操作系统整体规划与设计

ZJUNIX

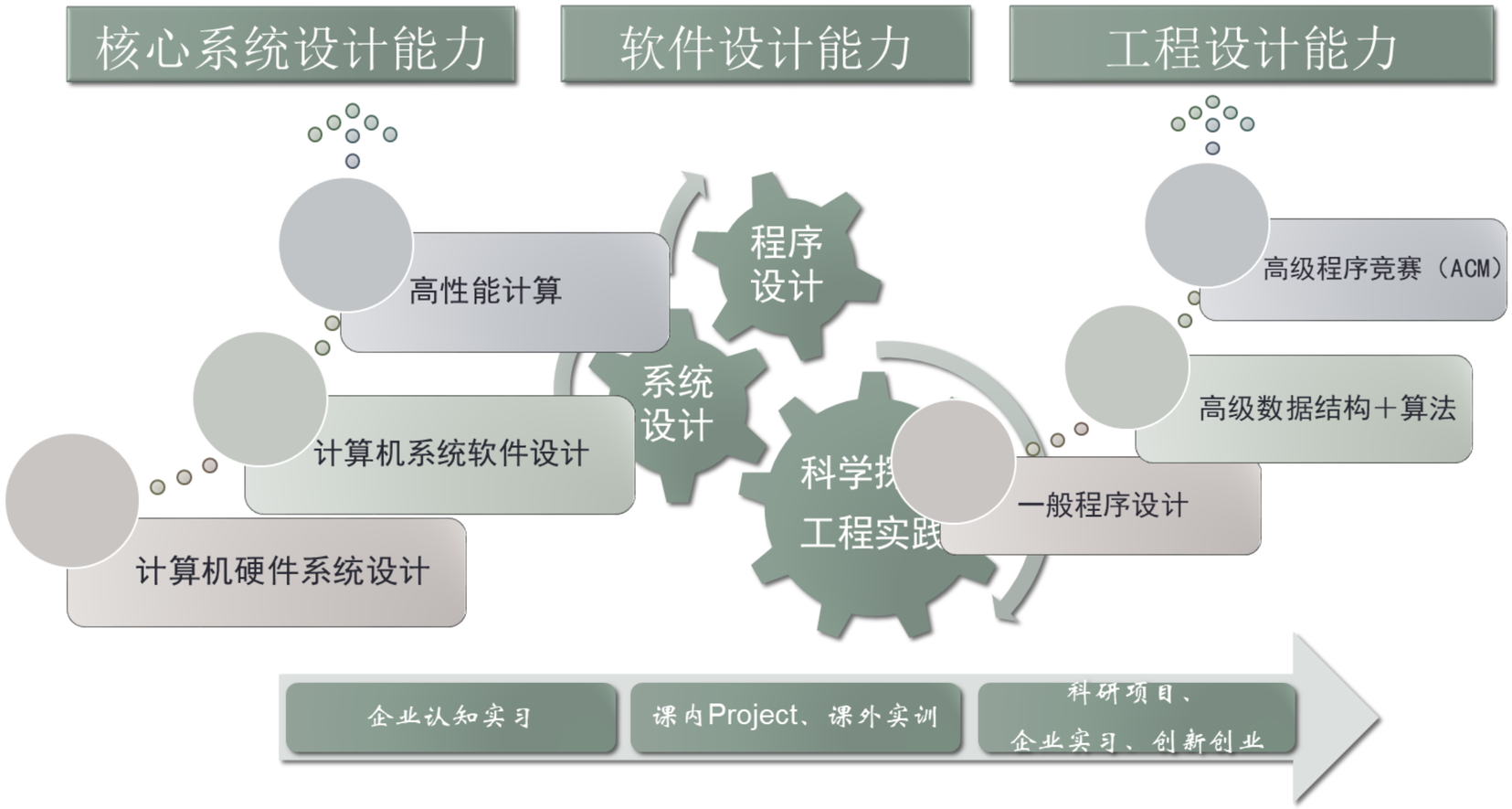
目录

系统能力贯通培养

- 系统能力的内涵
- 贯通教学课程体系

ZJUNIX

从提高能力着手，能力是多方面的



ZJUNIX

分级分层实施，突出能力训练

- 重组6门课
- 新增2门课
- 跨越5~6学期
- 分成两个层次，每个层次3个级别，互有衔接
- 覆盖不同层次的高校学生

系统能力培养核心课程系列

数字逻辑　　可根据教学要求分解或扩展实验

E01： 变量译码器　**E02：** 七段数码管译码器　**E03：** 多路选择器及测调环境
E04： 全加器设计　**E05：** 加减法器和 ALU　**E06：** 锁存器与触发器
E07： 状态机设计　**E08：** 寄存器与寄存器传输　**E09：** 计数/定时器设计与应用
E10： 移位寄存器　**E11：** 实验综合　▼**课程设计**

计算机组成　　可分解为课程实验和独立课程设计

- ▼多路选择器与 CPU 辅助模块设计
- ▼七段显示部件(设备)设计；
- ▼建立 CPU 测试和应用环境；
- ▼ALU、Regs 及 CPU IP 核集成
- ▼单周期 CPU 设计与集成替换
- ▼多周期 CPU 设计与集成替换
- ▼**课程设计：CPU** 最简单应用-**基本总线和 GPIO 设备设计实现**

计算机系统结构

- ▼多周期 CPU 回顾
- ▼基本五级流水线实现
- ▼stall 五级流水线实现
- ▼Fowarding 的五级流水线实现
- ▼控制竞争性能改进
- ▼流水线处理器指令扩展实现
- ▼流水线中断设计
- ▼Cache 设计实现
- ▼**课程设计：**流水线处理器集成 SOC

计算机硬件系统综合　　简易计算机系统实现最低要求

- ▼CPU 优化设计(任选一款组成实验设计的 CPU)
- ▼扩展简单总线设计
- ▼PS2 模块设计
- ▼VGA 模块设计
- ▼硬件集成
- ▼命令管理器设计(无盘 DOS/大监控)
- ▼**系统集成：软硬件集成**(含有意义的应用程序设计)

计算机系统综合实践—硬件篇：标准化硬件设计

三种模式四大阶段

- ▼ **开源 CPU 核模式**　▼ **自主 CPU 模式**　▼ **第三方硬核模式**
- ▼ **阶段一：**扩展重现前导课的流水线 MIPS CPU，50 条指令
- ▼ **阶段二：**系统总线标准化。替换为标准 Wishbone 总线
- ▼ **阶段三：扩展** CP0 功能，在硬件层支持 OS 的各项功能。
- ▼ **阶段四：**添加**外存储**和文件系统构建。

