

高等院校计算机实验与实践系列示范教材

操作系统实验教程

张坤 姜立秋 赵慧然 编著



清华大学出版社

高等院校计算机实验与实践系列示范教材

操作系统实验教程

张坤 姜立秋 赵慧然 编著

清华大学出版社
北京

内 容 简 介

本书是“操作系统”课程的辅助教材,通过其中的实验可加深对本课程概念的理解。全书共分为 10 章,从设计一个简单的操作系统引导程序开始,指导学生初步理解操作系统的设计原理和过程,使学生能够掌握简单的 Shell 编程,掌握进程、线程、进程管理、进程调度和进程通信等基本概念和技术,掌握内存管理基本概念和技术,掌握文件操作和磁盘调度的基本原理。

本书的实验以 Linux 操作系统为主。因为 Linux 是开放源码的,因此,在本书附录中给出了 Linux0.11 的部分源码分析,可以让读者得以一窥操作系统的内部实现机制。

书中的实验题目具有独立性,并且对每个实验中用到的知识,在前面都给予描述和指导,这样有利于读者通过自学掌握实验教程中的技术和方法。

本书可作为高等院校计算机及相关专业“操作系统”课程实验的辅助教材,也可作为读者学习操作系统技术的实验指导教程。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

操作系统实验教程/张坤,姜立秋,赵慧然编著. —北京:清华大学出版社,2008.6

(高等院校计算机实验与实践系列示范教材)

ISBN 978-7-302-17734-0

I. 操… II. ①张… ②姜… ③赵… III. 操作系统—高等学校—教材 IV. TP316

中国版本图书馆 CIP 数据核字(2008)第 076109 号

责任编辑:索 梅

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:17 字 数:414 千字

版 次:2008 年 6 月第 1 版 印 次:2008 年 6 月第 1 次印刷

印 数:1~4000

定 价:26.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028652-01

出版说明

当前,重视实验与实践教育是各国高等教育界的发展潮流,我国与国外教学工作的差距也主要表现在实践教学环节上。面对新的形式和新的挑战,完善实验与实践教育体系成为一种必然。为了培养具有高质量、高素质、高实践能力和高创新能力的人才,全国很多高等院校在实验与实践教学方面进行了大力改革,在实验与实践教学内容、教学方法、教学体系、实验室建设等方面积累了大量的宝贵经验,起到了教学示范作用。

实验与实践性教学与理论教学是相辅相成的,具有同等重要的地位。它是在开放教育的基础上,为配合理论教学、培养学生分析问题和解决问题的能力以及加强训练学生专业实践能力而设置的教学环节;对于完成教学计划、落实教学大纲,确保教学质量,培养学生分析问题、解决问题的能力 and 实际操作技能更具有特别重要的意义。同时,实践教学也是培养应用型人才的重要途径,实践教学质量的好坏,实际上也决定了应用人才培养质量的高低。因此,加强实践教学环节,提高实践教学质量,对培养高质量的应用型人才至关重要。

近年来,教育部把实验与实践教学作为对高等院校教学工作评估的关键性指标。2005年1月,在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出:“高等学校要强化实践育人的意识,区别不同学科对实践教学的要求,合理制定实践教学方案,完善实践教学体系。要切实加强实验、实习、社会实践、毕业设计(论文)等实践教学环节,保障各环节的时间和效果,不得降低要求。”“要不断改革实践教学内容,改进实践教学方法,通过政策引导,吸引高水平教师从事实践环节教学工作。要加强产学研合作教育,充分利用国内外资源,不断拓展校际之间、校企之间、高校与科研院所之间的合作,加强各种形式的实践教学基地和实验室建设。”

为了配合开展实践教学及适应教学改革的需要,我们在全中国各高等院校精心挖掘和遴选了一批在计算机实验与实践教学方面具有潜心研究并取得了富有特色、值得推广的教学成果的作者,把他们多年积累的教学经验编写成教材,为开展实践教学的学校起一个抛砖引玉的示范作用。

