第 24卷 第 4期
2004年 8月

桂 林 电 子 工 业 学 院 学 报
JOURNAL OF GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

Vol. 24, No. 4
Aug. 2004

一个实用教学操作系统的设计与实现*

黄廷辉
(桂林电子工业学院 计算机系 ,广西 桂林 541004)

摘 要: 通过分析目前用于教学操作系统的特点,根据操作系统课程的主要内容和较难理解的知识点,提出实用教学操作系统设计的基本原则,并设计了一个简单、易理解、可实现的教学操作系统。这个教学操作系统实现的功能包括内核程序的装入、物理内存管理、中断处理、线程调度、线程同步和用户程序的执行。该系统能够在基于 X86的 PC机上直接运行,可以很好地帮助学生理解抽象的操作系统原理。

关键词: 教学操作系统; 操作系统; NACHOS; 线程
中图分类号: G642.0 文献标识码: A 文章编号: 1001-7437(2004)04-39-04

引言

在当代计算机系统中,操作系统管理和控制计算机系统中所有软件、硬件资源,它为用户使用计算机提供一个方便、灵活、安全、可靠的工作环境,是其它各类应用软件赖以存在的基础。因此,学习并掌握计算机操作系统的基本原理,不仅对电子信息类专业的学生和研究人员是必要的,而且对一般的计算机应用人员也是非常有益的。但是,处于计算机系统基础、支撑和核心的操作系统却又具有概念抽象、结构复杂和难于掌握的特点。理论教学与实验教学是现代高等教育的两个重要组成部分。培养富有创新精神和竞争意识的人才的目标,对我们的教学从理论到实践都提出了较高的要求,而且实践教学对于学生实际工作能力的培养起着至关重要的作用。国际一流大学的操作系统课程中,学生毫不例外地必须参加设计小型操作系统的实习工作。实践是学习操作系统的最好途径,通过实践,学生会发现概念不是那么抽象的,操作系统的结构也是可以逐步分析的。因此,设计一个能用于教学的操作系统模型,作为实习项目来帮助学生掌握操作系统概念和原理是非常有意义的。

1 现有教学操作系统

1.1 NACHOS

在国外的操作系统教学中,使用最多的教学操作系统是 NACHOS。NACHOS有一个特殊的结构: 操作系统内核和机器模拟器被编译成一个单一可执行的任务,它作为一个普通进程在某些主机操作系统上运行。在 NACHOS中运行的用户级进程是运行在一个 MIPS处理器的指令级模拟器上,而 NACHOS核心却运行在主机系统的本地硬件上。这种设计方法的优点是可以使用主机系统的调试器来调试设计的操作系统内核,操作系统的出错只影响一个用户进程[1]。但 NACHOS只能运行在特定的 MIPS机型的模拟器上,会存在一些其它缺点: 首先,因为它的核心不是保存在模拟器的 RAM中,这样,它不会受到内存容量的限制。这意味着内核数据结构的大小不是影响系统性能的重要因素,使得数据结构大小和空间容量的权衡不用考虑,这在一个实际操作系统设计中是不现实的;其次,NACHOS是与底层硬件相隔离,所有机器模拟器和被模拟的硬件之间的接口由 C++ 对象设计。这意味着学生不需要了解要访问的真正硬件是如何工作的,而这一点对于编写核心程序的程序员来说是很重要的。并且 NACHOS这种特殊结构很难移植到实际硬件机器上直接执行。

1.2 MINIX

MINIX是另一个广泛用于教学的操作系统,它与 NACHOS不同,它是可以运行在实际硬件上,或 X86模拟器 Bochs上。然而,MINIX是一个包括了虚拟内存、文件系统、设备驱动程序、网络和一整套用户

* 收稿日期: 2004-02-25
基金项目: 广西教育科学"十五"课题资助项目(2003B22)
作者简介: 黄廷辉(1970-),男,广西桂林人,桂林电子工业学院计算机系讲师,理学学士,主要从事操作系统及嵌入式系统等方面的研究.