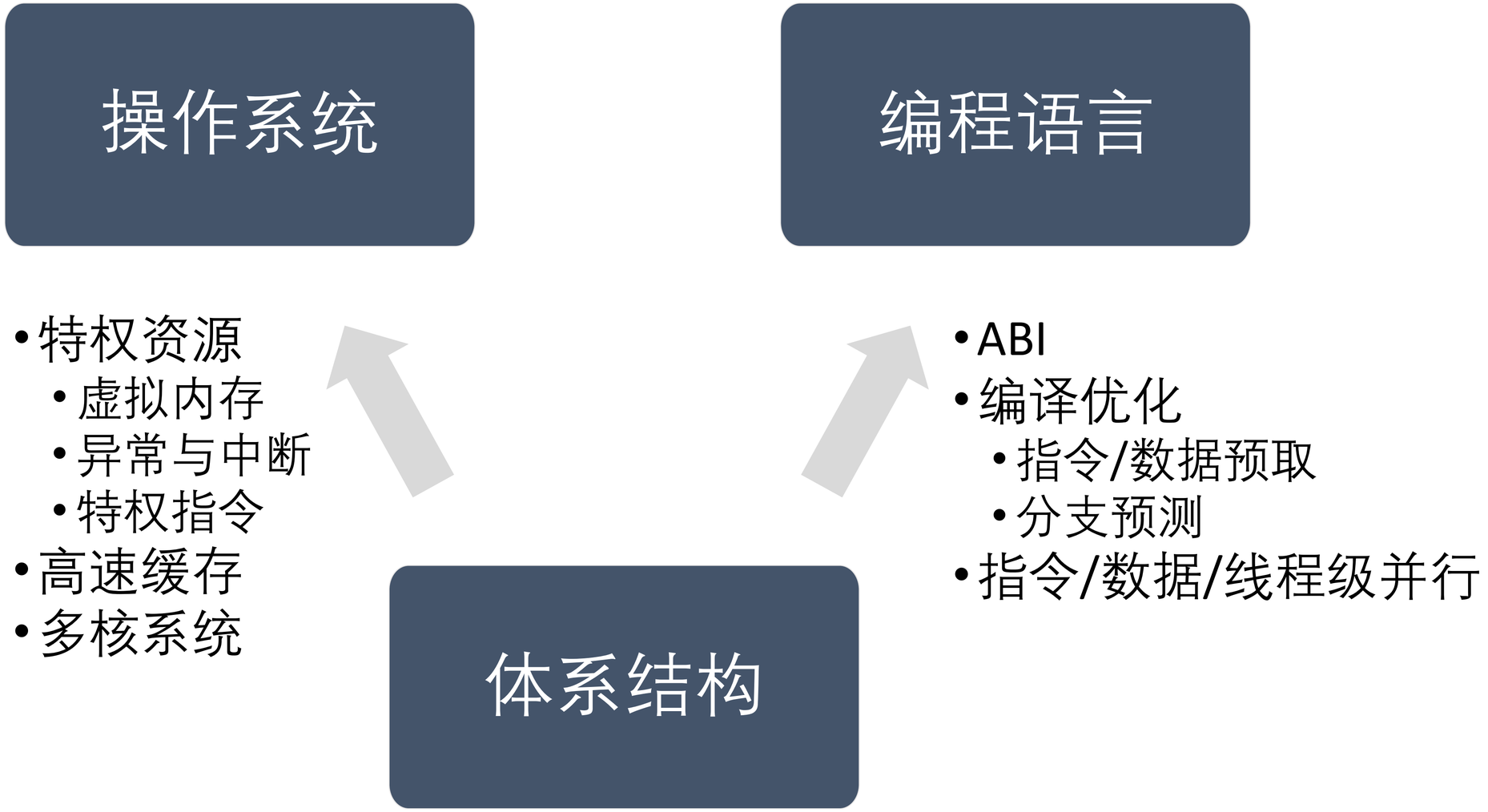
计算机系统综合实践—软件篇：操作系统设计

十大实验实现 OS 内核

- ▼ 实验环境搭建：**mips-gcc** 交叉编译器、模拟器

Exp01： 系统管理策略规划　**Exp02：** 启动与系统初始框
Exp03： 实现 **vga** 显示功能　**Exp04：** 实现键盘输入功能
Exp05： 添加时钟中断　**Exp06：** 添加内存管理
Exp07： 实现进程管理　**Exp08：** 实现文件系统
Exp09： 实现库、**shell** 程序　**Exp10：** 完善 **bootloader** 与系统

系统能力贯通培养


操作系统
编程语言
体系结构
• 特权资源
• 虚拟内存
• 异常与中断
• 特权指令
• 高速缓存
• 多核系统
• ABI
• 编译优化
• 指令/数据预取
• 分支预测
• 指令/数据/线程级并行

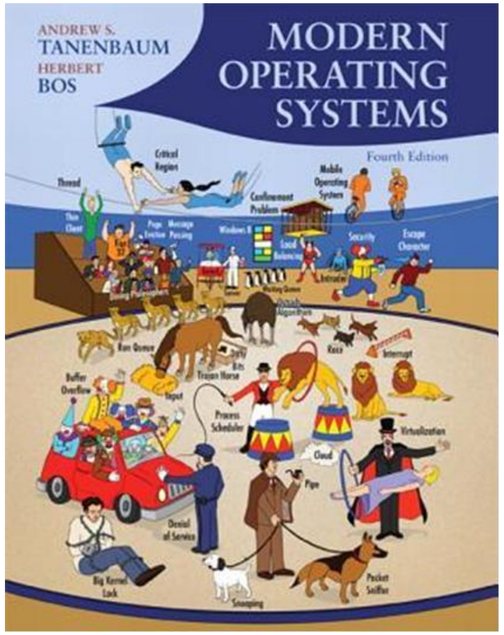
操作系统规划设计

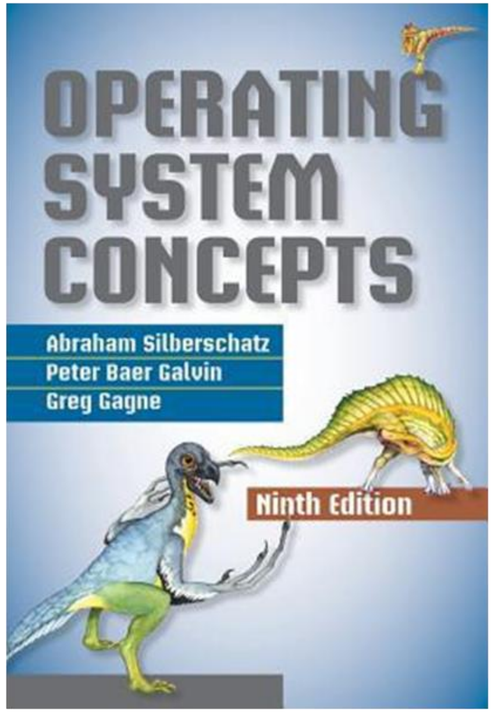
- 规划设计
- 内存管理
- 进程管理
- 文件系统
- 设备驱动
- 应用程序

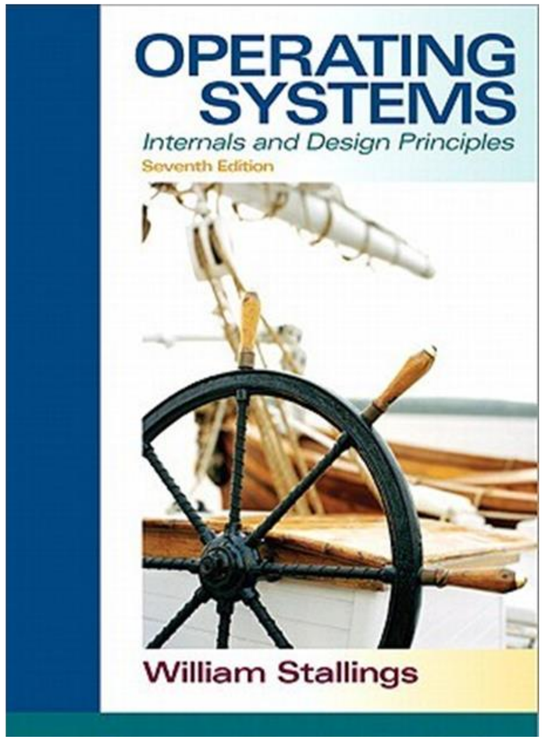
面向系统能力的操作系统课程设计

- 现有OS课程参考/教材书籍
  - 操作系统原理类
    - 从理论上介绍OS设计与实现所涉及的技术原理
  - 内核剖析类
    - 从代码角度分析OS主要模块的设计与实现
  - 动手实践类
    - 从零开始实现一个小型内核
- ZJUNIX
  - 动手实践类
  - 简化硬件平台, 衍生于逻辑、计组、SoC课程

