

第3章 实训内容

本章内容提要：

- 4 个使用级实训目标及要求；
- 8 个管理级实训目标及要求；
- 8 个观察分析级实训目标及要求；
- 6 个编程与实现级实训目标及要求；
- 两个源代码分析级实训目标及要求。

操作系统是一门理论性和实践性都很强的课程。要学好操作系统的设计原理，除了听课、看书、做习题外，最好的方法就是在实践中进行，包括使用操作系统、阅读和分析已有操作系统的源代码、自己设计小型系统、模块或模拟算法等。

本实践教程安排的实验内容按深度可分为 5 个层次，即使用级、观察级、系统管理级、源码阅读级和实现级，包括操作系统的进程管理、存储管理、设备管理、文件管理、安全管理和用户接口的使用等多个方面。由于内容相对较多，读者可根据自己的需要以及实验条件等酌情选择。

内容使用建议

面向应用、难度适中的使用级、系统管理级、观察分析级实验和用户级 API 编程实验是必做题，旨在锻炼独立使用、观察和分析操作系统的能力，以及独立利用操作系统提供的字符命令和图形用户接口、系统调用与服务等来管理、配置计算机和解决基于操作系统应用问题的实践能力，具体数量可根据需要裁剪。其余为选做题。

3.1 使用级

3.1.1 安装 Linux

1. 实训目的

在供实训的微型计算机上安装 Linux 操作系统，后续实训内容都将在此平台上进行。通过实训，要求：

- (1) 了解硬件资源要求。
- (2) 学会安装 Linux 系统。
- (3) 学会启动 Linux 系统。
- (4) 了解 Linux 多引导器的配置。

2. 实训内容

把 Linux 安装到本地硬盘。

主要安装步骤提示：

- (1) 执行下列操作之一。
 - ① 如果 BIOS 已设置为支持光盘启动，则插入 Linux 安装光盘，重新启动计算机。

② 如果已制作好 Linux 安装 U 盘,则检查确认 BIOS 已设置支持 U 盘启动,插入 U 盘并重新启动计算机。

③ 如果从 DOS 环境启动,则在 DOS 提示符下执行批处理命令,如 autoboot。

(2) 对硬盘分区,留出交换空间和根文件系统的空间。

(3) 按提示分阶段装入系统。

(4) 配置系统(可选默认值,但要搞清楚含义)。

3.1.2 安装 Windows XP

1. 实训目的

(1) 通过对 Windows XP 的安装操作,了解操作系统应用环境建立的初步过程。

(2) 掌握 Windows XP 操作系统的基本系统设置。

(3) 了解 Windows XP 多操作系统安装配置的方法。

2. 实训内容

把 Windows XP 安装到本地硬盘。注意:若要与 Linux 并存于同一硬盘上,则把硬盘分成至少两个分区,并且先安装 Windows XP。

主要安装步骤与安装 Linux 相似,在此从略。

注意,按屏幕提示选择合适的文件系统时,建议选择 NTFS。输入的管理员密码要记住。

3.1.3 Linux 系统用户接口和编程界面

1. 实训目的

(1) 熟悉 Linux 字符界面的常用命令和窗口系统。

(2) 熟悉 Linux 常用的编程工具。

(3) 熟悉 Linux 的在线帮助系统。

(4) 掌握在 Linux 操作系统环境上编辑、编译、调试和运行一个 C 语言程序的全过程。

2. 实训内容

(1) 熟悉开机后登录 Linux 系统和退出系统的过程。

(2) 熟悉 Linux 字符界面——虚拟终端窗口和 Shell,以及图形界面——X-Window(如 gnome 或 KDE):练习并掌握常用的 Linux 操作命令,如 ls、cat、ps、df、find、grep、cd、more、cp、rm、kill、at、vi、gcc、man 等;熟悉常用 Shell 的提示符;熟悉字符窗口与图形界面之间的切换。

(3) 学习使用 Linux 的在线帮助系统,如 man 和 help 命令等。

(4) 掌握一种 Linux 的编辑器,特别是字符界面的编辑器 vi 的使用。

(5) 用 vi 编辑一个输出“Hello, I am a C program”字符串的 C 语言程序,然后编译并运行它,记下整个过程。熟悉 gcc、gdb 等编译器、调试器的使用。

3. 要点提示

1) vi 的最基本使用

vi 编辑器有两种工作状态(或模式):命令状态和编辑状态。当 vi 处于命令状态时,把输入的内容全都视为命令;当 vi 处于编辑状态时,把输入的内容全都作为文本。即仅当 vi

处于编辑状态时,才可以往被编辑的文件(比如源程序)里输入文本内容,而当 vi 处于命令状态时,只能输入命令,这时无法往文件里输入文本。

vi 刚启动时处于命令状态。

从 vi 的命令状态切换到编辑状态需按 I 或 A 键;而从 vi 的编辑状态切换到命令状态需按 Esc 键。

vi 最常用命令(必须在命令状态下输入,注意大小写字母)如表 3-1 所示。

表 3-1 vi 最常用命令

命 令	功 能 说 明
a	从命令状态切换到编辑状态,并从光标所在处后面开始添加文本
i	从命令状态切换到编辑状态,并从光标所在处前面开始插入文本
:q!	放弃存盘,强行退出 vi
:wq	存盘,退出 vi
dd	删除光标所在的行
Ndd	删除从光标所在行开始的连续 n 行
Y	复制当前行到编辑缓冲区
NY	复制当前行开始的连续 n 行到编辑缓冲区
p	将编辑缓冲区的内容粘贴到光标后的一行
P	将编辑缓冲区的内容粘贴到光标前的一行

vi 常用的编辑键(在编辑状态下使用)如表 3-2 所示。

表 3-2 vi 常用的编辑键(功能键)