为了保证出版质量,本套教材中的每本书都经过编委会委员的精心筛选和严格评审,坚持宁缺毋滥的原则,力争把每本书都做成精品。同时,为了能够让更多、更好的实践教学成果应用于社会和各高等院校,我们热切期望在这方面有经验和成果的教师能够加入到本套丛书的编写队伍中,为实践教学的发展和取得成效做出贡献;也衷心地期望广大读者对本套教材提出宝贵意见,以便我们更好地为读者服务。

清华大学出版社

联系人:索梅 suom@tup.tsinghua.edu.cn

操作系统是现代计算机系统中不可缺少的系统软件。如果让用户去使用一台没有配置操作系统的计算机,那是难以想象的。操作系统控制和管理整个计算机系统中的软硬件资源,并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。本书遵循操作系统课程的教学大纲要求,针对应用型模式的专业定位和人才培养目标而编写。

操作系统课程的实验环节一直是操作系统教学的难点。本书通过 Windows 和 Linux 两个操作系统各自的编程接口,提供一些编程实例,以此加深学生对操作系统工作原理的领会和对操作系统实现方法的理解,并且使学生在程序设计方面得到基本训练。

全书共分为 10 章,从设计一个简单的操作系统引导程序开始,指导学生初步理解操作系统的设计原理和过程,使学生能够掌握简单的 Shell 编程,掌握进程、线程、进程管理、进程调度、进程通信等基本概念和技术,掌握内存管理基本概念和技术,掌握文件操作和磁盘调度的基本原理。

结合操作系统理论课程的学习,通过本书的全部实验,使学生能够理解操作系统的一般概念和方法,有能力去分析、设计和改变一个操作系统的某些组件。

本书的大部分程序在 Linux 下实现,可为后续的嵌入式 Linux 系统课程打下很好的基础。为了更好地让读者理解操作系统的原理和实现技术,其余程序是在 Windows 下实现的一些模拟算法。所有程序都经过测试和验证。

本书附有 Linux0.11 的部分源码分析,这主要是为了让读者能够更加深入地理解一个真实操作系统的内部原理。附录还精选了一些习题,供广大师生选用。

实验要求:

1. 明确实验目的,掌握相关背景知识。
2. 熟练掌握实验内容和方法。
3. 每个实验都要求学生写出实验报告,同时注意各实验中给出的“实验任务”。

4. 教师最后写出实验总结。

在本书的编写过程中,姜立秋编写了第4章、第9章和第10章,赵慧然编写了附录B、附录C、附录D和附录E,全书由张坤提出编写大纲并编写了其余章节。由于编者水平有限,错误和不妥之处敬请读者提出宝贵意见。

本书配有电子课件,读者可从清华大学出版社网站(<http://www.tup.com.cn>)下载。

编 者

2008年5月

第 1 章 引导操作系统的过程	1
1.1 实验目的	1
1.2 预备知识	1
1.3 实验内容	2
1.3.1 简单汇编程序设计	2
1.3.2 查看启动记录	3
1.3.3 修改启动记录	5
1.3.4 用 NASMW 编译一个自己的 启动程序(. bin 文件)	7
第 2 章 Linux 基本环境	9
2.1 实验目的	9
2.2 预备知识	9
2.2.1 Linux 的安装	11
2.2.2 Linux 命令行(Shell 环境)	15
2.2.3 文件系统命令	15
2.2.4 用户及系统管理常用命令	19
2.2.5 网络操作常用命令	23
2.2.6 Linux 下软件安装	24
2.2.7 使用编辑器 vi 编辑文件	27
2.2.8 GCC 编译器	29
2.2.9 Gdb 调试器	35
2.2.10 编写包含多文件的 Makefile	40
2.3 实验内容	41
2.3.1 Linux 基本操作练习	41
2.3.2 Makefile 的应用	41
第 3 章 Shell 编程	44
3.1 实验目的	44