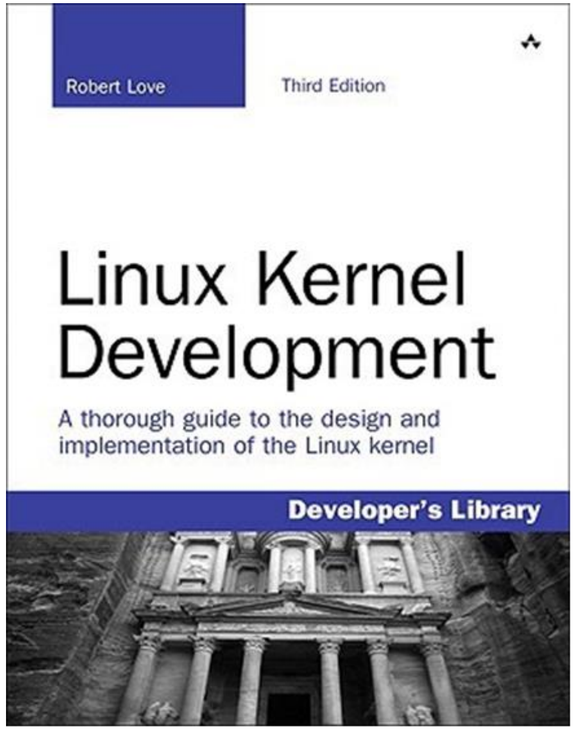
OS原理类

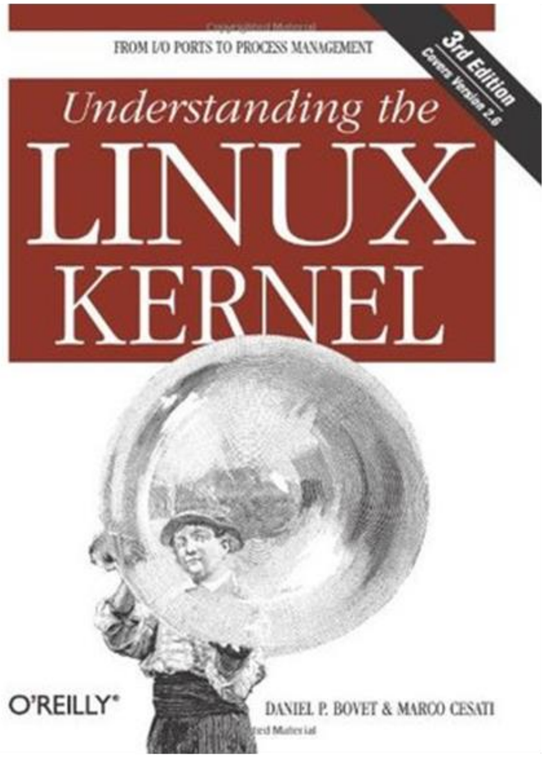

ANDREW S.
TANENBAUM
HERBERT
BOS
MODERN
OPERATING
SYSTEMS
Fourth Edition

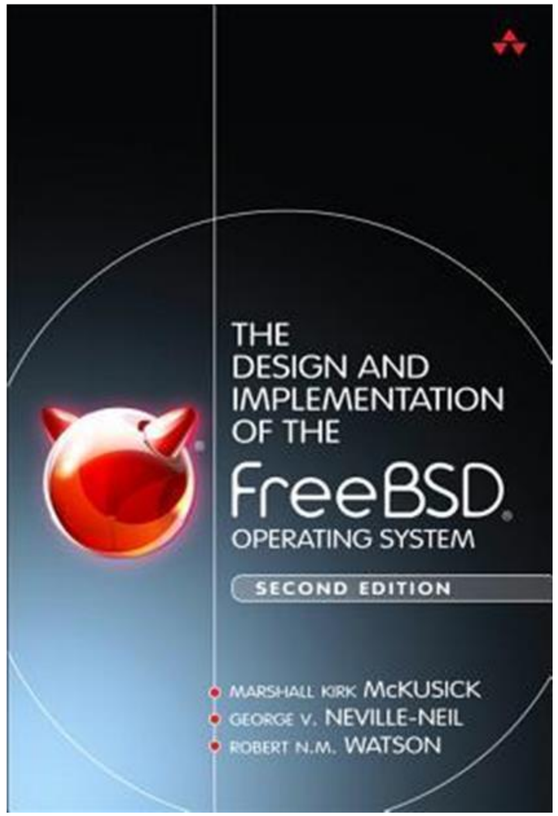

OPERATING
SYSTEM
CONCEPTS
Abraham Silberschatz
Peter Baer Galvin
Greg Gagne
Ninth Edition


OPERATING
SYSTEMS
Internals and Design Principles
Seventh Edition
William Stallings

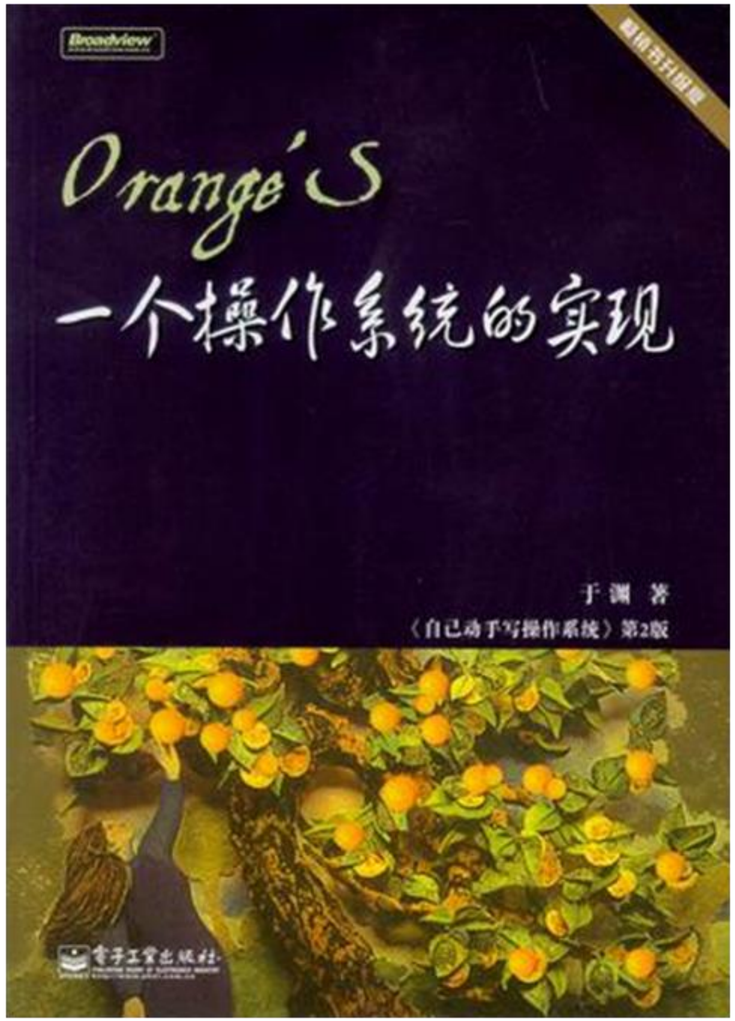
内核剖析类

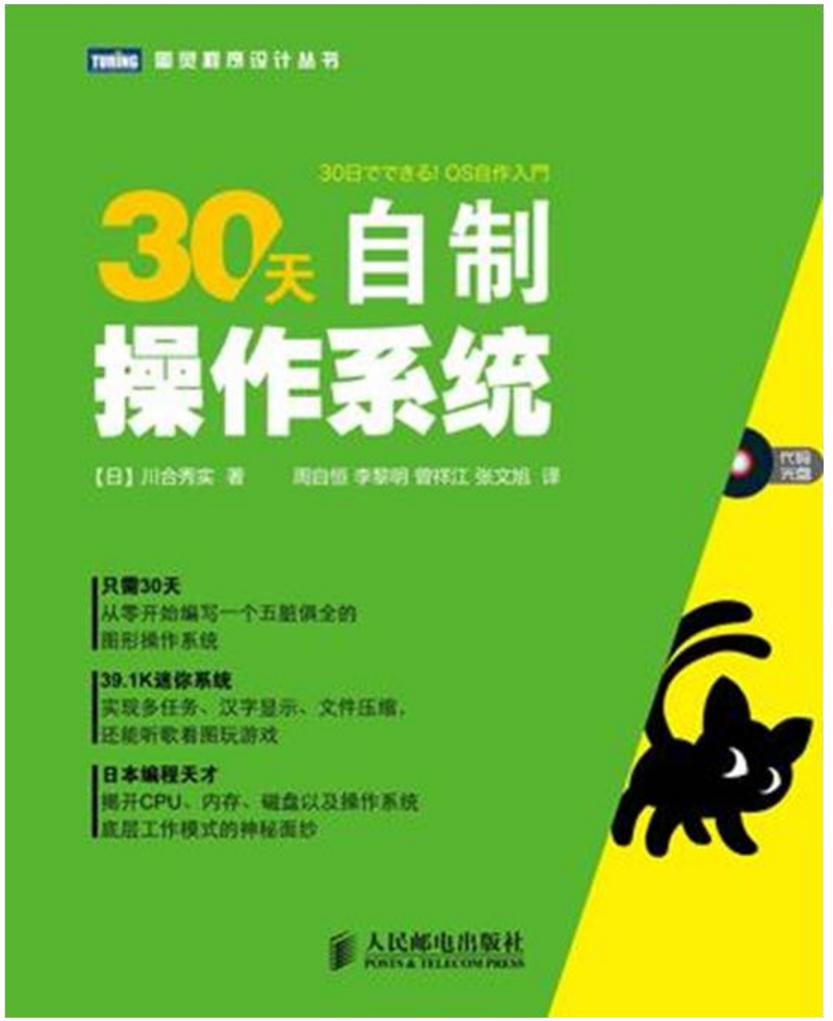

Robert Love
Third Edition
Linux Kernel Development
A thorough guide to the design and implementation of the Linux kernel
Developer's Library