编辑键	功 能 说 明	编辑键	功 能 说 明
←	光标向左移一个字符	→	光标向右移一个字符
↑	光标向上移一个字符	↓	光标向下移一个字符
Home	光标移至所在行之首	End	光标移至所在行之尾
Page Up	向前翻一页	Page Down	向后翻一页
Del	删除光标后的一个字符	Backspace	删除光标前的一个字符
Esc	从编辑状态切换到命令状态		

假如要编辑一个当前目录下名为 test.c 的 C 语言源程序,则首先需在命令行输入命令“vi test.c”回车,按 I 输进入编辑状态,输入完文本内容后,按 Esc 键切换到命令状态,再输入命令“:wq”(注意此命令第一个符号是个冒号)即可。

vi 是个功能强大的文本编辑器,若想了解更多的操作方法,请参阅相关书籍,或者用命令“man vi”查看其使用帮助信息。

2) C 程序编译及运行过程(主要是编译器 gcc 的使用)

以上面的 test.c 的编译运行过程为例进行说明。首先需在命令行输入命令 gcc test.c

回车,若直接出现命令行提示符,则说明该源程序编译成功,否则,会显示出错提示信息。若编译成功,则当前目录下有个名为 a.out 的可执行文件(这就是刚才 gcc 正确编译 test.c 后默认的输出结果),这时在命令行输入命令“./a.out”回车,即可执行该程序。

若用 gcc 编译时出错,则需要再次修改源程序,并再次编译它。

若不想用 gcc 默认的输出文件名,可用开关-o。例如,若想把上面的程序编译后的可执行结果放在文件 test 中,则只需在命令行输入命令“gcc test.c -o test”回车即可。此后,在命令行输入命令“./test”回车,即可执行该程序。

3.1.4 Windows 操作系统界面认识

1. 实训目的

- (1) 熟悉 Windows XP 字符界面的常用命令和窗口系统。
- (2) 熟悉 Windows XP 基本的文件管理和系统管理工具。
- (3) 熟悉 Windows XP 的在线帮助系统。

2. 实训内容

- (1) 熟悉开机后登录 Windows XP 和退出系统的过程。
- (2) 熟悉 Windows XP 字符界面——“命令提示符”窗口或“开始”|“运行”|CMD 窗口:练习并掌握常用的 Windows XP 操作命令,如 dir、type、edit、cd、copy、xcopy、del、help、exit、ping、netstat、ipconfig、regedit 等;熟悉常用 Windows XP 的命令行提示符;熟悉字符窗口与图形界面之间的切换。熟悉 Windows XP 的“工作桌面”、“开始程序组”及“任务栏”的组成内容。
- (3) 学习使用 Windows XP 的在线帮助系统,如“开始”|“帮助”窗口和 help 命令等。
- (4) 熟悉 Windows XP“资源管理器”或“我的电脑”的窗口组成和功能。
- (5) 了解 Windows XP“管理工具”的具体内容。

3.2 系统管理级

3.2.1 在 Linux 中添加、删除用户及用户组

1. 实训目的

- (1) 了解 Linux 提供的用户及用户组操作命令的功能和使用;
- (2) 掌握普通用户切换到 root 用户的方法;
- (3) 用上述内容在 Linux 中添加、删除用户及用户组。

2. 实训内容

- (1) 在普通用户的命令行提示符下相继输入命令 su 和 root 用户口令转入超级用户模式;
- (2) 用 useradd 命令创建用户 ccec;
- (3) 用 groupadd 命令创建 ccecgrou 组,并将用户 ccec 加入到组 ccecgrou 中;
- (4) 用 id 命令验证上述操作的正确性;
- (5) 用 userdel 命令删除 ccec 用户,再使用 groupdel 命令删除 ccecgrou 组;

(6) 用 ls 命令查看第(5)步的结果。

3.2.2 在 Windows XP 中添加、删除用户

1. 实训目的

了解并使用 Windows 图形界面的用户管理工具添加和删除用户。

2. 实训内容

- (1) 单击“开始”按钮,选择“设置”|“控制面板”,在控制面板里选择“管理工具”,在管理工具里选择“计算机管理”;
- (2) 单击左侧“本地用户和组”,选择用户,在界面右侧,即能看到当前系统中已存在的用户;
- (3) 在左侧栏用鼠标右击,在弹出菜单中可新建用户;在右侧栏选中用户,右击,可删除用户。

3.2.3 在 Linux 中使用 U 盘

1. 实训目的

- (1) 了解 mount 和 umount 命令的功能和使用;
- (2) 了解 Linux 设备(重点是硬盘和 U 盘)文件的命名;
- (3) 用上述所了解的内容解决在 Linux 中使用 U 盘的问题。

