Windows & Linux

操作系统实验教程

吴国伟　主编

大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

操作系统实验教程

主　编　吴国伟
副主编　张　杰　刘梓红　徐子川
范庆娜　赵修伟　任广臣

大连理工大学出版社

图书在版编目(CIP)数据

操作系统实验教程/吴国伟主编. —大连:大连理工大学出版社,2007.11
ISBN 978-7-5611-3818-2

Ⅰ.操… Ⅱ.吴… 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(2007)第 174318 号

大连理工大学出版社出版
地址:大连市软件园路 80 号　邮政编码:116023
电话:0411-84708842　邮购:0411-84703636　传真:0411-84701466
E-mail:dutp@dutp.cn　URL:http://www.dutp.cn
大连理工印刷有限公司印刷　　大连理工大学出版社发行

幅面尺寸:185mm×260mm　印张:10.75　字数:243 千字
附件:光盘一张　印数:1～2500
2007 年 11 月第 1 版　2007 年 11 月第 1 次印刷

责任编辑:高智银　责任校对:达　理
封面设计:宋　蕾

ISBN 978-7-5611-3818-2　定　价:22.00 元

前　言

操作系统是计算机科学中的重要基础课程，它在计算机系统中所处的位置使得它成为计算机类专业中重要的专业基础课程之一。操作系统是概念多、涉及面广、难度较大、抽象性较高的一门课程，一直以来，在这门课程的教学中存在着一个非常大的难题：即如何才能将抽象的知识、理论以及繁多的技术形象化地传授给学生，使学生能深入浅出地掌握相关知识并能实际体验和应用。本书从实践上下手，将目前广泛应用的 Windows 和 Linux 操作系统引进来，穿插到各个章节中。结合 Windows 和 Linux 实例设计，让学生用所学过的语言工具去实践，将理论知识的学习和实践紧密地结合在一起，有助于学生对所学内容的深入理解，而且锻炼了他们的实际动手能力，达到不仅知其然，知其所以然，还要学会其如何然的目的。

本书分为两大部分实验内容，分别为 Windows 实验内容和 Linux 实验内容，其中 Windows 实验内容包括 13 个实验内容，以 Visual C++ 6.0 为平台编程实现操作系统的各个部分原理，包括进程创建、多线程创建、进程间通信、进程间互斥、死锁处理、安全机制等。Linux 实验内容包括 12 个内容，以 gcc 编译器和 vi 编辑器为开发工具，完成 Linux 中进程创建、进程间通信、驱动设计、定时器、模块设计等。本书配备了 Windows 实验部分的全部源代码，Linux 部分的程序为了锻炼使用者对 vi 编辑器的使用，没有给出。

本书编写过程中得到了大连理工大学出版社吕志军和高智银先生的大力支持，非常感谢高智银先生认真细致和兢兢业业的工作态度，没有他的帮助，就没有本书高质量的出版。也非常感谢我的研究生徐子川、范庆娜、赵修伟、

任广臣，他们为本书做了很多编辑和源程序的开发工作，在此对他们表示感谢。也感谢辽宁师范大学的唐棣教授，对本书提出了中肯的意见，我的同事覃振权博士也对本书提出了很多意见，一并表示感谢。

本书可作为计算机科学与技术、软件工程或相关专业的操作系统实验教材，也可作为从事嵌入式软件和系统软件设计等科技人员自学或参考用书。

编者

2007 年 10 月

目 录

Windows 平台

Linux 平台

Windows 平台

实验一　基本操作命令实践

一、实验内容

操作实践最常见的操作系统管理的一些命令。

二、实验目的

配合操作系统课程的学习，加深操作系统提供命令接口的功能的理解。

三、实验要求

在操作系统 Windows XP 或 2000 及 Windows Server 2003 下，操作以下命令，各个命令的作用解释在每条命令后都已经给出。

实践命令：

1. 最基本、最常用的测试物理网络连通性的命令 ping，ping 命令用于确定本地主机是否与网络上另外一台主机连通。

格式为：ping IP 地址 命令选项

例如：ping 192. 168. 0. 8 -t -l 2048 -n 8

解释：-t 表示一直发包，直到被用户用 Ctrl+C 中断为止。

-l 表示发送的数据包的大小。本例中为 2048。

-n 表示发几个就停止发包。本例中为 8 个。

2. 查看 DNS、IP 地址、MAC 地址的命令 ipconfig。

格式为：ipconfig 命令选项

例如：ipconfig /all

解释：/all 选项会看到 Windows IP Configuration 和 Ethernet Adapter 信息，包括 IP 地址、DNS 及 MAC 地址。请注意“ipconfig /all”是获得本机网卡的物理地址的最常用的方法。

