
  - 在内核的启动阶段，还需要配置启动页表，并启用 MMU。
  - 为操作系统提供了一种实现虚拟内存管理的机制，使得操作系统可以更好地管理进程的地址空间。



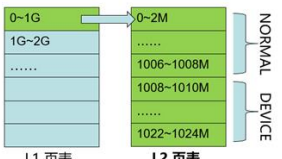
无页表 | 启动页表由TTBR0 EL1指向，虚拟地址A映射到物理地址A | 内核页表由TTBR1 EL1指向，虚拟地址0xfffff000000000+A映射到物理地址A

#### PART 1: 配置内核启动页表

- 建立低地址空间的内核启动页表

```
/* Normal memory: PHYSMEM_START - PERIPHERAL_BASE */
for (; vaddr < PERIPHERAL_BASE; vaddr += SIZE_2M) {
    boot_ttbr0_l2[GET_L2_INDEX(vaddr)] =
        (vaddr) /* low mem, va = pa */
        | UXN /* Unprivileged execute never */
        | ACCESSED /* Set access flag */
        | NG /* Mark as not global */
        | INNER_SHARABLE /* Shareability */
        | NORMAL_MEMORY /* Normal memory */
        | IS_VALID;
}

/* Peripheral memory: PERIPHERAL_BASE - PHYSMEM_END */
for (vaddr = PERIPHERAL_BASE; vaddr < PHYSMEM_END; vaddr += SIZE_2M) {
    boot_ttbr0_l2[GET_L2_INDEX(vaddr)] =
        (vaddr) /* low mem, va = pa */
        | UXN /* Unprivileged execute never */
        | ACCESSED /* Set access flag */
        | NG /* Mark as not global */
        | DEVICE_MEMORY /* Device memory */
        | IS_VALID;
}
```



#### 特点:

- 从“0”开始构建整个框架
- 内核小而简，并将部分特性移入用户态，体现微内核的特点

#### 缺点: 难度偏高


- 代码量较大

#### 用户进程和线程

```
void main(vaddr_t boot_flag)
{
    uart_init();
    ms_init();
    arch_interrupt_init();
    create_root_thread();
    ret_to_thread(switch_context());
}

void create_root_thread(void)
{
    struct cap_group *root_cap_group;
    int thread_cap;
    struct thread *root_thread;

    ① root_cap_group = create_root_cap_group(ROOT_NAME, strlen(ROOT_NAME));
    ② thread_cap = __create_root_thread(root_cap_group, &root_thread);
    root_thread = obj_get(root_cap_group, thread_cap, TYPE_THREAD);
    obj_put(root_thread);
    ③ switch_to_thread(root_thread);
}
```

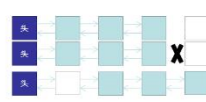


#### 2-1 协作式调度

- Round Robin调度策略
  - 每个CPU维护一个FIFO（先入先出）的线程就绪队列，所有可以执行的线程均放在该就绪队列中。调度器将每次将当前正在执行的线程放回到就绪队列中，并从该就绪队列中按照先入先出的顺序取一个就绪的线程执行

```
struct queue meta_rr_ready_queue_meta[PLAT_CPU_NUM];
struct thread idle_threads[PLAT_CPU_NUM];
```

- 入队操作rr\_sched\_enqueue
- 出队操作rr\_sched\_dequeue
- 选择待执行的线程rr\_sched\_choose\_thread
- 执行调度rr\_sched



适合锻炼学生

## 实验模块：

1. 操作系统启动
2. 内存管理
3. 进程管理
4. 多核启动、调度、IPC
5. 文件系统与Shell

## 驱动方法：

**1 带学生入门：**系统启动的实验中，我们进行手把手教授，让学生真正能够上手操作系统启动流程；

## 2 大实验部分：

**2.1 实验讲解：**讲解原理和代码，让学生能够掌握实验的基本原理；

**2.2 志愿者小组：**邀请学生作为志愿者，参与实验的设计与讲解环节，变被动为主动；

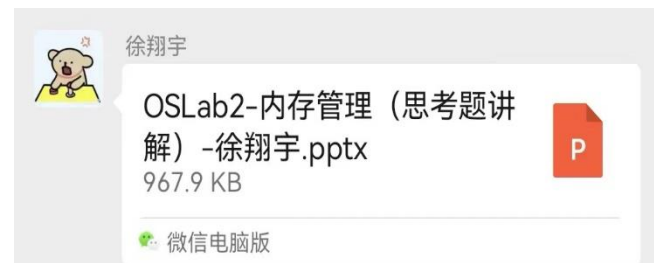
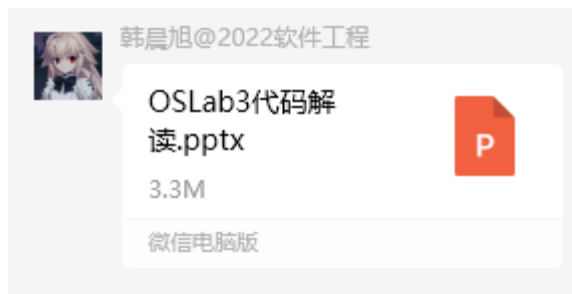
**2.3 分组实验：**构建讨论小组，大家能够互帮互助，共同学习；