2. 实训内容

- (1) 在普通用户的命令行提示符下相继输入命令 su 和 root 用户口令转入超级用户模式;
- (2) 用 mount 命令把 U 盘的设备文件安装到/mnt 目录;
- (3) 进入/mnt 目录,使用 U 盘(比如,往里复制一个文件);
- (4) 退出/mnt 目录,用 umount 命令把 U 盘从系统中卸载;
- (5) 按 Ctrl+D 键,退出超级用户模式(即返回到普通用户模式);
- (6) 总结在 Linux 中使用 U 盘的过程。(注意,在超级用户模式下请勿随意删除文件或更改口令。)

3.2.4 屏蔽 Windows XP 桌面上的“回收站”

1. 实训目的

- (1) 了解 Windows XP 注册表的作用,熟悉注册表编辑器;
- (2) 熟悉 Windows XP 注册表中 5 个主要根键的内容与作用;
- (3) 了解备份注册表的重要性,掌握备份注册表的基本方法;
- (4) 学会通过修改注册表,屏蔽桌面“回收站”图标的方法。

2. 实训内容

- (1) 备份注册表(或者打开后导出注册表)。
- (2) 运行 regedit 命令打开注册表。
- (3) 选择注册表中 HKEY_LOCAL_MACHINE 子窗口(即找到此根键),定位到

HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows \ CurrentVersion \ Explorer\Desktop\NameSpace 分支,在该分支下有多个子键对应桌面上的某些系统图标。

- (4) 删除键值为 Recycle Bin 的子键,如子键 645FF040-5081-1-1B-9F08-00AA002F954E。
- (5) 重新启动计算机。
- (6) 根据备份的注册表内容,恢复注册表中被删除的子键,重新启动计算机。
- (7) 观察两次重启后的情况,总结备份、修改和恢复注册表的一般过程。

3.2.5 停止 Windows XP“自动升级”服务

1. 实训目的

- (1) 了解 Windows XP 启动后自动开启的服务;
- (2) 熟悉 Windows XP 服务管理工具;
- (3) 学会通过 Windows XP 服务管理工具停止和启动指定服务的方法。

2. 实训内容

(1) 搜集资料,了解 Windows XP 启动后自动开启的服务有哪些,其中哪些是可以关闭或应该禁止的。

(2) 单击“开始”|“设置”|“控制面板”|“管理工具”|“服务”,启动 Windows XP“服务管理器”。

(3) 在窗口右边的服务列表中找到 automatic updates 所在的行,双击它,在出现的“常规”对话框中,“启动类型”选择“手动”,“服务状态”选择“停止”,单击“确定”按钮,关闭打开的各窗口。

(4) 重新启动计算机,观察前后的变化。

(5) 注意,这个实训内容在 Windows XP 中对应的操作过程还可以是:右击“我的电脑”,单击“属性”,单击“自动更新”,在“通知设置”一栏中选择“关闭自动更新。我将手动更新计算机”一项。

3.2.6 在 Linux 中配置 FTP 服务器

1. 实训目的

- (1) 了解 FTP;
- (2) 了解 Linux FTP 服务器实现方法;
- (3) 用上述所了解的一种方法在 Linux 中安装、配置、测试 FTP 服务器。

2. 实训内容

(1) 在普通用户的命令行提示符下相继输入命令 su 和 root 用户口令转入超级用户模式;

(2) 安装 vsftpd;

(3) 用 vsftpd 安装、配置、测试 FTP 服务器;

(4) 总结在 Linux 中配置 FTP 服务器的过程。

3.2.7 在 Linux 中配置 AMP 环境

1. 实训目的

- (1) 了解 LAMP;
- (2) 学会 LAMP 的安装及简要配置方法;
- (3) 掌握简单测试所配置的 LAMP 的方法。

2. 实训内容

- (1) 在普通用户的命令行提示符下相继输入命令 `su` 和 `root` 用户口令转入超级用户模式;
- (2) 安装 Linux Apache、MySQL Server、PHP;
- (3) 配置 Apache、MySQL Server、PHP;
- (4) 测试 LAMP;
- (5) 总结 Apache、MySQL 以及 PHP 如何配合起来提供 Web 服务。

3.2.8 在 Windows 上配置 IIS 服务

1. 实训目的

- (1) 了解 Windows 组件及 IIS;
- (2) 了解并掌握 IIS 安装及配置方法;
- (3) 学会配置 Windows FTP 服务器的方法。

2. 实训内容

- (1) 打开“控制面板”,双击“添加或删除程序”图标,选择左侧的“添加/删除 Windows 组件”,进入 Windows 组件向导;
- (2) 勾选“文件传输协议(FTP)服务”,安装 IIS;
- (3) 配置并测试 IIS;
- (4) 配置并测试 FTP 服务器;
- (5) 总结 Windows 上配置 IIS 服务的一般过程。

