基于RISC-V的操作系统教学尝试

宫晓利，李先铎，方睿丽，张蔚，向勇

THE MAIN CONTENTS

Part 1

目标和动机

01 操作系统的教学目标：

- 操作系统基础知识
  - 如进程调度、内存管理、文件系统等
- 操作系统是软硬件的接口
  - 软件如何配合和利用好硬件提供的功能
- 操作系统是一个典型的复杂工程问题
- 如果可以修改处理器，你觉得有哪些地方值得改进
- 如果你来设计一个操作系统，如何让它支持多种不同的处理器

02 操作系统需要一个好的硬件平台:

- X86：复杂，太多向前兼容的设计
- ARM：复杂，启动过程缺少标准
  - 并且以上两个硬件为成熟商用产品，不能修改定制
- RISC-V:简单，可定制，具备新特性
- MIPS：龙芯授权，也在尝试中

Part 2

实验内容

实验内容

JOS:

来自MIT 6.828课程的实验，一个非常简易的OS，能够实现启动、内存管理、进程、中断、系统调用等基本功能的学习

存在问题：代码太过简单，没有兼容和扩展性的考虑；网上的低质答案太多

ucore:

是一个由清华大学开发的教学用操作系统。它参考了UNIX,xv6的代码设计，支持系统引导、中断异常处理、基于页的虚拟内存管理、进程管理、进程间通信、文件系统等功能。保留了许多核心数据结构和编码方式。

该系统可以基于x86平台上运行。

实验内容

01 操作系统兼容体系结构的差异，即ucore在risc-v上的移植

- 系统启动
- 中断管理
- 基于页面的虚拟内存管理
- 内核线程与用户进程
- 进程调度通信与文件系统

02 体系结构的特性在操作系统中的使用

- PUM模式
- Cache模式

01 操作系统兼容体系结构的差异，即ucore在risc-v上的移植

1.1 系统启动

- x86：BIOS实现
- RISC-V：Berkeley Boot-loader，模拟了一个可以实现部分I/O功能的RISC-V实例。
  - Supervisor Binary Interface：通过ecall调用实现

提供配置好的环境以及文档供学生直接使用，做原理讲解，理解x86的复杂所在

1.2 中断管理

◆RISC-V：众多CSR寄存器

Name	Description
sstatus/mstatus	各种状态信息
sie/mie	处理器目前能处理和必须忽略的中断
stvec/mtvec	PC被设置为stvec
sscratch/mscrach	暂时存放一个字大小的数据
sepc/mepc	发生例外的指令被存入sepc
scause/mcause	表示异常类型
sbadaddr/mbadaddr	出错地址（这个CSR的功能在Pv1.10中被合并到另外一个寄存器里）
sip/mip	指出目前正准备处理(pending)的中断
sptbr	Page-table base register
medeleg	控制哪些异常（同步）委托给S模式
mideleg	控制哪些中断（异步）委托给S模式

S-MODE下的中断管理

- 硬件自动对S-MODE下的CSR进行一些处理
  - 此时的PC存入sepc，PC则被设为stcev中的值，也就是切入通用中断处理函数
  - scause会根据异常类型存入相应的错误码
  - sstatus中的SIE位的值存入SPIE中后置0全局屏蔽中断
  - 异常触发时所在的特权级放入sstatus中的SPP域，然后把当前特权级设为S-MODE
- 处理中断
  - 将所有会用到的整数寄存器和CSR的值保存起来
  - 进入trap函数
- 中断返回
  - 用之前保存的寄存器值恢复中断触发现场。

中断管理的比较

- x86：中断向量　中断描述符表
- RISC-V：为中断提供了多个CSR寄存器帮助处理，能够很快识别包括中断使能、中断委托、异常类型、出错地址、异常处理程序入口地址等在内的多个寄存器内容

寄存器的多少，是RISC与CISC的主要差别之一，也是影响性能的主要因素之一

1.3 虚拟内存管理

- X86：段页式内存管理
- RISC-V：页式内存管理
  - 页表格式：

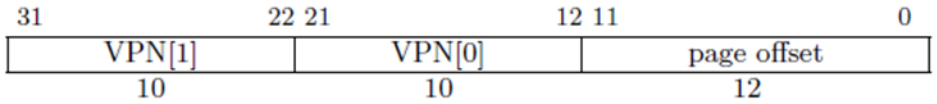


Figure 4.11: Sv32 virtual address.