其他选项请同学们自己摸索。查看其他选项可以使用如下命令：ipconfig /? 来查看。

3. 查看 DNS 与 IP 地址命令 nslookup，它主要用于解析一个域名所对应的 IP 地址。

格式为：nslookup 主机域名

例如：nslookup

解释：查看缺省 DNS 服务器及其 IP 地址。

又例如：nslookup ssdut. dlut. edu. cn

解释：返回该域名所对应的 IP 地址。

4. 网络信使命令 net send。

格式为：net send 计算机名/IP 地址 传播的消息(注意不能跨网段)

例如：net send 210. 30. 103. 121 hello

解释：发送消息“hello”给 IP 地址为 210. 30. 103. 121 的机器，如图 1-1 所示。

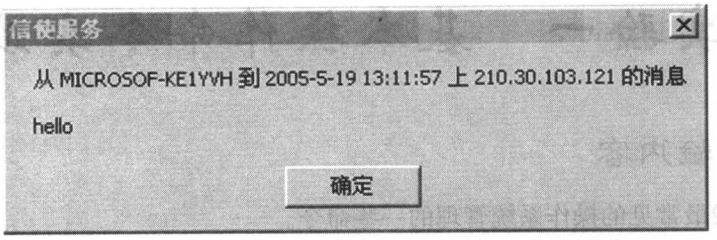

图 1-1　网络信使命令

另外，停止网络信使服务的命令为 net stop messenger，开始网络信使服务的命令为 net start messenger。

5. 探测对方计算机名、所在的组、域及当前用户名的命令 nbtstat

格式为：nbtstat -a IP 地址

例如：nbtstat -a 192. 168. 0. 6

解释：探测 192. 168. 0. 6 计算机名、所在的组、域及当前用户名。

另外也可用 ping -a IP 地址 -t 来获得，请同学们自己实践。

6. 显示计算机当前所开放的所有端口的命令 netstat。

格式为：netstat -a

例如：netstat -a

解释：显示计算机当前所开放的所有端口。

另外 netstat -s -e 会显示比较详细的网络资料，包括 TCP、UDP、ICMP、IP 的统计。

7. 探测 ARP 绑定列表命令 arp。

格式为：arp 选项

例如：arp -a

解释：显示了所有连接了本地计算机的对方计算机的 IP 和 MAC 地址。

例如：arp -s 192. 168. 10. 59　　00-50-FF-6C-08-75

解释：捆绑 IP 和 MAC 地址，解决局域网内盗用 IP。

例如：arp -d 00-50-FF-6C-08-75 192. 168. 10. 59

解释：解除网卡的 MAC 地址与 IP 的绑定。