3.2	预备知识	44
3.2.1	Shell 概述	44
3.2.2	Shell 的特点和命令行书写规则	44
3.2.3	常用 Shell 类型	45
3.3	实验内容	45
3.3.1	简单 Shell 程序设计	45
3.3.2	Shell 脚本的建立和执行	46
3.3.3	Shell 变量	47
3.3.4	Shell 中的特殊字符	50
3.3.5	表达式的比较	51
3.3.6	控制结构	53
3.3.7	综合应用	59
第 4 章	进程管理	61
4.1	实验目的	61
4.2	预备知识	61
4.2.1	进程相关基本概念	61
4.2.2	Linux 下系统调用	62
4.2.3	Windows 下的系统调用	65
4.2.4	进程调度算法	68
4.3	实验内容	68
4.3.1	进程的创建	68
4.3.2	进程的控制	69
4.3.3	文件的加锁、解锁	70
4.3.4	Windows 下的进程管理	71
4.3.5	进程调度模拟算法	79
第 5 章	进程间通信	84
5.1	实验目的	84
5.2	预备知识	84
5.2.1	管道	84
5.2.2	消息	84
5.2.3	共享内存	88
5.2.4	信号机制	91
5.3	实验内容	94
5.3.1	进程的管道通信	94
5.3.2	消息的创建、发送和接收	95
5.3.3	共享存储区的创建、附接和段接	96



5.3.4	消息队列和共享存储区性能比较	98
5.3.5	信号机制举例	98
第 6 章	进程(或线程)同步与多线程编程	100
6.1	实验目的	100
6.2	预备知识	100
6.2.1	进程(或线程)同步概述	100
6.2.2	线程概述	102
6.3	实验内容	105
6.3.1	生产者-消费者问题	105
6.3.2	进程、线程综合应用	108
第 7 章	死锁避免——银行家算法	109
7.1	实验目的	109
7.2	预备知识	109
7.2.1	死锁的概念	109
7.2.2	死锁预防	109
7.2.3	死锁避免	110
7.3	实验内容	110
7.3.1	实现银行家算法所用的数据结构	110
7.3.2	银行家算法	110
7.3.3	源程序清单	111
7.3.4	设计输入数据、验证银行家算法	115
第 8 章	存储管理	116
8.1	实验目的	116
8.2	预备知识	116
8.3	实验内容	119
8.3.1	可变分区主存分配和回收	119
8.3.2	请求页式存储管理	123
第 9 章	文件操作	129
9.1	实验目的	129
9.2	预备知识	129
9.3	实验内容	131
第 10 章	磁盘调度	140
10.1	实验目的	140
10.2	预备知识	140

10.3 实验内容	141
附录 A 80386 基础	149
附录 B 操作系统练习题与参考答案	160
附录 C 综合测试题及其参考答案	184
附录 D 操作系统自测题	189
附录 E Linux0.11 系统引导程序	215
附录 F Linux0.11 进程调度	227
附录 G Linux0.11 中信号的处理	240
附录 H Linux0.11 的内存管理	247
参考文献	260

1.1 实验目的

1. 通过简单汇编程序设计及 DEBUG 调试程序的使用,了解学习操作系统课程必备的基础知识(计算机体系结构、CPU、内存、BIOS 等)。
2. 能够在软盘上创建一个简单的系统引导程序。

1.2 预备知识

本节通过学习 DOS 的启动程序,了解操作系统的启动顺序,然后仿照 DOS 的启动程序设计自己的启动程序。

首先说明当按下电源按钮后,计算机都做了什么?