3.3 系统行为观察与分析级

3.3.1 观察 Linux 进程/线程的异步并发执行

1. 实训目的

- (1) 了解进程、线程与程序的区别,加深对进程、线程概念的理解;
- (2) 掌握进程并发执行的原理,理解进程并发执行的特点,区分进程并发执行与串行执行;
- (3) 了解 `fork()` 系统调用的功能和使用格式,掌握用 `fork()` 创建进程的方法;
- (4) 熟悉 `waitpid`、`exit` 等系统调用。

2. 实训内容

- (1) 编写一个 C 语言程序,实现在程序运行时通过系统调用 `fork()` 创建两个子进程,

使父、子三个进程并发执行,父亲进程执行时屏幕显示“I am father”,儿子进程执行时屏幕显示“I am son”,女儿进程执行时屏幕显示“I am daughter”。

(2) 编译后,多次连续反复地运行这个程序,观察屏幕显示结果的顺序,直至出现不一样的情况为止。记下这种情况,试简单分析其原因。

(3) 修改程序,在父、子进程中分别使用 wait、exit 等系统调用“实现”其同步推进,多次反复运行改进后的程序,观察、记录并分析运行结果。

(4) 把(1)中的进程改为线程,重新编程观察并分析运行结果。

3. 要点提示

fork()函数共有三个返回值:0、正数、-1。0 值表示该系统调用创建子进程成功,CPU 控制权在子进程手里(即这是返回给子进程的值);正数值也表示该系统调用创建子进程成功,但控制权在父进程手里(即这是返回给父进程的值,此值实为子进程的标识符);-1 值表示该系统调用创建子进程失败。

3.3.2 观察 Linux 进程状态

1. 实训目的

学习 Linux 操作系统的进程状态,并通过编写一些简单代码来观察各种情况下,Linux 进程的状态,进一步理解进程的状态及其转换机制。

2. 实训内容

(1) 了解 Linux 进程状态转换机制和查看进程状态的 ps 命令、向进程发送消息的 kill 命令等的功能与使用方法。

(2) 编写一个嵌套两级大循环的 C 语言程序 run_status.c,编译后,后台运行它,并使用 ps 命令观察该进程的状态。

(3) 编写一个 C 语言程序 interruptible_status.c,该程序调用 sleep 函数,使其进入睡眠状态。编译后,后台运行它,并使用 ps 命令观察该进程的状态。

(4) 编写一个 C 语言程序 uninterruptible_status.c,该程序调用 vfork 函数创建一个子进程,子进程调用 sleep 函数进入睡眠状态。编译后,后台运行它,并使用 ps 命令观察该进程的状态。

(5) 编写一个 C 语言程序(zombie_status.c),该程序的功能是创建一个子进程,然后调用 sleep 函数进入睡眠状态,而子进程什么也不做,直接退出。编译后,后台运行它,并使用 ps 命令观察该进程的状态。

(6) 后台运行上面的 run_status 进程,使用 kill 命令,向其发送 SIGSTOP 信号,并使用 ps 命令观察其状态,然后用 kill 命令向其发送 SIGCONT 信号,再使用 ps 命令观察其状态。

(7) 总结 Linux 进程的状态定义及转换机制。

3.3.3 在 Linux 中使用信号量实现进程互斥与同步

1. 实训目的

学习 Linux 信号量通信机制,通过实验进一步理解进程间同步与互斥、临界区与临界资源的概念及含义,掌握使用 POSIX 有名信号量进行进程同步、互斥的基本方法。

2. 实训内容

(1) 了解 POSIX 有名信号量机制有关的系统调用 `sem_open`、`sem_wait`、`sem_post`、`sem_close`、`sem_unlink` 等的功能、参数含义、返回值和使用方法。

(2) 编写一个 C 语言程序,进行 10 次循环,每个循环中,屏幕输出两次给定的字符。在使用互斥和不使用互斥的两种情况下,观察多个进程运行时的输出情况。通过实验结果理解进程互斥的概念。

(3) 编写两个 C 语言程序,分别模拟下象棋过程中的红方与黑方。简化的走子规则为:开始红先黑后,而后黑、红双方轮流走子,到第 10 步,红方胜,黑方输。要求使用 POSIX 有名信号量实现红黑方走子过程中的同步。将正确编译链接后得到的红黑方程序,分别在前后台同时执行,观察运行结果。

(4) 总结利用 POSIX 有名信号量机制实现进程间互斥与同步的方法。

3.3.4 在 Linux 中实现进程间高级通信

1. 实训目的

学习 Linux 进程间通信机制,通过实现一个简化的经典 IPC 问题——生产者/消费者问题示例程序,掌握通过 System V 共享内存接口实现进程间高级通信的基本方法,同时进一步加深对于使用 Linux 信号量实现进程同步的理解。

2. 实训内容

(1) 学习 Linux 进程通信机制,特别是共享存储区机制、消息缓冲队列和管道机制等高级通信机制,了解 System V 共享存储区通信机制有关的系统调用 `shmget`、`shmat`、`shmdt`、`shmctl` 等的功能、参数含义、返回值和使用方法。

(2) 编写一个 C 语言程序,解决简化的生产者-消费者问题。要求使用 System V 共享内存来模拟有界的环形缓冲池,而生产者与消费者进程之间的同步与互斥的实现,仍采用 POSIX 的有名信号量机制。

一个简化的生产者-消费者问题描述如下。

一组生产者进程通过一个具有 10 个缓冲区的缓冲池不断地向一组消费者进程提供产品。要求:

① 生产者进程:依次在不同的空缓冲区存放不同的产品,在第 0 个缓冲区写入字符 a,在第 1 个缓冲区写入字符 b,……,在第 9 个缓冲区写入 j,每生产一个产品,在屏幕上打印“Producer-pid: write m”,其中 pid 为该生产者进程的进程 ID,m 为此次生产者进程写入的字符。

② 消费者进程:依次从满缓冲区中读出生产者进程写入的字符,并在读出后,在该缓冲区写入字符 X(如消费者进程读出字符 X,则显然进程同步出现问题),并在屏幕上打印“Customer-pid: read m”,其中 pid 为该生产者进程的进程 ID,m 为此次消费者进程读出的字符。

(3) 总结利用共享存储区机制、信号量机制实现进程间通信的方法。

3.3.5 在 Linux 中共享文件

1. 实训目的

(1) 熟悉 Linux 支持的文件共享方式;

(2) 掌握 Linux 的基于索引节点和基于符号链接的文件共享方法。

2. 实训内容

- (1) 复习教材或查资料,了解并熟悉 Linux 支持的几种文件共享方式;
- (2) 熟悉 ln 命令的功能和用法;
- (3) 掌握用 ln 命令实现基于索引节点和基于符号链接的文件共享的方法;
- (4) 用 cat、ls 等命令验证自己实现的结果;
- (5) 总结该实训过程,写出实训报告。

3. 要点提示

ln 命令不带开关“-s”时可实现基于索引节点的文件共享,带开关“-s”时可实现基于符号链接的文件共享。注意,实现符号链接时,作为参数之一的被链接文件的名称应采用绝对路径名。

3.3.6 观察 Linux 内存分配结果

1. 实训目的

学习如何利用 Linux 的 malloc 函数动态申请一段内存空间。

2. 实训内容

- (1) 了解 Linux 进程空间布局。
- (2) 了解 malloc 函数的功能和 Linux 虚拟内存管理的原理。
- (3) 编写一个 C 语言程序,用 malloc 函数申请一段存储空间,并在终端上显示其起始地址。
- (4) 运行该程序,观察、记录其运行结果,并分析说明结果的地址是否为物理地址。

3.3.7 观察 Windows XP 注册表的内容

1. 实训目的

了解 Windows XP 注册表的组成、作用和基本的使用方法。

2. 实训内容

- (1) 了解 Windows XP 注册表的由来、组成(包括键、子键、键名、键值等)和作用;
- (2) 了解 Windows XP 注册表的备份及恢复的方法;
- (3) 了解 Windows XP 注册表的编辑器;
- (4) 单击“开始”、“运行”或者在“命令提示符”窗口,输入“regedit”,打开注册表编辑器,熟悉该窗口的各组成部分;
- (5) 注册表窗口的左子窗口中显示的是树状的注册表键值结构,记下注册表各个根键的名称,并分析其作用;
- (6) 记录在注册表键 HKEY_LOCAL_MACHINE\SOFTWARE 中包含的子键内容,分析其作用;
- (7) 查看“自启动”程序,记录或者导出欲修改的自启动分支项,删除右边窗格中的一些键值来禁止某些程序的自启动;
- (8) 重新启动系统,查看程序的自启动情况,再用保存的内容恢复对注册表的改动。

3. 要点提示

Windows XP 的自启动程序对应的注册表项主要是 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion 子键下的 Run 分支、RunOnce 分支和 RunOnceEx 分支。

3.3.8 观察并分析 Windows XP 任务管理器显示的内容

1. 实训目的

通过观察分析 Windows XP 任务管理器显示的内容,学会利用它管理计算机、分析计算机性能瓶颈。

2. 实训内容

- (1) 查资料,了解 Windows XP 任务管理器的功能和显示的内容。
- (2) 打开 Windows XP 的任务管理器,观察并分析其显示的有关进程、性能(特别是内存、CPU 利用情况)等内容。
- (3) 总结如何利用屏幕显示的内容管理计算机、了解计算机性能瓶颈。

3.4 实现级

本级的前 3 个实训内容既可在 Linux,也可在 Windows XP 操作系统平台下进行。

3.4.1 进程调度模拟程序设计

1. 实训目的

加深对进程的概念、属性及进程调度过程/算法的理解。

2. 实训内容

- (1) 给出进程调度的算法描述。
- (2) 用 C 或 C++ 语言设计一个对 n 个并发进程进行调度的程序,每个进程由一个进程控制块(PCB)结构表示,该进程控制块应包括下述信息:进程标识 ID、进程优先数 PRIORITY(优先数与优先权成正比)、进程需要的服务时间、进程已运行的时间、进程还需要运行的时间(初始值为进程服务时间,当进程运行结束时,为 0)、进程的状态 STATE(设每个进程处于运行 E(executing)、就绪 R(ready)和完成 F(finish)三种状态之一,并假设起始状态都是就绪状态 R)。为简化起见,所有的时间均以时间片为单位。
- (3) 模拟调度程序应实现先来先服务 FCFS、时间片轮转(RR)以及简单动态优先级三种调度算法,运行时选择一种,以利于各种方法的分析与比较。为简化起见,本实验对动态优先级进行简化:进程每运行一个时间片,优先级减 1,直到进程的最低优先级 1;进程调度时,选择就绪队列中优先级最高的进程运行。
- (4) 程序应能显示或打印各种进程状态和参数变化情况,便于观察分析。即要在每个时间片开始时,显示或打印每个进程的状态、优先级、还需要的服务时间、已运行时间等信息,并统计完成进程的周转时间及加权周转时间。