8. net 系列命令的使用。包括 net view、net user、net use、net session、net share 等。

格式：net 命令

例如：net view IP 地址

解释：显示当前工作组服务器列表，当不带选项使用本命令时，它就会显示当前域或网络上的计算机上的列表。

例如：net user

解释：查看计算机上的用户帐号列表。

例如：net use z: //192. 168. 10. 8/movie

解释：将这个 192.168.10.8 的 movie 共享目录映射为本地的 Z 盘。

例如：net session

解释：记录链接情况，包括计算机、用户名、客户类型和打开空闲时间。

例如：net share

解释：查看机器的共享资源。

例如：net share c$ /d

net share d$ /d

net share ipc$ /d

net share admin$ /d

解释：手工删除共享。

例如：net share mymovie=e:\downloads\movie /users:1

解释：增加一个共享 mymovie。同时限制链接用户数为 1 人。

9. 路由跟踪命令 tracert 及 pathping。

格式为：tracert 选项 或 pathping 选项

例如：tracert www.sina.com.cn 或 pathping www.sina.com.cn

解释：显示了连接 www.sina.com.cn 所经过的路由。

10. 设置开机密码命令 syskey。

格式为：syskey

解释：设置开机密码。

11. 远程桌面连接命令 mstsc。

格式为：mstsc

解释：调出建立远程桌面连接的对话框，如图 1-2 所示。

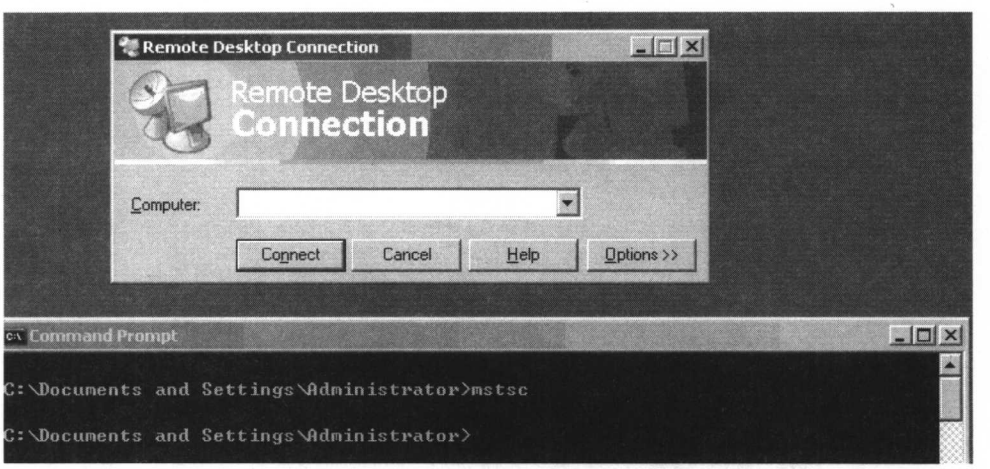

图 1-2 远程桌面连接

12. 本地服务设置命令 services.msc。

格式：services.msc

解释：调出服务设置窗口，对服务进行相应的设置，如启动服务、终止服务等。如图1-3所示。

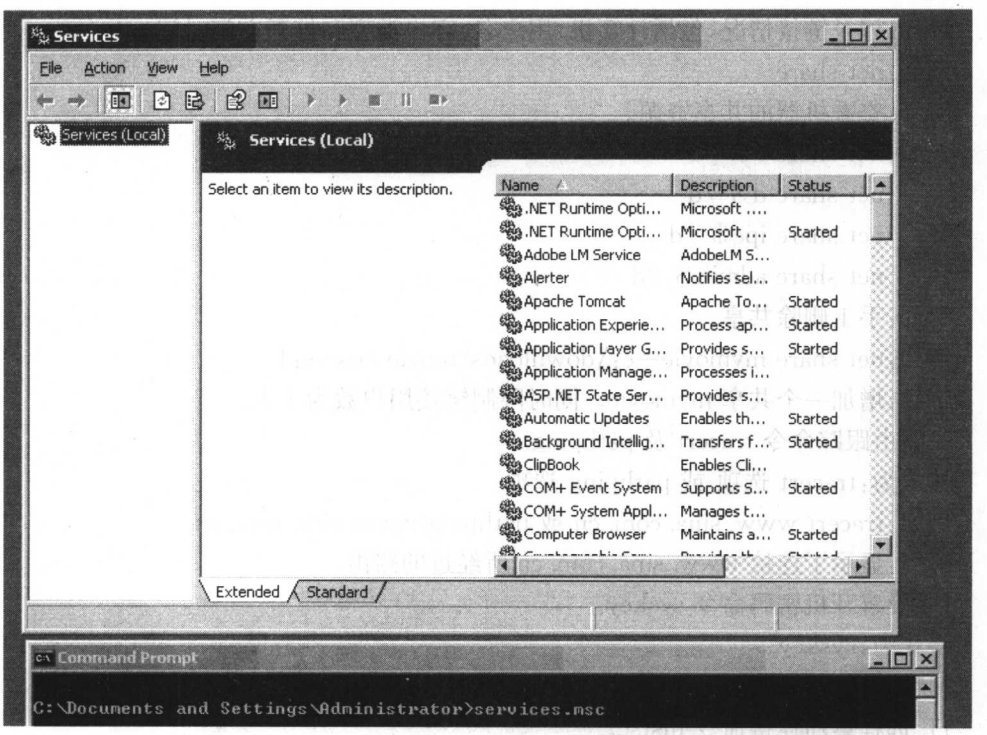

图1-3　本地服务设置

实验二　进程控制实践

一、实验内容

使用 MFC 提供的 CreateProcess 函数创建进程，并给出系统的进程的基本信息，如 ID 号，使用 TerminateProcess 函数终止所创建进程。