FROM I/O PORTS TO PROCESS MANAGEMENT
3rd Edition
Covers Version 2.6
Understanding the
LINUX
KERNEL
O'REILLY
DANIEL P. BOVET & MARCO CESATI


THE
DESIGN AND
IMPLEMENTATION
OF THE
FreeBSD
OPERATING SYSTEM
SECOND EDITION
MARSHALL KIRK McKUSICK
GEORGE V. NEVILLE-NEIL
ROBERT N.M. WATSON

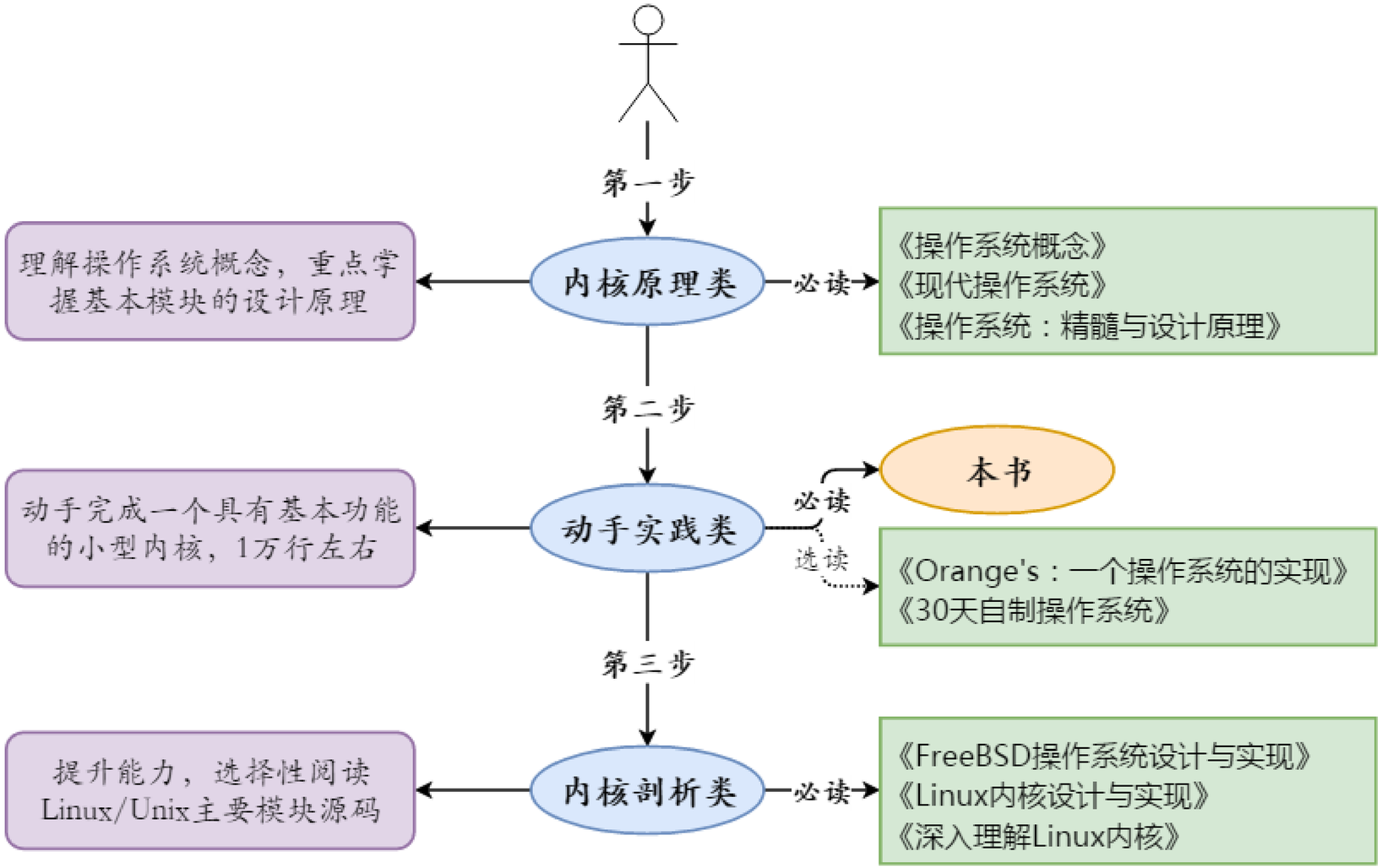
动手实践类


Broadview
Orange'S
一个操作系统的实现
于渊 著
《自己动手写操作系统》第2版


TURING
30日でできる! OS自作入門
30天自制操作系统
【日】川合秀实 著
周自恒 李黎明 曾祥江 张文旭 译
只需30天
从零开始编写一个五脏俱全的
图形操作系统
39.1K迷你系统
实现多任务、汉字显示、文件压缩,
还能听歌看图玩游戏
日本编程天才
揭开CPU、内存、磁盘以及操作系统
底层工作模式的神秘面纱
人民邮电出版社

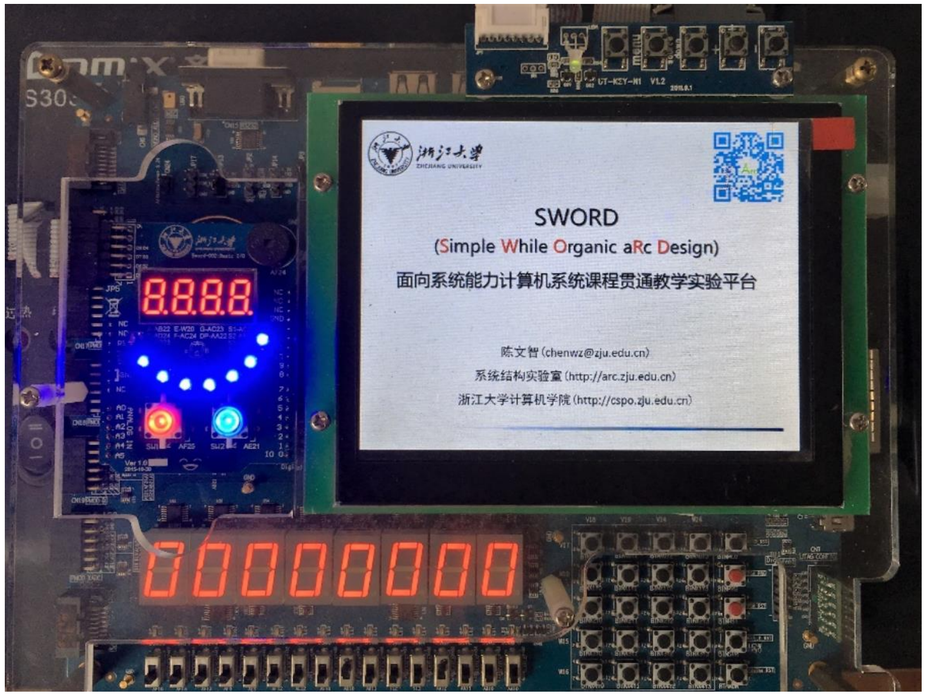

“基于系统能力培养的计算机专业课程建设研究”项目规划教材
操作系统
设计与实现
Operating System
Design and
Implementation
陈文智 主编
陈文智 施青松 龙鹏 编著
高等教育出版社

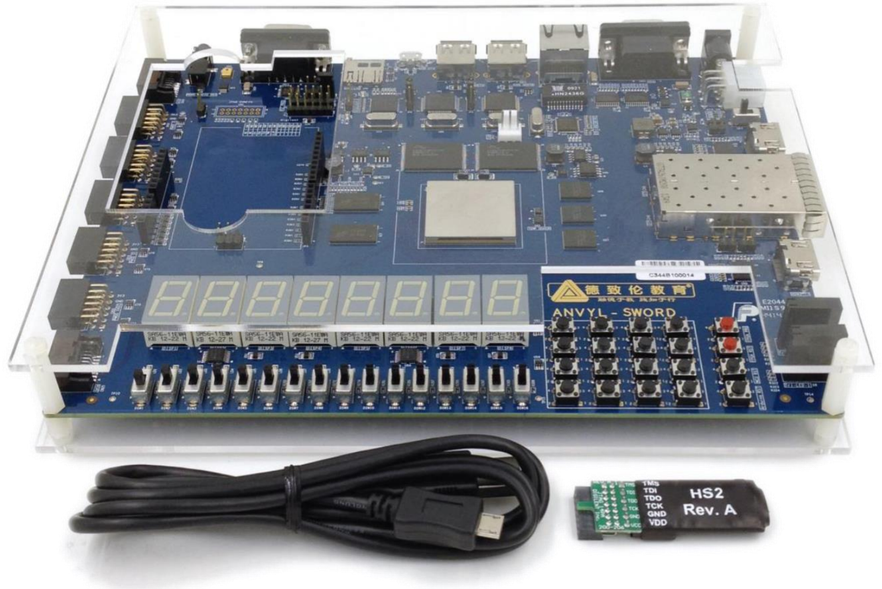
建议的OS学习路线


第一步
内核原理类
理解操作系统概念，重点掌握基本模块的设计原理
必读
《操作系统概念》
《现代操作系统》
《操作系统：精髓与设计原理》
第二步
动手实践类
动手完成一个具有基本功能的小型内核，1万行左右
必读
本书
选读
《Orange's：一个操作系统的实现》
《30天自制操作系统》
第三步
内核剖析类
提升能力，选择性阅读Linux/Unix主要模块源码
必读
《FreeBSD操作系统设计与实现》
《Linux内核设计与实现》
《深入理解Linux内核》