当按下计算机电源按钮时,同这个按钮相连的电线就会送出一个电信号给主板,主板将此电信号传给供电系统,供电系统开始为整个系统供电,同时送出一个电信号给 BIOS(基本输入输出系统),通知 BIOS 供电系统已经准备完毕。随后,BIOS 启动一个程序,进行主机自检。主机自检的主要工作是确保系统的每一部分都得到了电源支持,内存存储器、主板上的其他芯片、键盘、鼠标、磁盘控制器及一些 I/O 端口正常可用。此后,自检程序将控制权交还给 BIOS。

此时,BIOS 开始启动操作系统。

BIOS 首先访问启动盘的第 1 个扇区(0 磁道,1 扇区,一共是 512 字节),这一部分称为 DOS 启动记录(DOS Boot Record, DBR)。BIOS 将这第一扇区中的内容调入内存的 0x7c00 地址处,然后 BIOS 把控制权交给这段引导程序。这是启动系统的第一关,引导程序通常会简单地执行一些指令,如输出一段文字、显示一个启动界面等。但最重要的是,引导程序将会启动一个更大的程序——操作系统内核。之后,系统将控制权交给操作系统。

现在我们的任务就是写这样一个引导程序,用它来引导(或者称为启动)计算机。引导程序有如下两个特点。

- (1) 大小只能是 512 字节,不能多,也不能少。因为 BIOS 只能读

512 字节的数据到内存中,多的部分 BIOS 不会理睬。

(2) 必须以“55 AA”结尾,即最后两个字节必须是(511,512)。这是引导区程序结束的标志,没有这个标志 BIOS 不会将它作为引导程序看待。

把所编写的引导程序放在磁盘的 0 磁道 1 扇区中,这样,此磁盘就可以用来引导系统,而且用的是自己编写的引导程序! 由于不能随便更改硬盘(否则系统无法进入原来的操作系统),所以这里用软盘来试验。

注意: 在 CMOS 中将软盘设为第一启动盘。

1.3 实验内容

1.3.1 简单汇编程序设计

首先熟悉汇编语言的汇编环境和一些基本命令。请看下面两个例子的执行。

例 1.1

```
CODE SEGMENT
    ASSUME CS:CODE
START: MOV  BL,30H
RRR:   MOV  AL,BL
        INC  BL
        CMP  BL,3AH
        JA   STOP
        MOV  DL,AL
        MOV  AH,02H
        INT  21H
        MOV  DL,2CH
        MOV  AH,02H
        INT  21H
        JMP  RRR
STOP:   MOV  AX,4C00H
        INT  21H
CODE ENDS
END START
```

用 MASM 和 LINK 命令汇编和运行这段程序,DOS 命令如下:

```
C:\>masm 1. asm
C:\>link 1
C:\>1
```

实验任务: 写出显示结果并用至少两种方法修改程序,去掉 9 后面的逗号。

例 1.2

```
DATA SEGMENT
    STR DB 'HELLO!','$'
```

```

DATA    ENDS
STA     SEGMENT STACK
        DW 50 DUP(0)

STA     ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA,SS:STA
START:  MOV  AX,DATA
        MOV  DS,AX
        LEA  DX,STR
        MOV  AH,9
        INT  21H
        MOV  AH,4CH
        INT  21H
CODE    ENDS
        END  START

```

运行这段程序,将在屏幕上显示字符串“HELLO!”,DOS 命令如下:

```

C:\>masm 2. asm
C:\link 2
C:\2
C:\debug 2. exe
;下面反汇编,找到最后一条指令 INT  21H 的偏移地址 000f
-u
;下面运行到偏移地址 000f
-g 000f
HELLO!
AX=4C24  BX=0000  CX=0091  DX=0000  SP=0064  BP=0000  SI=0000  DI=0000
DS=0BB0  ES=0BA0  SS=0BB1  CS=0BB8  IP=000F  NV UP EI PL NZ NA PO NC
0BB8:000F CD21          INT    21
-d0
0BB0:0000  48 45 4C 4C 4F 21 24 00-00 00 00 00 00 00 00 00  HELLO! $......
0BB0:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....
.....