模式程序的比较完整的操作系统,它由两万多行代码组成。它对于教学有点过于庞大和复杂,实现有一定难度。而且它已经实现了操作系统的全部基本功能,没有留下合适的练习让学生自己完成,这不利于操作系统教学。

1.3 通用操作系统

在现实中最后的一种方式是使用一个通用的操作系统用于教学。但这样做会有许多缺点:较难的设备驱动程序问题;通用的操作系统结构庞大很复杂,学生短时间很难掌握,有很少的教学价值。而且,实际的操作系统几乎已经实现了所有功能,学生不需要设计或实现一些子系统和自己感兴趣的部分。因此介绍涉及到核心程序的操作系统课程应避免采用通用的操作系统作为实习项目。

2 实用教学操作系统的设计与实现

2.1 设计原则

要开发一个优秀的教学操作系统,应该具备下列目标:提供一个真实的执行环境;便于调式;使用简单,符合课程的需要,可留出一些练习作为学生的作业;帮助学生熟悉真正操作系统的结构和设计;在操作系统课程结束后能实现一个小的用户程序真正运行的操作系统;提供可读性好和健壮的代码[2]。

作为一个操作系统内核的教程,保持系统的简单易实现是非常重要的。为了实现这一目的,在系统设计时,尽力地限制了操作系统的功能,仅保留操作系统最基本的功能。系统应该可以灵活的留出部分作业,让学生去完成,使他们能亲自参与到设计实践中去,激发他们学习操作系统的兴趣。

2.2 设计与实现过程

根据上面的设计原则,参照操作系统课程的主要内容和一些较难理解的知识点,我们设计了一个实用的教学操作系统,已经实现的功能主要有:系统引导;中断处理;堆内存分配程序;基于时间片内核级线程的静态优先级调度;用于内核级线程同步的互斥信号量和条件变量;基于段的内存保护和一个简单系统调用接口的用户模式;键盘以及显示适配器文本显示模式的设备驱动程序[3]。下面是各功能的具体实现。

2.2.1 系统的装入与运行

此系统是一个由软盘启动的操作系统。引导程序首先将一个 16位的 setup程序和内核映象从软盘调入内存,然后跳到 setup程序执行。setup程序主要是初始化硬件,确保系统进入 32位保护模式,并设置初始化内核级线程的堆栈,接着跳到 main( )函数执行。main( )函数是内核的入口点。

main( )函数首先调用用于初始化各种内核子系统的函数:BSS栈,屏幕,内存,TSS栈,中断处理器,线程调度器,时钟和键盘。然后开始执行用户程序,用户程序在此被定义为与内核相链接的特定的数据结构。

2.2.2 内存管理

目前此系统不支持虚拟存储,因此所有的内核代码均使用物理地址。系统以及它的数据结构使用物理内存的情况如图 1所示。引导程序将内核置于 10000h处。setup程序的代码和数据放在 9000h处,一旦内核的 main( )函数被调用,setup程序将不再有用,它所占用的内存可重新使用。从 A0000h到 100000h是 ISA段,保留给 PC硬件设备,目前仅用于 VGA文本显示空间,起始地址为 B8000h.

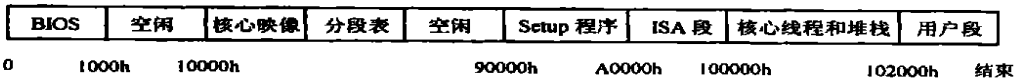

图 1 物理内存使用格式

在用 0初始化 BSS栈和初始化 VGA为文本模式之后,内核马上建立内存段表数据结构,这是一个记录数组,数组中的每一项对应于物理内存中的每一个段的使用情况,它是在内核映象被导入之后马上建立的。