3.4.2 页面置换模拟程序设计

1. 实训目的

加深对请求页式存储管理实现原理的理解,掌握页面置换算法。

2. 实训内容

(1) 用 C 或 C++ 语言设计一个程序,模拟一个进程的执行过程。设该进程共有 320 条指令,其地址空间为 32 页(即每个页面中可存放 10 条指令),初始获得 4 个内存块,目前它的所有页面都还未调入内存。在模拟过程中,如果所访问的指令已经在内存,则显示其物理地址,并转下一条指令。如果所访问的指令尚未装入内存,则发生缺页,此时需记录缺页的次数,并将相应页调入内存。如果 4 个内存块中均已装入该作业的虚页面,则需进行页面置换。最后显示其物理地址,并转下一条指令。在所有 320 条指令执行完毕后,计算并显示进程运行过程中发生的缺页率。

(2) 置换算法:请分别考虑 OPT、FIFO 和 LRU 算法。

(3) 进程中指令的访问次序要求按下述原则生成。

① 50% 的指令是顺序执行的。

② 25% 的指令是均匀分布在前地址(即低地址)部分。

③ 25% 的指令是均匀分布在后地址(即高地址)部分。

具体的实施办法是:

① 在 $[0, 319]$ 之间随机选取一条起始执行指令,其序号为 m ;

② 顺序执行下一条指令,即序号为 $m+1$ 的指令;

③ 通过随机数,跳转到前地址部分 $[0, m-1]$ 中的某条指令处,其序号为 $m1$;

④ 顺序执行下一条指令,即序号为 $m1+1$ 的指令;

⑤ 通过随机数,跳转到后地址部分 $[m1+2, 319]$ 中的某条指令处,其序号为 $m2$;

⑥ 顺序执行下一条指令,即序号为 $m2+1$ 的指令;

⑦ 重复“跳转到前地址部分、顺序执行、跳转到后地址部分、顺序执行”的过程,直至执行完全部 320 条指令。

(4) 分析程序运行的结果,谈一下自己的认识。

3.4.3 文件系统模拟设计

1. 实训目的

加深对文件系统的内部功能及其实现的理解。

2. 实训内容

用 C 或 C++ 语言设计一个简单的二级文件系统。要求做到以下几点。

(1) 可以实现下列几条命令(至少 4 条)。

① login 用户登录

② dir 列文件目录

③ create 创建文件

④ delete 删除文件

⑤ open 打开文件

⑥ close 关闭文件

⑦ read 读文件

⑧ write 写文件

(2) 列目录时要列出文件名、物理地址、保护码和文件长度。

(3) 源文件可以进行读写保护。

3.4.4 为 Linux 添加一个系统调用

1. 实训目的

(1) 熟悉 Linux 系统调用工作原理；

(2) 掌握为 Linux 添加系统调用的方法。

2. 实训内容

(1) 复习教材或查资料,了解并熟悉 Linux 系统调用作用及工作原理；

(2) 熟悉编译内核的方法；

(3) 了解添加系统调用的方法；

(4) 添加系统调用,编程验证自己实现的结果；

(5) 总结该实训过程,写出实训报告。

3.4.5 为 Linux 添加一个内核模块

1. 实训目的

(1) 熟悉 Linux 内核模块的组成、注册及卸载；

(2) 掌握 Linux 内核模块的创建、编译、测试方法。

2. 实训内容

(1) 复习教材或查资料,了解并熟悉 Linux 动态可加载内核模块的概念、组成、注册、卸载及简要开发过程；

(2) 创建 Linux 内核模块；

(3) 编译 Linux 内核模块；

(4) 测试 Linux 内核模块；

(5) 总结该实训过程,写出实训报告。

3.4.6 Linux 中简单的字符设备驱动程序设计

1. 实验目的

(1) 理解 Linux 设备驱动程序的基本原理；

(2) 掌握 Linux 字符设备驱动程序的编写方法。

2. 实训内容

(1) 复习教材或查资料,了解设备文件、设备号、文件操作；

(2) 了解并熟悉 Linux 字符设备驱动程序的构成、注册、注销及简要开发过程；

(3) 编写一个简单的 Linux 字符设备驱动程序；

(4) 编译字符设备驱动程序；

(5) 加载并测试字符设备驱动程序；

- (6) 卸载字符设备驱动程序；
- (7) 总结该实训过程,写出实训报告。

3.5 源代码阅读级

这部分实训内容难度较大,但很有意义,也很有意思,建议参与者分组协作完成。

3.5.1 Linux 源代码专题分析——进程调度程序

1. 实训目的

- (1) 了解 Linux 源代码的分布；
- (2) 了解阅读 Linux 源代码的一般方法；
- (3) 熟悉 Linux 管理进程用的主要数据结构；
- (4) 通过阅读 Linux 进程调度有关函数的源代码,理解 Linux 的进程调度算法及其实现所用的主要数据结构；
- (5) 锻炼源代码阅读、分析能力和团队协作能力。