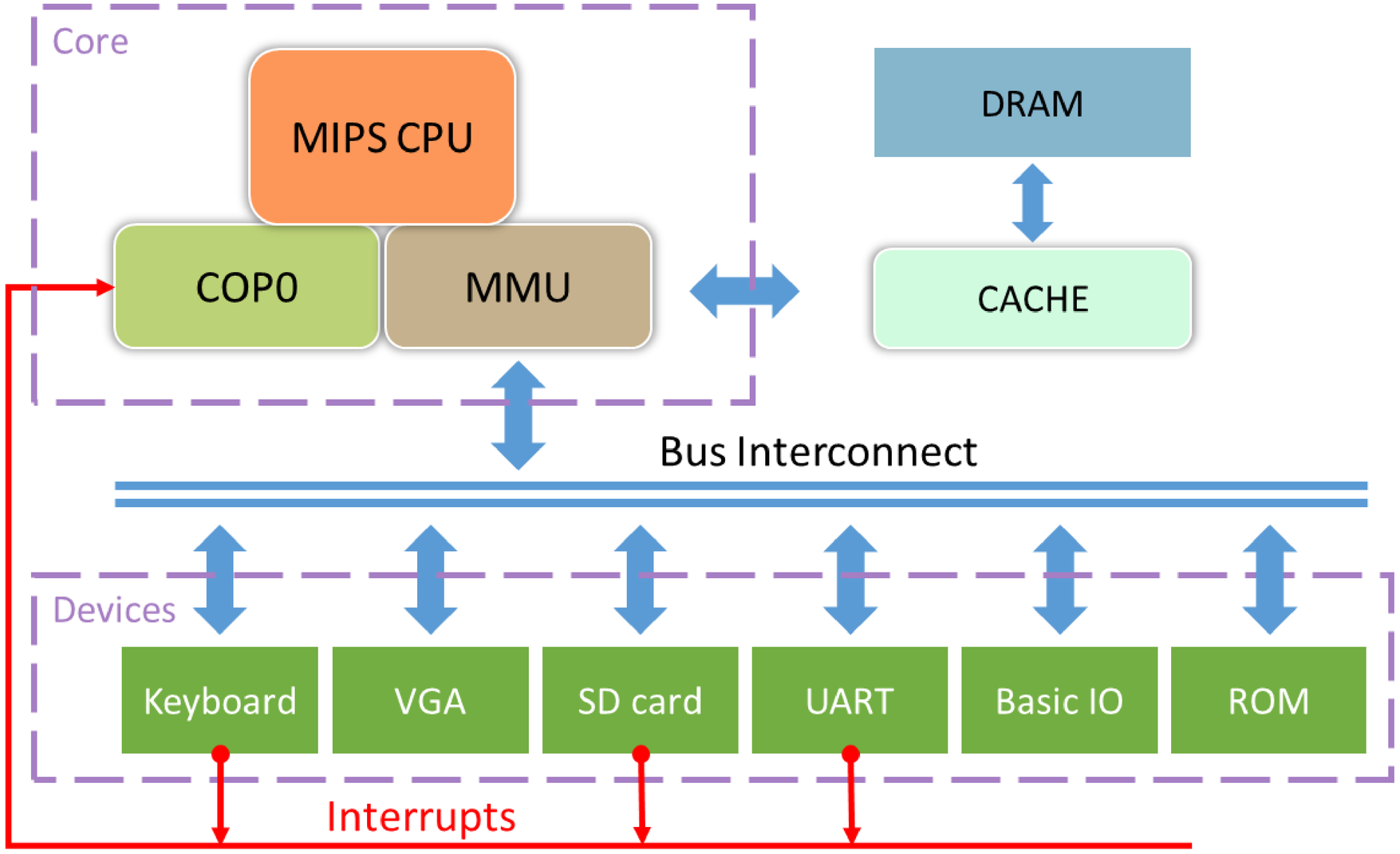
SWORD实验平台

- SWORD 是一款系统能力培养贯通教学实验平台
- 核心芯片为Xilinx生产的Kintex-7系列FPGA
  - Kintex™-7 XC7K325T-1FFG676





系统架构


Core
MIPS CPU
COP0
MMU
DRAM
CACHE
Bus Interconnect
Devices
Keyboard
VGA
SD card
UART
Basic IO
ROM
Interrupts

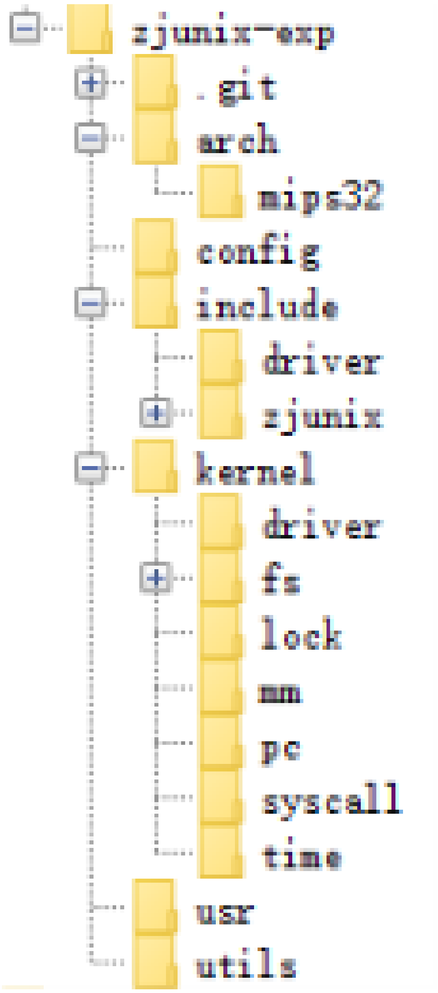
指令列表

指令类型(共87)	指令列表
算术运算(22)	add, addi, addiu, addu, sub, subu, slt, slti, sltiu, sltu, mul, mult, multu, madd, maddu, msub, msubu, div, divu, clo, clz, lui
逻辑运算(13)	and, andi, or, ori, xor, xori, nor, sll, sllv, srl, srlv, sra, srav
内存访问(12)	lb, lbu, lh, lhu, lwl, lwr, lw, sb, sh, swl, swr, sw
跳转(12)	beq, bne, blez, bgez, bltz, bgtz, bltzal, bgezal, j, jal, jr, jalr
移动(6)	movz, movn, mfhi, mthi, mflo, mtlo
陷入(12)	tge, tgei, tgeiu, tgeu, tlt, tlti, tltiu, tltu, teq, teqi, tne, tnei
COP0(7)	mfc0, mtc0, tlbr, tlbwi, tlbwr, tlbp, eret
其他(3)	syscall, break, cache