```

通过例 1.1 和例 1.2 可以观察计算机程序在内存中的实际执行状况,了解各种寄存器的功能和中断调用的执行过程,熟悉 DEBUG 命令的使用。

1.3.2 查看启动记录

例 1.3 使用 DEBUG 程序查看软盘的启动记录(Boot Record)。

对应刚刚格式化好的驱动器 A,在 DEBUG 程序中输入如下命令:

```
-l 0 0 0 1(前面是 l 不是 1)
```

这个命令是把软盘中内容加载到内存,后四个数字的顺序是:加载数据的内存首地

址、驱动器号(0 表示一软盘驱动)、盘中要加载的第一个扇区号和共有几个扇区要加载。

再输入命令:

-d 0

此时可以看见软盘启动记录中前 128 字节,共 8 行的内容如下:

```
0AF6:0000 EB 3C 90 4D 53 44 4F 53-35 2E 30 00 02 01 01 00 .<. MSDOS5.0.....
0AF6:0010 02 E0 00 40 0B F0 09 00-12 00 02 00 00 00 00 00 ...@.....
0AF6:0020 00 00 00 00 00 00 29 F6-63 30 88 4E 4F 20 4E 41 .....). c0. NO NA
0AF6:0030 4D 45 20 20 20 20 46 41-54 31 32 20 20 20 33 C9 ME FAT12 3.
0AF6:0040 8E D1 BC F0 7B 8E D9 B8-00 20 8E C0 FC BD 00 7C ....{.... ....|
0AF6:0050 38 4E 24 7D 24 8B C1 99-E8 3C 01 72 1C 83 EB 3A. 8N$ } $....<. r...:
0AF6:0060 66 A1 1C 7C 26 66 3B 07-26 8A 57 FC 75 06 80 CA f..|&f;,. &. W. u...
0AF6:0070 02 88 56 02 80 C3 10 73-EB33 C9 8A 46 10 98 F7 ..V....s. 3.. F...
```

其中,左边是段首地址和偏移量,中间是数据,右边是对应的翻译字符。

输入如下命令:

-u 0

列出该段数据对应的代码如下,其中右边是代码段数据对应的反汇编指令:

```
0AF6:0000 EB3C      JMP      003E
0AF6:0002 90          NOP
0AF6:0003 4D          DEC     BP
0AF6:0004 53          PUSH    BX
0AF6:0005 44          INC     SP
0AF6:0006 4F          DEC     DI
0AF6:0007 53          PUSH    BX
0AF6:0008 352E30      XOR     AX,302E
0AF6:000B 0002      ADD     [BP+SI],AL
0AF6:000D 0101      ADD     [BX+DI],AX
0AF6:000F 0002      ADD     [BP+SI],AL
0AF6:0011 E000      LOOPNZ 0013
0AF6:0013 40          INC     AX
0AF6:0014 0BF0      OR      SI,AX
0AF6:0016 0900      OR      [BX+SI],AX
0AF6:0018 1200      ADC     AL,[BX+SI]
0AF6:001A 0200      ADD     AL,[BX+SI]
0AF6:001C 0000      ADD     [BX+SI],AL
0AF6:001E 0000      ADD     [BX+SI],AL
```

实验任务: 写出自己上述-u0 后显示器前 4 行的显示内容。

上述反汇编出来的第 1 条指令是跳转到地址 0x3E 处。查看该指令的作用可输入如下命令:

-u 3E

-d 180

屏幕显示如下：

```

0AFC:0180 18 01 27 0D 0A 49 6E 76-61 6C 69 64 20 73 79 73  ..'..Invalid sys
0AFC:0190 74 65 6D 20 64 69 73 6B-FF 0D 0A 44 69 73 6B 20  tem disk...Disk
0AFC:01A0 49 2F 4F 20 65 72 72 6F-72 FF 0D 0A 52 65 70 6C  I/O error...Repl
0AFC:01B0 61 63 65 20 74 68 65 20-64 69 73 6B 2C 20 61 6E  ace the disk,an
0AFC:01C0 64 20 74 68 65 6E 20 70-72 65 73 73 20 61 6E 79  d then press any
0AFC:01D0 20 6B 65 79 0D 0A 00 00-49 4F 20 20 20 20 20 20  key....IO
0AFC:01E0 53 59 53 4D 53 44 4F 53-20 20 20 53 59 53 7F 01  SYSMSDOS SYS..
0AFC:01F0 00 41 BB 00 07 60 66 6A-00 E9 3B FF 00 00 55 AA  .A...fj...;...U.