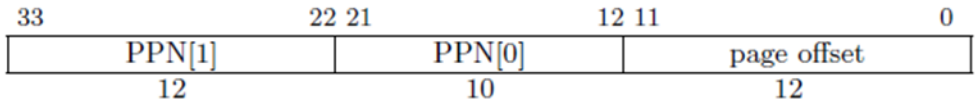
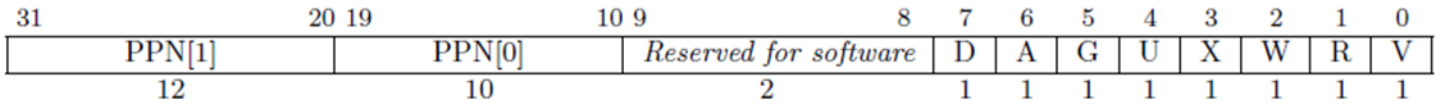


Figure 4.12: Sv32 physical address.



虚拟内存映射机制 （以SV32为例）

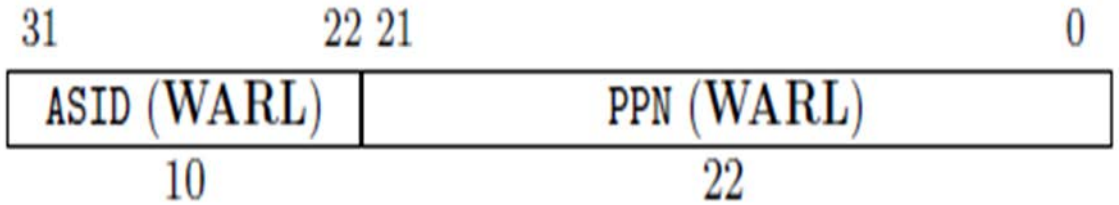
◆页表项中权限位编码

X	W	R	Meaning
0	0	0	Pointer to next level of page table.
0	0	1	Read-only page.
0	1	0	*Reserved for future use.*
0	1	1	Read-write page.
1	0	0	Execute-only page.
1	0	1	Read-execute page.
1	1	0	*Reserved for future use.*
1	1	1	Read-write-execute page.