二、实验目的

配合操作系统课程的学习，加深对进程的控制与描述的理解并熟悉 VC 的使用。

三、实验原理

操作系统的一个主要职责就是控制进程的执行，从进程的产生和终止来理解这次实验的原理。

1. 进程的产生

当有一个新进程要加入当前进程的队列的时候，操作系统就产生一个控制进程的数据结构并且为该进程分配地址空间。这样，新进程就产生了，在 Windows 中这个工作主要由 CreateProcess 函数完成。

通常有以下四种事件会导致新进程的产生：

- 在终端输入命令。
- 在批处理环境中为了响应一个任务的要求。
- 获取到用户程序提出的要求后，OS 可以代用户程序产生进程以实现某种功能。
- 基于应用进程的需要。

2. 进程的终止

使进程终止的条件很多，比如正常终止、超时限制、内存不足、超界等。

3. 相关函数说明

- CreateProcess 函数

当一个线程调用 CreateProcess 时，系统就会创建一个进程内核对象，该进程内核对象不是进程本身，而是操作系统管理进程的一个较小的数据结构。可以将进程内核对象视为由进程的统计信息组成的一个较小的数据结构。然后，系统为新进程创建一个虚拟地址空间，并将可执行文件或任何必要的 DLL 文件的代码和数据加载到该进程的地址空间中。

- TerminateProcess 函数

只有当无法用另一种方法来迫使进程退出时，才应该使用 TerminateProcess。该函数是一个异步运行的函数，也就是说，它会告诉系统，你想要进程终止运行，但是当函数返

回的时候，你无法保证该进程已经终止运行。

四、实验要求

程序需要给出四个 Button 控件，ID、Caption 分别为 IDC_CREATPRO_BUTTON、“创建进程”，IDC_GETPROCESS、“获取进程”，IDC_KILLPROCESS、“杀死进程”，IDCANCEL、“退出”，一个 ID 为 IDC_LIST1 的 List Control 控件(如图 2-1 所示)。