系统架构升级说明

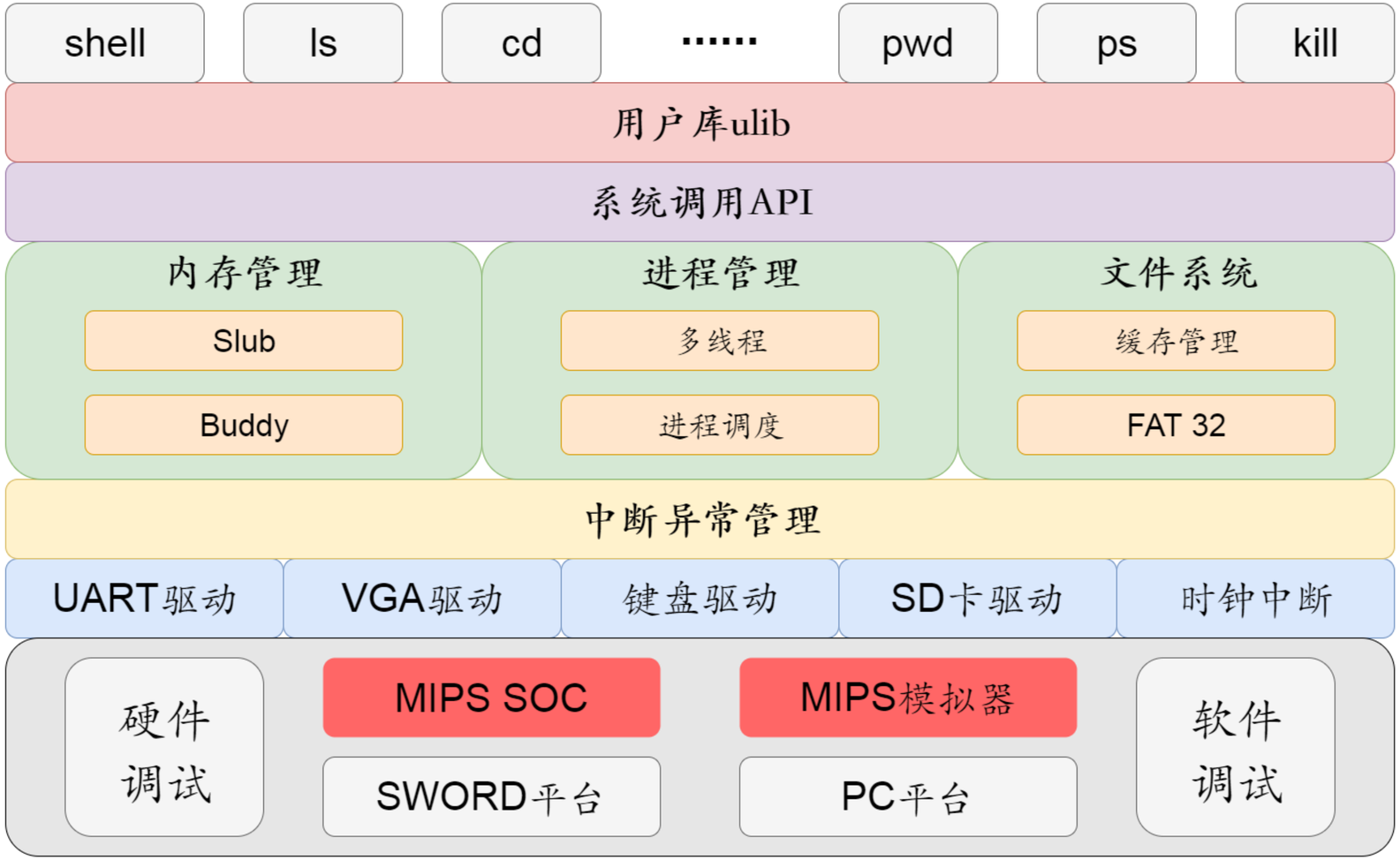
- 升级目的:
  - 尽可能与MIPS32规范兼容
  - 充分利用现有MIPS工具链和软件资源
- COP0(协处理器0):
  - 17个COP0寄存器，全部与MIPS32规范兼容
- 虚拟地址空间映射:
  - 与MIPS32规范兼容的TLB MMU设计
  - 与MIPS32规范兼容的虚拟地址划分
- 高速缓存(Cache):
  - 修改了强制写回方式，提升性能

