第二次学习报告

本周学习的内容如下

# 保护模式下的内存管理

## 学习了内存寻址

在80X86中，地址总线的宽度位32位，这意味着可以寻址4GB的物理内存。X86使用段寻址技术进行内存寻址。

## 学习了地址变换机制

X86实现了由逻辑地址向物理地址变化的技术。

程序中有逻辑地址，由16位的段选择符和32位的偏移量组成。处理器通过分段机制和分页机制将逻辑地址转换为物理地址。

## 学习了分段机制和分页机制

分段机制的操作过程：我们现在有一个逻辑地址，其中有16位的段选择符和32位的偏移量。首先由16位的段选择符在段描述符表中索引到相应的段描述符，段描述符中记录了相应段基地址。而32位偏移量指定了字节在段中的偏移量。于是我们就找到了线性地址。

分页机制的操作过程：现在我们有32位的线性地址。首先我们使用线性地址中的高20位作为页表数组的索引值，从而找到物理地址的基地址。而把线性地址的低12位作为页面中的偏移量，加上基地址，从而得到最终的物理地址。

分段机制中有两个重要的概念：段选择符、段描述符。它们中的一些位与操作系统的其他属性相关，这里就不扩展开来了。总之需要知道，由段选择符得到段描述符，由段描述符和偏移量得到线性地址。

分页机制中需要知道页表结构：X86使用二级页表结构，第一级表是页目录表，是存放第二级表的数组。而第二级表是存放物理基地址的数组。X86通过二级页表结构减少了页表对内存的占用。

## 学习了保护机制

保护机制包括分段保护机制和分页保护机制。

当段选择符寻找到段描述符的过程，需要将段选择符装入段寄存器中。这时我们需要进行段限长检查、段类型检查以及特权级检查。

# 记录锁

## fcntl函数的使用

记录锁的功能是，当一个进程正在读或修改文件的某个部分时，它可以阻止其他进程修改同一文件区。记录锁可以锁文件中的一个区域，也可以锁整个文件。

了解了unix编程中一个与记录锁有关的函数, int fcntl(int fields, int cmd, struct flock\* flockptr); cmd有三个参数F\_GETLK、F\_SETLK和F\_SETLKW。这三种模式在此就不展开讲了。

## linux源码fcntl.c

阅读了linux源码中对fcntl.c的实现。我发现它实际上是对进程文件结构指针数组项的值进行操作。fcntl.c中有sys\_dup()、sys\_dup2()函数，用于复制文件描述符。sys\_fcntl()函数用于为fcntl()函数提供实现，里面使用switch语句来选择不同的执行模式。

相对于上星期看的fork.c函数而言，fcntl.c比较简单。我想这也得益于自己对X86硬件体系结构的理解。

# 线程

## 线程标识

学习了如何在unix编程环境下使用线程。

用pthread\_create()创建线程，用pthread\_equal()比较线程，用pthread\_exit()终止线程，用pthread\_join()等待线程结束和获取线程的返回状态。

## 线程同步

学习了如何处理竞争条件。

unix下提供了互斥量来解决竞争条件。用pthread\_mutex\_init()初始化锁，用pthread\_mutex\_lock()对互斥量加锁，用pthread\_mutex\_unlock()对互斥量解锁，用pthread\_mutex\_destroy()销毁锁。具体如何使用在此就不展开讲啦，我自己有写例程体验一下的。

有了互斥量就会产生死锁。可以通过小心地控制互斥量加锁的顺序来避免死锁的发生。

**看了这章也能回答老师你第一次见面时问我的问题了：为什么不都使用原子操作呢？因为那样系统的并发性就很差。如果锁的粒度太粗，就会出现许多线程阻塞等待相同的锁，并发性就微乎其微。如果锁的粒度太细，那么过多的锁开销会使系统性能收到影响，而且代码将变得相当复杂。**

还学习了读写锁与条件变量。

# 内核编程语言

## as汇编语言

学习了as指令语句的标准格式，了解了操作数和操作码的基本格式。了解了内存引用、区、寻址方式等相关知识。

看了几个汇编的例子， 基本能看懂。

## 嵌入汇编

学习了嵌入汇编语句的基本格式asm(“汇编语句” :输出寄存器 : 输入寄存器 : 会被修改的寄存器)。并且阅读了几个例子。

## C与汇编程序相互调用的机制

了解到函数调用操作需要借助栈来支持，其中栈帧结构特别关键，有两个关键的指针，一个是栈指针，一个是帧指针。阅读了汇编语言下函数调用的例子。了解了汇编语言如何调用C函数，以及C程序中如何调用汇编函数。

# 课内内容

把课文内容看到了当前老师所讲的进度。

做了第六章的一些课后习题。

# 下周计划的学习任务

1. 看《Linux内核设计与实现》的内核同步部分，并对照着看源码。
2. 学习linux下的中断和异常保护。
3. 学习linux下的任务管理。
4. 学习bochs虚拟机的使用，学习一个多任务内核实例。
5. 跟紧老师上理论课的进度，写课后题。
6. 根据实际情况扩展阅读。