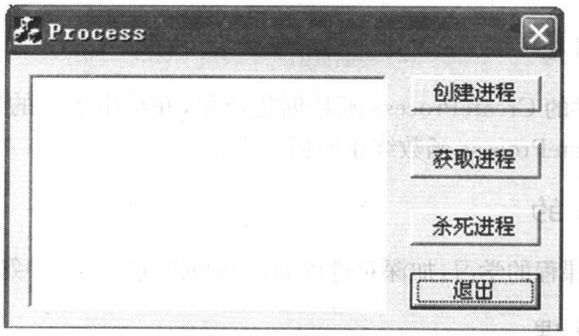

图 2-1　进程控制

五、实验步骤

1. 打开 VC，创建一个名为 Process 的工程(如图 2-2 所示)。

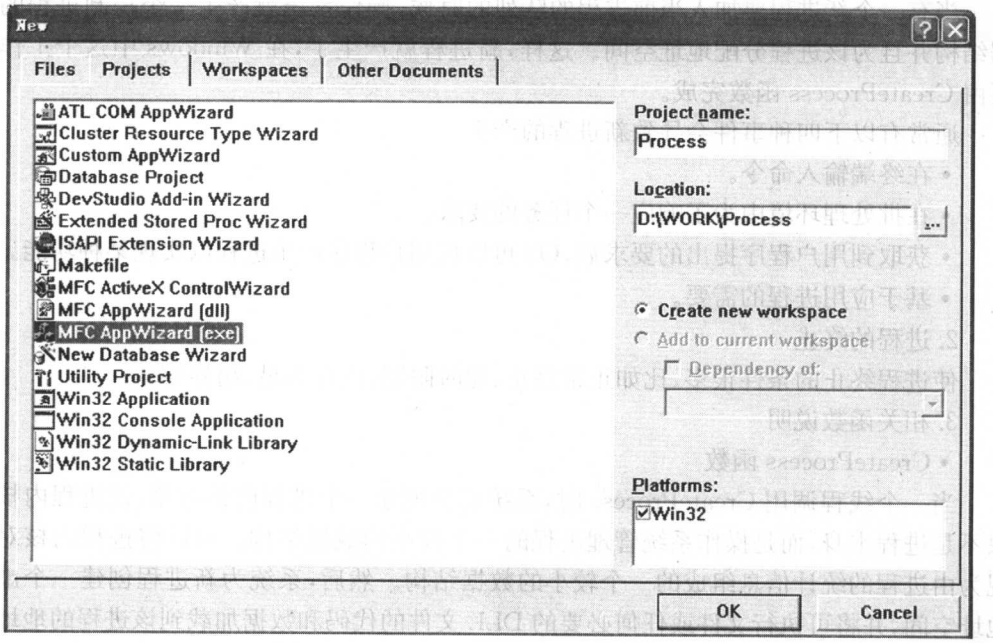

图 2-2　创建工程

下一步选择基于对话框类型(如图 2-3 所示)。

2. 把控件拖到对话框上，布局见图 2-1。

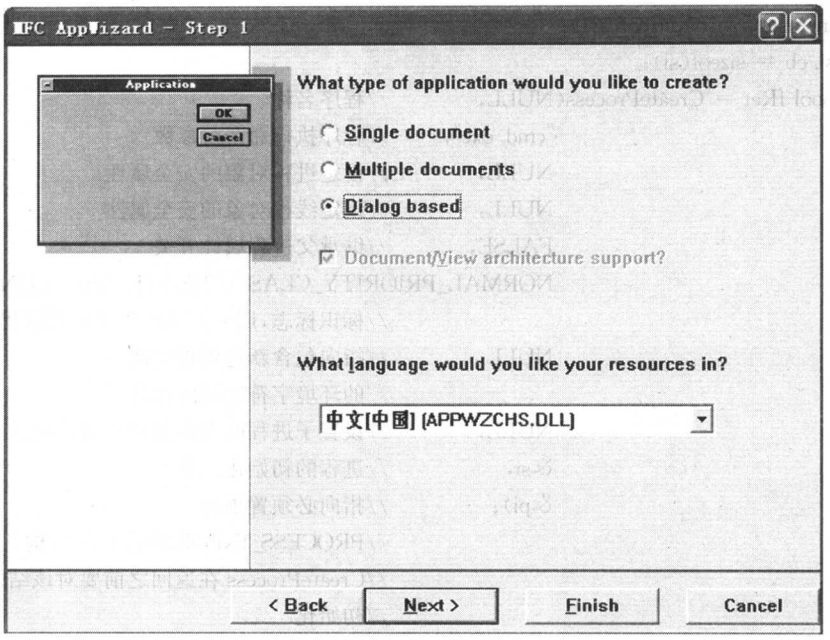

图 2-3 工程类型

3. 右击 List Control 控件，选择 ClassWizard…（如图 2-4 所示），为 IDC_LIST1 添加类型为 CListCtrl 的成员变量 m_list。