```

启动记录准确占用盘中的 512 字节。加载到内存后开始地址为 0，最后的字节的地址是 0x1FF。如果向上看两个字节的地址 0x1FE 和 0x1FF，它们的数据是 0x55 和 0xAA。这两个字节是启动盘标志，如果 BIOS 没有检测到这两个字节就不会认为它是启动盘。

实验任务：写出上述“-d 180”命令后，读者的显示器前 8 行显示内容并思考为什么从偏移地址 180 开始显示？

1.3.3 修改启动记录

例 1.4 修改启动记录。要求在不改变启动部分的数据的情况下，替代启动加载代码。如果被改变的某些数据是违法的，Windows 将出现错误信息。

(1) 首先需要跳转到 0x3E 处去修改代码。运行 DOS DEBUG 程序并在 0 地址处加载已格式化的软盘到内存。输入如下命令：

```
-u 3E
```

查看此处的指令。然后就开始修改代码。输入如下命令：

```
-a 3E
```

接下来输入：

```
jmp 3E
```

整个程序在计算机看起来是这样的：

```

-a 3E
0AFC:003E jmp 3E
0AFC:0040

```

可以看到，第 1 条命令是跳转命令，创建了一个无限跳转。如果马上停止 DEBUG 程序运行，没有变化可以保存。

现在开始修改启动部分。输入如下命令：

```
-w 0 0 0 1
```

这条“写”命令的使用方法同 L 命令相同。这条“写”命令必须小心使用,它会被任何驱动改写,造成数据丢失。

现在可以启动软盘。启动后, BIOS 可以加载第 1 部分内容到内存,并执行。然后执行跳转到 0x3E 指令。计算机只是停在那里什么也没有做,但是新的“操作系统”已经运行。

(2) 如果希望在程序运行时能够显示一些信息,即看见一些有标记的代码确实在运行,则可进行如下的修改。因为引导时,操作系统还没有启动,不能使用 DOS 中断,为此,可用一些 BIOS 中断调用。寄存器设置如下:

AH=0x0E

AL=要显示的 ASCII 字符码

BL=字符的颜色

指令如下:

-a 3E

0AF6:003E mov ah,0e

0AF6:0040 mov al,48

0AF6:0042 mov bl,07

0AF6:0044 int 10

0AF6:0046 jmp 46

0AF6:0048

-w 0 0 0 1

上述代码首先令 AH 置 0e(显示字符的功能号),AL 为 0x48(ASCII 码字符“H”),BL 为 7(颜色为黑白色);然后调用中断 0x10 处理显示控制。最后一条指令创建了一个无限跳转,所以就停止了。保存修改的磁盘中的启动部分(-w 0 0 0 1)并重新启动,这时会看见字母“H”被显示出来。

将上述程序进一步修改为可与用户交互的程序:

-a 3E

0AF6:003E mov ah,0

0AF6:0040 int 16

0AF6:0042 mov ah,0e

0AF6:0044 mov bl,07

0AF6:0046 int 10

0AF6:0048 jmp 48

保存修改的磁盘中的启动部分(-w 0 0 0 1)并重新启动,可以从键盘输入任意一个字符显示出来。如果将最后一条语句修改为“jmp 3E”,可以实现无限循环输入并显示。

实验任务: 仿照上面程序,编程完成输入 10 个字符显示,然后进入无限循环(提示:用 loop 循环指令)。