代码结构

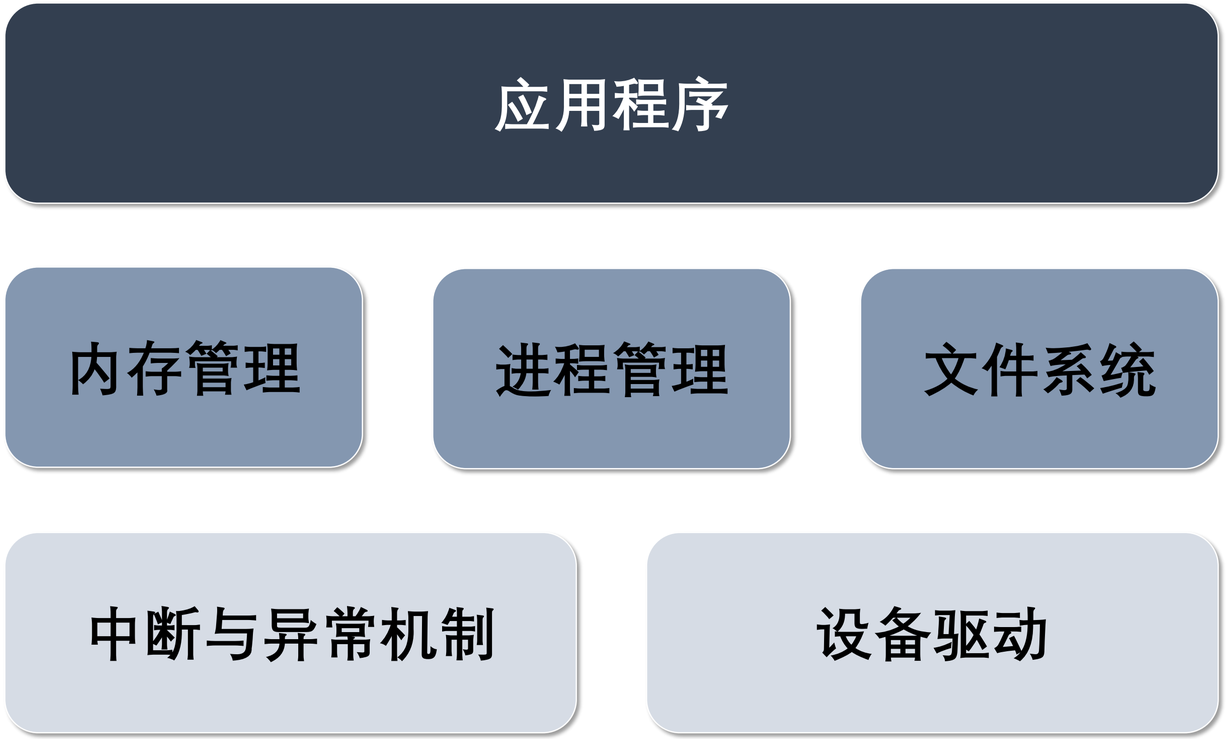


代码类型	数量
C文件	6478行
H文件	1189行
汇编文件	520行

ZJUNIX整体架构

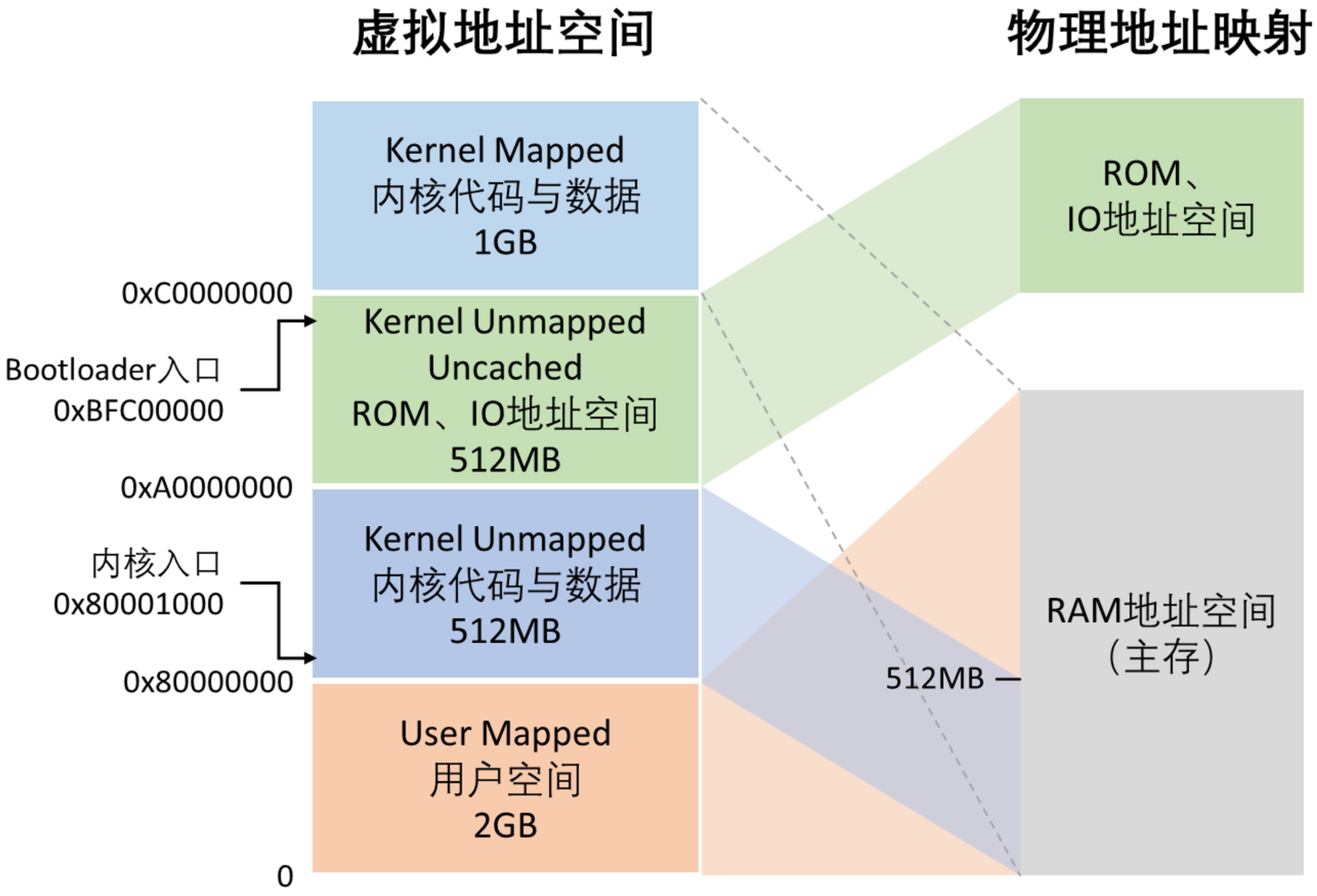

shell
ls
cd
.....
pwd
ps
kill
用户库ulib
系统调用API
内存管理
Slub
Buddy
进程管理
多线程
进程调度
文件系统
缓存管理
FAT 32
中断异常管理
UART驱动
VGA驱动
键盘驱动
SD卡驱动
时钟中断
硬件调试
MIPS SOC
SWORD平台
MIPS模拟器
PC平台
软件调试

ZJUNIX模块规划



ZJUNIX

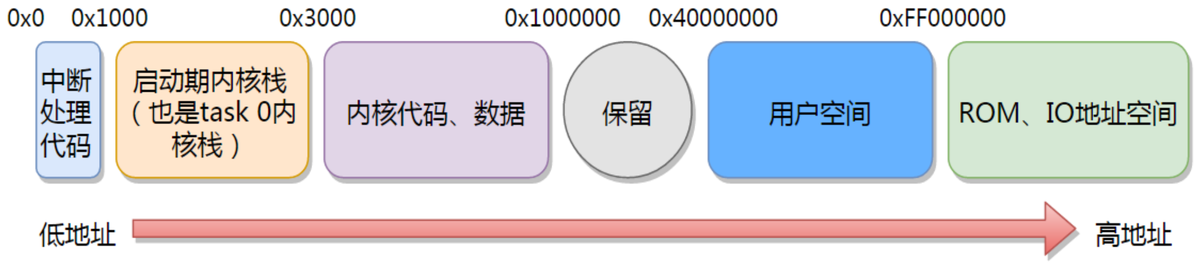
内存管理

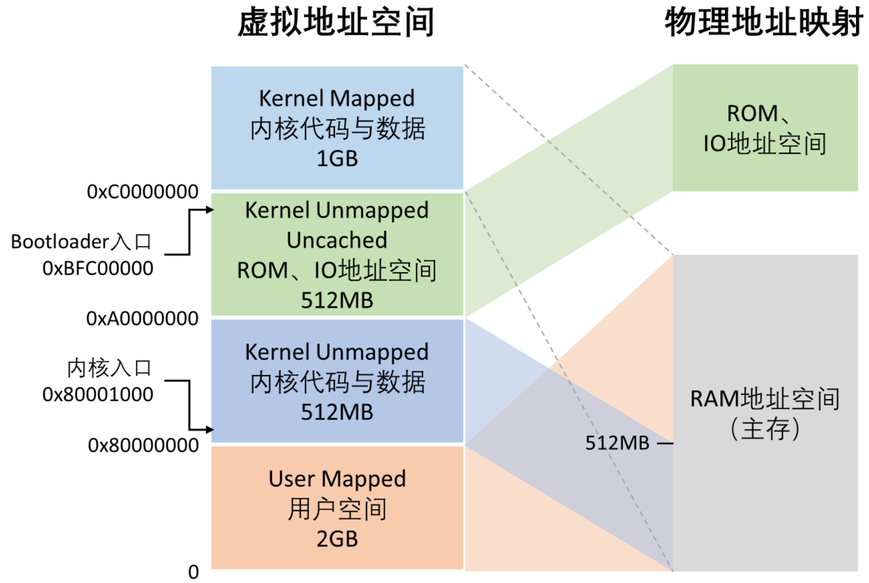

虚拟地址空间
物理地址映射
Kernel Mapped
内核代码与数据
1GB
0xC0000000
Kernel Unmapped
Uncached
ROM、IO地址空间
512MB
Bootloader入口
0xBFC00000
0xA0000000
Kernel Unmapped
内核代码与数据
512MB
内核入口
0x80001000
0x80000000
User Mapped
用户空间
2GB
0
ROM、
IO地址空间
512MB
RAM地址空间
（主存）