虚拟内存映射机制 （以SV32为例）

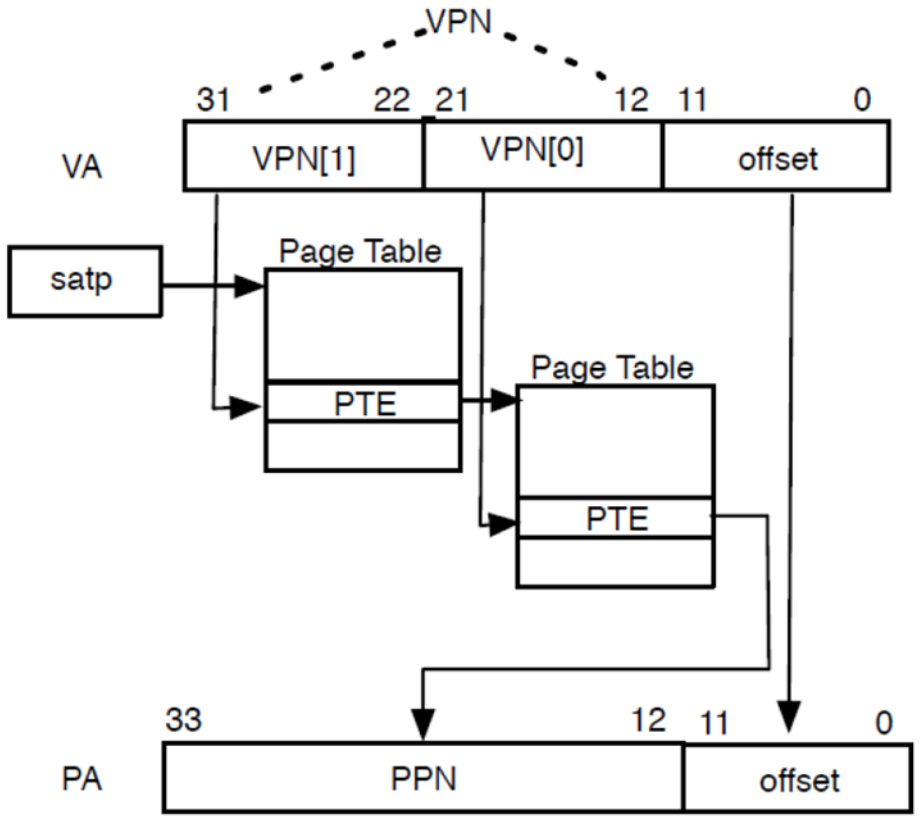
◆Sptbr寄存器

- ◆类似x86中cr3寄存器
- ◆保证取ASID和PPN为一个原子操作



虚拟内存映射机制 （以SV32为例）

◆地址转换过程



1.4 进程管理

◆ucore在RISC-V平台上的相关数据结构和执行逻辑与X86类似。

◆两个体系结构上的差异主要表现为两种体系结构寄存器名称、功能、设置的不同。

如何划分和融合面向体系结构的代码和面向OS流程的代码?

1.5 进程调度 进程同步

- 进程调度
  - 缺省的Round-Robin 调度算法
  - Stride Scheduling调度算法
- 进程同步
  - 信号量机制
  - 管程机制

X86和RISC-V两种体系架构上面的实现相同，让学生体会平台无关代码的优势。

02 体系结构的特性在操作系统中的使用

- 2.1 PUM模式
- 2.2 Cache模式

2.1 虚拟内存管理的新特性：PUM模式

◆内存保护措施：PUM模式

- ◆PUM模式是在虚拟内存模式下提供的一种内存保护措施。
- ◆通过设置mstatus 中的PUM 位。PUM=0 的时，一切访问和内存保护照常进行，如果PUM=1，则S_MODE 下不可以访问U_MODE（PTE中U=1）下可以访问的内存。

◆讨论

- ◆原有操作系统特权模式的安全隐患
- ◆如何应用这一机制到OS中？哪些部分会受到影响？

2.2 Cache的设计和管理

◆具体表现

- ◆传统80386平台：不带有cache
- ◆ARM920T平台：cache设置成VIVT虚地址索引，通过协处理器进行cache的刷新
- ◆RISC-V：L1cache和L2cache可以采用PIPT寻址模式

有助于学生加深对虚拟地址概念的理解

Part 3

当前状态

具体工作

- ucore 到Risc-V的移植
  - 张蔚（清华大学2015级学生），向勇，陈渝
- ucore on risc-v的复现与文档整理
  - 方睿丽（南开大学2015级学生），李先铎（中国矿业大学2016级学生），宫晓利
- 2018年秋，募集优秀学生进行实验尝试
- 2019年秋，在南开大学面向优秀学生进行试点

实验分级

- jos on x86
- ucore on x86
- ucore on risc-v

以X86为主，鼓励尝试，比对分析

Part 4

相关资源

相关资源

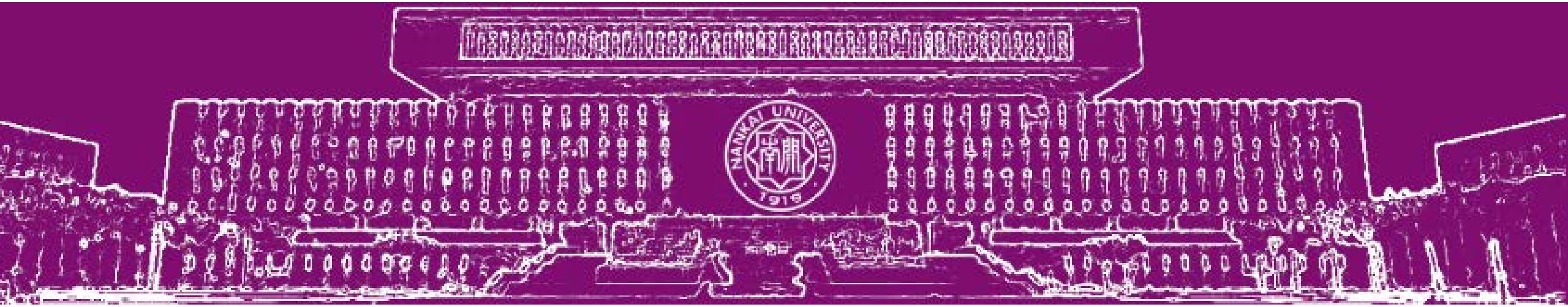
清华大学ucore教学网站：http://os.cs.tsinghua.edu.cn/oscourse/

Ucore on risc-v源码站： https://github.com/ring00/bbl-ucore

Ucore on risc-v 移植文档：

https://github.com/rllly/ucore_on_riscv_recordings

Thank you!



Embedded System And Information Security Lab