段分配程序通过 Alloc_Page( )和 Free_Page( )函数来分配物理内存,可分配的段由所有地址低于 A0000h的,但不包括 0段(包含 BIOS数据)和用于内核镜像的段和段表的一系列段组成。段分配程序目前只用于创建内核级线程对象及其堆栈。

2.2.3 内核级线程

内核级线程是系统中的调度实体。之所以选择线程模型来实现是因为现代操作系统都引入了线程,对于使用用户级线程的用户来说应当熟悉。内核级线程结构定义如下:

```
struct Kernel_Thread {
    unsigned long esp; //线程挂起时保存线程的堆栈指针
```

```
unsigned long numTicks; /*定时器
int priority; /*线程优先权
DEFINE_LINK(Thread_Queue, Kernel_Thread); /*线程在线程队列中的前后域
void* stackPage; /*线程的堆栈指针
struct User_Context* userContext; /*指向线程的用户执行内容
struct Kernel_Thread* owner; /*线程的拥有者
int refCount;
Boolean alive;
struct Mutex joinLock; /*线程同步的互斥机构
struct Condition joinCond; /*线程同步的条件变量
};
```

内核级线程创建有两种方式: 完全以内核模式执行的线程由 Start_Kernel_Thread( )函数创建,函数将带回一个指向线程实体的函数起始地址的指针;而以用户模式执行的线程由 Start_User_Thread( )函数创建,函数将带回一个指向用户程序和用户程序在内存中的代码入口点的指针. 撤消内核级线程只要线程主动调用 Exit( )函数即可.

2.2.4 中断和线程调度

系统有两种情况会引起内核级线程上下文发生切换. 第一,一个线程可能通过调用 Yield( )和 Wait( )函数自愿放弃使用 CPU.在以上每个函数中,当前线程将被放到线程队列中,同时通过调用 Schedule( )函数选择一个新线程来执行. Yield( )函数将调用的线程放入 Run队列中,该线程有可能再次被调度. Wait( )函数将调用的线程放入等待队列中,线程一直处于等待状态,直到其他的线程或中断处理器调用 Wake_Up( )函数将其放回到运行队列中. 第二,如果允许中断,则正在执行的线程可能会在任何时候被抢占. 时钟中断处理程序在每一个时钟中断发生时,给正在执行的线程的 numTicks域加一,如果 numTicks值超过了给定的时间片,则该线程将被挂起,一个新的线程被选择来执行.

当前调度算法采用了一个静态优先权方案来决定下一个将被执行的线程. 同等优先权的线程将以轮流的方式共享 CPU.

2.2.5 线程同步

为了保证内核级线程同步,系统提供了互斥信号量和条件变量,它们可以由那些 POSIX线程提供的变量建模. 互斥信号量用来实现多个线程对某一数据结构的互斥性访问. 条件变量让线程等待一个条件是否满足,或者在一个条件被满足时发出一个信号.

2.2.6 用户模式

任何一个操作系统内核的用处最终在于它能否运行其他程序的能力. 它必须能保护内核不受用户程序所干扰和用户程序之间互不干扰. 系统内核采用了 X86的分段机制为用户程序创建一个受限制的环境. 用户代码可以访问和修改它自己的数据,也可以进行系统功能调用,但所有其他的操作系统资源对他来说是不能访问的. 企图通过某一用户程序访问它的代码和数据以外的任何操作都是非法的,系统将终止该非法线程.

系统通过把一个用户上下文附属到内核线程上来创建一个用户进程. 目前系统不支持存储器驱动程序或文件系统,它只提供了一种将用户程序作为数据对象直接链接到内核的方式. 用户可执行的代码首先被链接在基地址为 0的地方,从每一个可执行的代码中,产生一块连续的二进制映像. 然后,用一个 Perl脚本将二进制映像转换成 C数据结构. 这些数据结构被收集在一个表中,系统内核可以通过名字来查找它们,接下来将它们发送到 Start_User_Program( )函数.