ZJUNIX

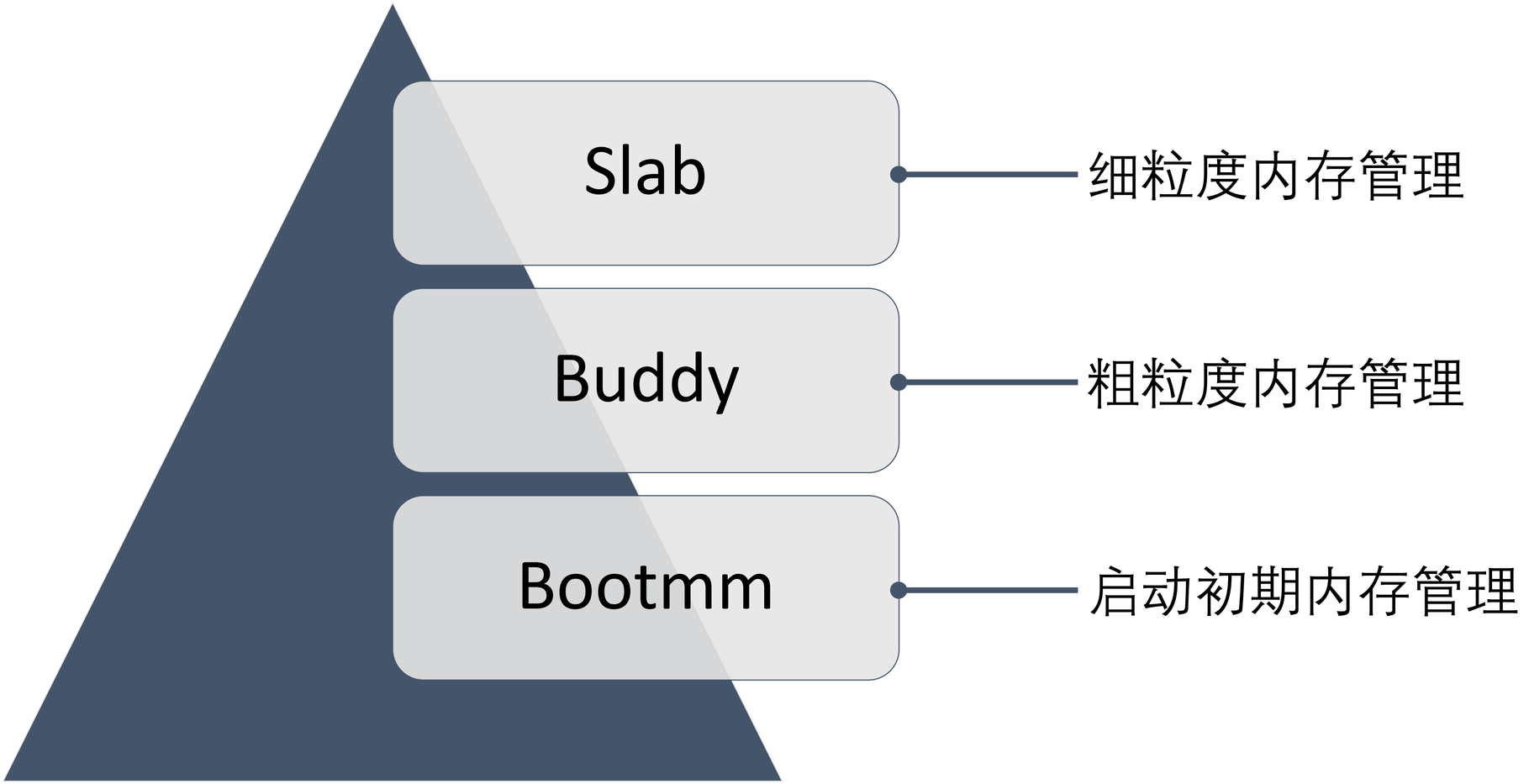
内存管理

- 虚拟地址空间划分、物理地址映射已经过升级，与原书设计不同
- 目前的设计与MIPS32规范兼容
- 实验时请以实验材料和软件代码为准



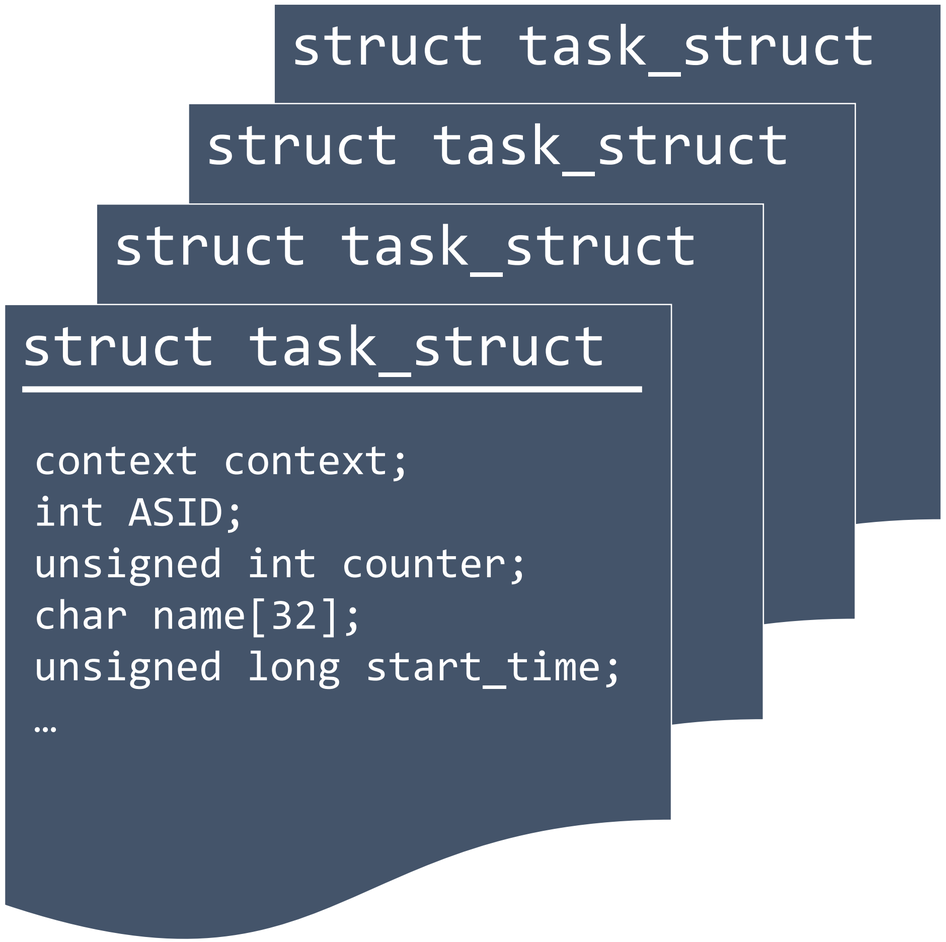


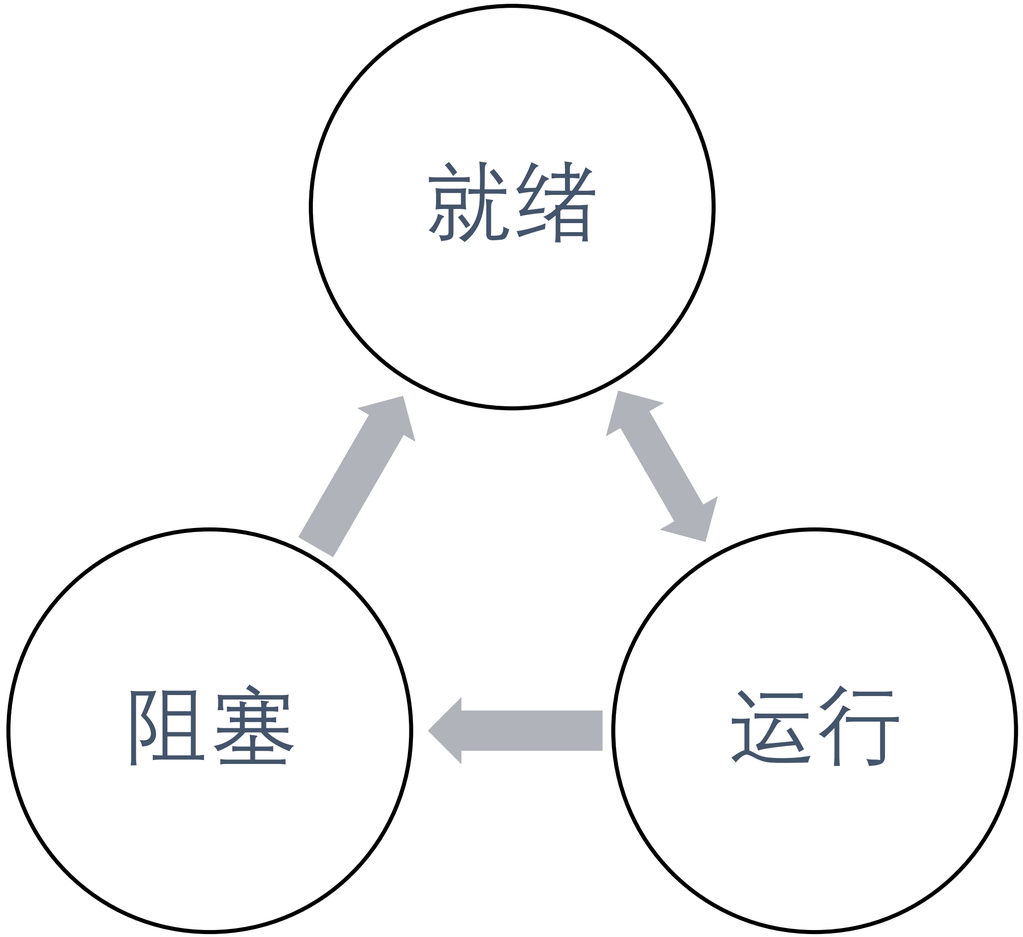
内存管理-模块划分



ZJUNIX

进程管理


struct task_struct
struct task_struct
struct task_struct
struct task_struct
context context;
int ASID;
unsigned int counter;
char name[32];
unsigned long start_time;
…


就绪
阻塞
运行

文件系统

- 结构简单
- 易于实现
- 跨平台通用性好
- 适合作为启动分区

设备驱动

- PS/2键盘驱动程序
- VGA显示驱动程序
- SD卡驱动程序
- UART串口驱动程序

应用程序

- Shell: 集成终端
- myvi: 文本编辑器
- ls、cat、ps等实用工具

实验分解及安排

3

操作系统整体规划与设计					
整体规划	内存管理设计	进程管理策略	文件系统选择	设备驱动种类	应用程序设计
操作系统启动与初始化					
系统启动流程		Bootloader设计		内核初始化流程	
中断与异常设计实现					
MIPS32异常机制		异常机制初始化		异常处理代码	

设备驱动设计与实现				
设备驱动架构	显示驱动	存储驱动	键盘驱动	其他设备
内存管理设计与实现				
整体设计	Bootmm设计	Buddy设计	Slab设计	应用接口
进程管理设计与实现				
虚拟内存与地址翻译	进程结构	进程调度		

文件系统设计与实现				
文件系统 初始化	文件控制块	FAT缓存机制	文件内容 缓存机制	文件操作
应用程序设计				
用户态机制	C库与系统调用	应用程序示例		


THANK YOU