2. 实训内容

- (1) 通过查阅参考书或者上网找资料,熟悉/usr/src/linux(注意:这里最后一级目录名可能是一个含具体内核版本号和“linux”字符串的名字)下各子目录的内容,即所含 Linux 源代码的情况。
- (2) 在概览 Linux 启动和初始化部分源代码基础上,分析 Linux 进程调度有关函数的源代码,主要是 schedule()函数和 goodness()函数,并且要对它们引用的头文件等一并分析。
- (3) 归纳总结出 Linux 的进程调度算法及其实现所用的主要数据结构。

3.5.2 跟踪系统查找文件过程

1. 实训目的

- (1) 了解 Linux 源代码的分布；
- (2) 了解阅读 Linux 源代码的一般方法；
- (3) 熟悉 Linux 管理文件用的主要数据结构；
- (4) 通过分析 Linux 文件系统部分源代码,跟踪系统查找文件的过程。

2. 实训内容

- (1) 通过查阅参考书或者上网找资料,熟悉/usr/src/linux(注意:这里最后一级目录名可能是一个含具体内核版本号和“linux”字符串的名字)下各子目录的内容,即所含 Linux 源代码的情况。
- (2) 在概览 Linux 启动和初始化部分源代码基础上,分析 Linux 虚拟文件系统及 EXT3 文件系统的部分内核源代码,即其超级块、组描述符、数据块位图、索引节点表、目录项结构及其相关操作的源代码实现,并利用各种可能工具,跟踪、展示 Linux 文件系统管理模块搜索 EXT3 文件系统的/usr/include/stdio.h 文件的过程。
- (3) 归纳总结出 Linux 根据哪些主要数据结构先检索出文件/usr/include/stdio.h 的索引节点,进而读出其内容的全过程。

第4章 实训指导

本章内容提要：

- 4 个使用级实训指导；
- 8 个系统管理级实训指导；
- 8 个观察分析级实训指导；
- 6 个编程与实现级实训指导；
- 两个源码分析级实训指导。

内容使用建议

面向应用、难度适中的使用级、系统管理级、观察分析级实验和用户级 API 编程实验属于必做题，建议读者先按照第 3 章相应的实验目标及要求提前做好查阅资料 and 设计方案等方面的准备工作，然后，最好能先做实验，再看本章的指导内容。完成实验后，一定要认真思考总结，写好实验报告。

说明

考虑到本教程多数读者可能对 Windows 比较熟悉，故本章所有涉及 Windows 平台的实训指导内容均从简叙述，读者若有需求可参考其他资料，或者直接与作者联系。

4.1 使用级

4.1.1 安装 Linux

1. 实验说明

学习和动手安装 Linux 操作系统 CentOS 5.4，掌握操作系统的系统配置，了解建立操作系统应用环境的过程。

2. CentOS 简介

CentOS(Community Enterprise Operating System, 社区企业操作系统)是 Linux 发行版之一，它是来自于 Red Hat Enterprise Linux(RHEL)依照“开放源代码”规定发布的源代码编译而成。由于出自于同样的源代码，因此有些对稳定性要求高的服务器以 CentOS 来替代商业版本的 RHEL。两者的不同，主要在于 CentOS 不包含“封闭源代码”软件，同时移除了不能自由使用的 Red Hat 商标。CentOS 于 2014 年宣布与 Red Hat 合作，但 CentOS 将会在新的委员会下继续运作，并不受 RHEL 的影响。

CentOS 版本号有两个部分，一个主要版本号和一个次要版本号，分别对应于 RHEL 的主要版本与更新包，如 CentOS 5.4 对应于 RHEL 5.0 的更新第 4 版。截至 2014 年 12 月月底，CentOS 的最新版本是 CentOS 7.0。

CentOS 支持主流的 i386(Intel 32 位)、x86_64(AMD64 和 Intel64, 64 位)、PowerPC、龙芯(Loongson)等处理器，而在硬件要求上，CentOS 需要 256MB 内存，4GB 磁盘空间来安

装,因此大多数用户均可以在其个人计算机上安装 CentOS 版本。

3. CentOS 安装过程

不同类型及不同版本的 Linux 操作系统安装过程可能不一样,但基本的概念和安装过程类似,本节以从 Internet 下载的 CentOS-5. 4-i386-bin-DVD. iso 镜像为例,介绍 CentOS 5. 4 的安装过程,以供读者参考。

1) 安装前准备

(1) 准备安装光盘: 使用 UltraISO 等 ISO 管理工具刻录安装光盘,如读者已有安装光盘,可以跳过这一步。

(2) 设置光盘启动: 启动计算机,进入 BIOS,设置为光盘启动。

注意: 也可使用 USB 进行安装,需要制作 USB 安装盘,并在计算机启动项中设置相应启动项。使用 USB 安装过程和使用光盘安装基本一样,读者可以自行参考相关资料。