用户程序通过系统调用接口与内核进行通信. 系统调用通过将调用的功能号送到 eax寄存器来进行的,将参数装入其他的寄存器,同时产生软中断. 中断处理程序将检查系统调用是否是一个合法的和是否允许中断,然后调用处理函数,关中断,将结果送回 eax寄存器. 系统提供了 copyin( )和 copyout( )函数用于在内核空间与用户空间之间进行数据传输.

2.2.7 学生作业

学生在学习操作系统课程的同时,可以熟悉教学操作系统的功能模块,然后在 PC机上运行操作系统,帮助他们理解课程学到的理论知识. 学生在真正了解系统的设计思想后,可对系统进行扩充,完成他们的作业: 设计一个简单的文件系统;修改线程的调度算法;增加网络功能;实现任务的动态创建和运行;实现虚拟存储管理.

2.3 开发环境和运行方式

用于开发教学操作系统的计算机,需要构建一个装配有如下开发工具的环境: Linux操作系统平台,汇编语言编译器 nasm, C语言编译器 gcc,一个连接器和“make”工具,还有一个特别有用的工具——Bochs. Bochs是一个 x86硬件平台的模拟器,它是开发和调式操作系统核心代码最有效的工具;对新加入的代码行进行测试不用重新启动用于开发的机器;不

需要使用软盘作为操作系统的存储介质[4]。

运行教学操作系统有两种方式:一是可以在Bochs模拟器上执行编译得到的可执行核心映像文件来运行;二是把编译得到的可执行核心映像文件拷贝到软盘中,做成系统盘,然后直接用软盘启动实际计算机来运行。

3 结束语

这是一个用于教学的操作系统,简单实现了操作系统的启动、内存管理、中断处理、线程调度和用户程序执行,可以较好地帮助学生理解操作系统概念和原理。但这个系统还没有实现分页虚拟存储管理、文件系统和网络通信等操作系统基本功能,需要不断完善。当然在扩展这些功能的同时,必须保持系统简单、易用的特点,使它适合于操作系统课程的教学。

参考文献:

[1] Thomas Narten. A Road Map Through Nachos [EB/OL]. http://www.cs.duke.edu/~narten/110/nachos/main/main.html, 1997-02-03.

[2] David A Holland, et al. A New Instructional Operating System [EB/OL]. http://www.eecs.harvard.edu/~syrah/papers/sigcse-02/, 2002-12-13.

[3] GeekOS web site [EB/OL]. http://geekos.sourceforge.net, 2003-12-17.

[4] Lawton K. Bochs the open source ia-32 emulation project [EB/OL]. http://bochs.sourceforge.net, 2004-01-13.

[5] 顾宝根,等.操作系统实验教程-核心技术与编程实例[M].北京:科学出版社,2003.

[6] 胡元义,徐甲同.操作系统实践教程[M].西安:西安电子科技大学出版社,2001.

[7] 陈向群,等. Windows内核实验教程[M].北京:机械工业出版社,2002.

[8] (美)William Stallings著,魏迎梅,等译.操作系统-内核与设计原理[M].北京:电子工业出版社,2001.

The Designing and Implementing of a Practical Teaching Operating System

HUANG Ting-hui

(Dept. of Computer, Guilin 541004, China)

Abstract To master the principle of operating system, it is necessary for us to participate in the designing practice. The characteristics of the present operating system for teaching are discussed in this paper. Then, a basic principle is presented to design a teaching operating system which is simple, easily realized and understandable. The designing and the implementing of the teaching operating system are described in detail. The mainly features of this system include the load of kernel program, the management of physical memory, the handling of interrupt, the thread synchronization, and the running of the user program. The operating system kernel could be run on real hardware(X86-based PCs) and could help students learn the principles of the system.

Key words teaching operating system, operating system, NACHOS, thread

(责任编辑 陶晓玲)