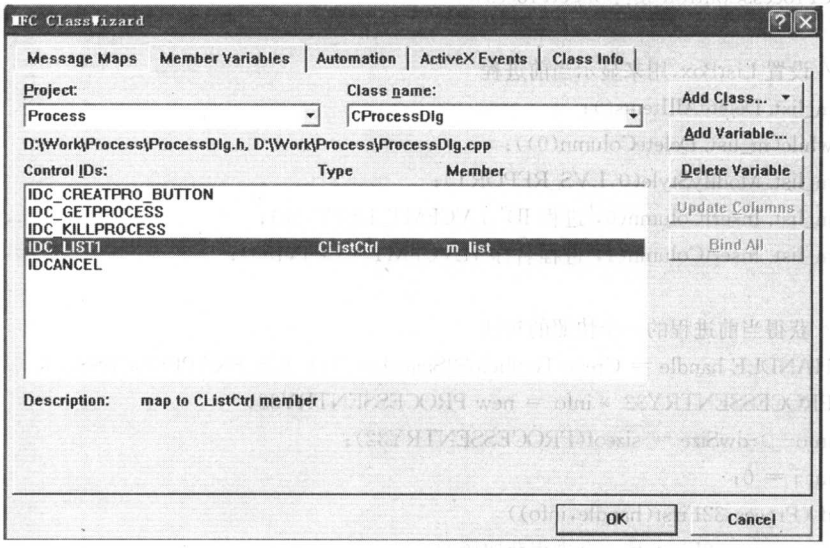

图 2-4 添加成员变量

4. 双击“创建进程”按钮，添加如下代码：

```
void CProcessControlDlg::OnCreateproButton()
{
    PROCESS_INFORMATION pi;
    STARTUPINFO si;
```

```
    memset(&si,0,sizeof(si));
    si.cb = sizeof(si);
    bool fRet = CreateProcess(NULL,          //程序名称
                              "cmd.exe",     //程序执行命令行参数
                              NULL,          //设定进程对象的安全属性
                              NULL,          //设定线程对象的安全属性
                              FALSE,         //继承父进程属性开关
                              NORMAL_PRIORITY_CLASS|CREATE_NEW_CONSOLE,
                                             //标识标志,用来表示如何来创建进程
                              NULL,          //指向包含新进程将要使用
                                             //的环境字符串的内存块
                              NULL,          //设置子进程的当前驱动器和目录
                              &si,           //进程的初始化信息
                              &pi);          //指向必须置顶的
                                             //PROCESS_INFORMATION 结构,
                                             //CreateProcess 在返回之前要对该结构进行
                                             //初始化
}
```

5. 双击“获取进程”按钮,添加如下代码:

```
void CProcessControlDlg::OngetProcess()
{
    //设置 ListBox,用来显示当前进程
    m_list.DeleteAllItems();
    while(m_list.DeleteColumn(0));
    m_list.ModifyStyle(0,LVS_REPORT);
    m_list.InsertColumn(0,"进程 ID",LVCFMT_LEFT,80);
    m_list.InsertColumn(1,"进程名称",LVCFMT_LEFT,150);

    //获得当前进程的一个快照的句柄
    HANDLE handle = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);
    PROCESSENTRY32 * info = new PROCESSENTRY32;
    info->dwSize = sizeof(PROCESSENTRY32);
    int i = 0;
    if (Process32First(handle,info))
    /* process32First 是一个进程获取函数,
    当我们利用函数 CreateToolhelp32Snapshot()获得当前运行进程
    的快照后,我们可以利用 process32First 函数来获得第一个进程的
    句柄,其原型为(用的是 vfp):
    DECLARE INTEGER Process32First IN WIN32API ;
    INTEGER hSnapshot,STRING @ lppe
    在 C 语言中如下:
```

```
    BOOL WINAPI Process32First(HANDLE hSnapshot, LPPROCESSENTRY32 lppe);
    */
    {
        if (GetLastError()== ERROR_NO_MORE_FILES)
        {
            AfxMessageBox("No more process");
        }
        else
        {
            CString id;
            id.Format("%d",info->th32ProcessID);
            m_list.InsertItem(i,id);
            m_list.SetItemData(i,info->th32ProcessID);
            id.Format("%s",info->szExeFile);
            m_list.SetItemText(i,1,id);
            i++;
            while (Process32Next(handle,info)! =FALSE)
            //与 Process32First 类似,获得下一个进程信息
            {
                //将进程信息显示到 ListBox
                id.Format("%5d",info->th32ProcessID);
                m_list.InsertItem(i,id);
                m_list.SetItemData(i,info->th32ProcessID);
                id.Format("%s",info->szExeFile);
                m_list.SetItemText(i,1,id);
                i ++;
            }
        }
    }
    CloseHandle(handle);
}
```

6. 双击“杀死进程”按钮,添加如下代码:

```
void CProcessControlDlg::OnkillProcess()
{
    POSITION pos;
    pos = m_list.GetFirstSelectedItemPosition();
    int select = m_list.GetNextSelectedItem(pos);
    //获得选中进程的句柄
    HANDLE h = OpenProcess(PROCESS_ALL_ACCESS, TRUE, m_list.GetItemData(se-
    lect));
    if (h ! = NULL)
    {
```