2) 安装步骤

光盘插入计算机,启动计算机,即可进入安装。下面简要介绍安装过程。

(1) 步骤 1——选择安装界面。看到如图 4-1 所示的安装光盘启动画面,直接按回车键,进入图形安装界面。



图 4-1 Linux 安装光盘启动界面

(2) 步骤 2——光盘检测。如图 4-2 所示界面询问是否进行光盘检测,选择 Skip 跳过。

(3) 步骤 3——选择安装语言。如图 4-3 所示界面询问安装过程使用的语言,默认为 English,建议选择简体中文。

(4) 步骤 4——选择键盘。如图 4-4 所示界面提示为系统选择键盘,保持默认即可。单击“下一步”按钮,开始进行硬盘分区和格式化。

(5) 步骤 5——硬盘分区及格式化。该步骤会进行硬盘分区及格式化,安装时,请仔细阅读提示,避免数据丢失,即使全部采用默认设置,也要做到心中有数。该步骤主要分为以下 5 个子步骤。



图 4-2 Linux 安装前提示是否进行光盘检测的界面



图 4-3 选择 Linux 安装过程使用语言的界面



图 4-4 系统选择键盘的界面

① 步骤 5.1——选择硬盘及是否修改分区方案。图 4-5 为安装程序对硬盘分区前提示选择硬盘及是否修改分区方案的界面,安装程序首先自动检测到计算机中的硬盘,本教程实训环境中为 8GB 的 SCSI 硬盘(注:如果使用 USB 安装,在其中也会出现 USB 盘,在后续的选择中,应避免选择对 USB 安装盘进行操作)。“检验和修改分区方案”复选框默认未勾选,如要自定义分区方案,请勾选该选项。单击“下一步”按钮,显示如图 4-6 所示的当前默认的分



图 4-5 选择硬盘及是否修改分区方案的界面



图 4-6 默认的分

② 步骤 5.2——自定义分区方案，删除默认分区方案中的逻辑卷。在如图 4-6 所示的当前默认的分區方案中，安装程序使用逻辑卷管理器(LVM)来管理系统中的硬盘，如对逻辑卷管理不熟悉，或不想使用逻辑卷，可选择逻辑卷组，如 VolGroup00，并单击“删除”按钮，删除逻辑卷，删除后的分区方案如图 4-7 所示。



图 4-7 删除默认分区方案中逻辑卷后的分区方案

③ 步骤 5.3——继续自定义分区方案，再删除默认分区所在的硬盘。在如图 4-7 所示的删除默认分区方案中逻辑卷后的分区方案里，安装程序将硬盘默认划分成了两个分区，如想继续自定义，选择硬盘驱动器/dev/sda，继续删除，则该硬盘上的分区方案如图 4-8 所示。



图 4-8 删除默认分区方案中逻辑卷及硬盘后的分区方案

④ 步骤 5.4——继续自定义分区方案,在删除了所有默认分区的硬盘上(如图 4-8 所示),选择需要创建分区空间,使用上方“新建”、“编辑”、“删除”等按钮,用户可以根据自己的要求自行定义分区方案。自定义的分区方案至少有一个独立的根分区,至多有 5 个独立的分区,通常需要对可创建的 5 种类型的分区进行选取考虑。a. /分区: 根分区,必要。挂载点是根目录“/”。b. swap 分区: 交换分区,可选,建议创建。在虚拟内存管理中,需要进行内存和外存的数据交换,使用独立的交换分区可提高数据交换效率;交换分区的大小一般为系统内存大小的 2 倍左右。c. boot 分区: 启动分区,可选。挂载点为/boot,用于存放系统启动文件。独立的启动分区可在一定程度上避免对启动文件的误操作所导致的系统不能正常启动的现象发生;boot 分区不必太大,一般在 100~150MB。d. var 分区: 可选,挂载点为/var。/var 目录一般用于系统运行过程中日志记录,如果系统在运行过程中产生大量的日志信息,建议独立 var 分区,以免因日志信息大量占用根分区空间而使系统崩溃。e. usr/local: 可选,挂载点为/usr/local。/usr/local 目录一般用于用户应用程序的安装及使用。设立独立的/usr/local 分区可以避免用户程序破坏根分区。作为样例,本教程实训环境的自定义分区方案只包含一个根分区和一个交换分区,具体操作步骤为:在图 4-8 中,选择空闲空间,单击“新建”按钮,可以添加分区,如图 4-9 所示。



图 4-9 自定义分区方案,添加分区

⑤ 步骤 5.5——继续自定义分区方案,在选择需要创建分区空间中,相继添加一个根分区和一个交换分区,具体操作步骤为:在弹出的“添加分区”对话框(见图 4-9)中,依次选择或指定所添加分区的挂载点(指示分区挂载到哪个目录)、文件系统类型(指示分区文件系统类型,一般选择 ext3;但对于交换分区,则要选择 swap,而且不必选择挂载点)和大小等,结果如图 4-10 所示(这里交换分区的大小设为内存大小的 2 倍。剩余空间都给了根分区)。

⑥ 步骤 6——选择 GRUB 多引导装载程序的安装位置。如图 4-11 所示界面询问是否安装 GRUB 多引导装载程序,若机器上还装有 Windows 等其他操作系统,则应在此选择