**2.4 企业走入课堂：**激发学生前沿实验思考！

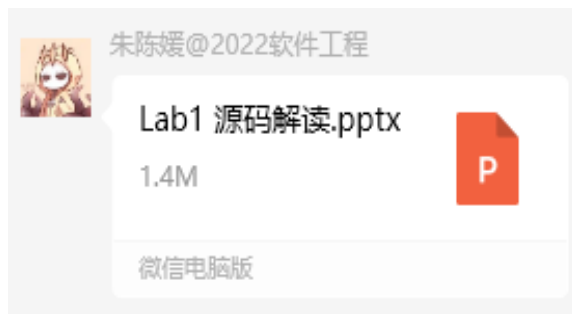
# 实验教学的兴趣驱动

在实验环节，让部分学生主动担任实验解读志愿者。

提前动员安排，通过助教组进行提前的训练和培训，保障志愿者讲解的质量



10月30日 中午12:48



**达成目标：激发学生学习的兴趣，让他不再是学，而是教的角色转变！**

根据模块，在进程管理、内存管理、文件系统和设备管理四个部分，提供四篇与基于理论相关的前沿学术论文，分成小组进行阅读思考！

**论文选择标准：**计算机领域顶级会议中对经典理论的前沿热点优化，让学生充分感受操作系统基础理论的创新机会！

- 进程管理：进程锁机制的优化设计
- 内存管理：内存大页设计
- 文件系统：最新的文件系统设计
- 设备管理：SPDK的设备访问模式





- iwu Lo, Han-Ting Lin, Yao-Hung Hsieh, Chao-Ting Lin, Yu-Hsueh Fang, Ching-Shen Lin, Ching-Chun (Jim) Huang, Kam-Yiu Lam, Yuan-Hao Chang: RON: One-Way Circular Shortest Routing to Achieve Efficient and Bounded-waiting **Spinlocks**. *OSDI 2023*: 17-31
- Ashish Panwar, Sorav Bansal, K. Gopinath: HawkEye: Efficient Fine-grained OS Support for **Huge Pages**. *ASPLOS 2019*: 347-360
- David Domingo, Sudarsun Kannan: pFSCK: Accelerating **File System Checking and Repair** for Modern Storage. *FAST 2021*: 113-126
- Didona D, Pfefferle J, Ioannou N, et al. **Understanding modern storage APIs**: a systematic study of libaio, SPDK, and io\_uring[C]//*Proceedings of the 15th ACM International Conference on Systems and Storage*. *2022*: 120-127.

**达成目标：**本科期间就能对操作系统领域至少4篇论文完成解读，激发研究兴趣

在不同的模块构建不同的思政案例，激发学生的思考和学习热情：

- 1 进程管理：如何通过解决优先级反转问题，攻克手机卡顿的问题的案例
- 2 内存管理：如何通过交换技术解决手机进程杀死问题，让进程在内存中压缩
- 3 文件系统：如何通过文件系统的页高速缓存设计，实现性能的大幅改进？用AI解决文件系统问题
- 4 IO设备管理：给同学们介绍各种不同的IO处理模式如何成为黑科技？

而以上的每一项技术，真正落地到实处，就能成为解决用户问题的核心技术

**达成目标：**让学生知道操作系统的力量，构建知识的价值

作为一项理论和实践要求极高的本科生基础课程，同时也是国家“卡脖子”技术，操作系统的教学至关重要。我们通过兴趣驱动学生的学习，希望达成以下目标：

- 1 激发学生的学习兴趣，让操作系统变为一个轻松的课程
- 2 激发学生的实践兴趣，让学生真正的懂操作系统，写操作系统
- 3 激发学生的思考兴趣，站在巨人的肩膀上看得更远，思考的更深
- 4 激发学生的价值兴趣，感受操作系统的力量，为自主设计国产操作系统打下基础



2023 CCF CHINASOFT  
中国软件大会

感谢观看

