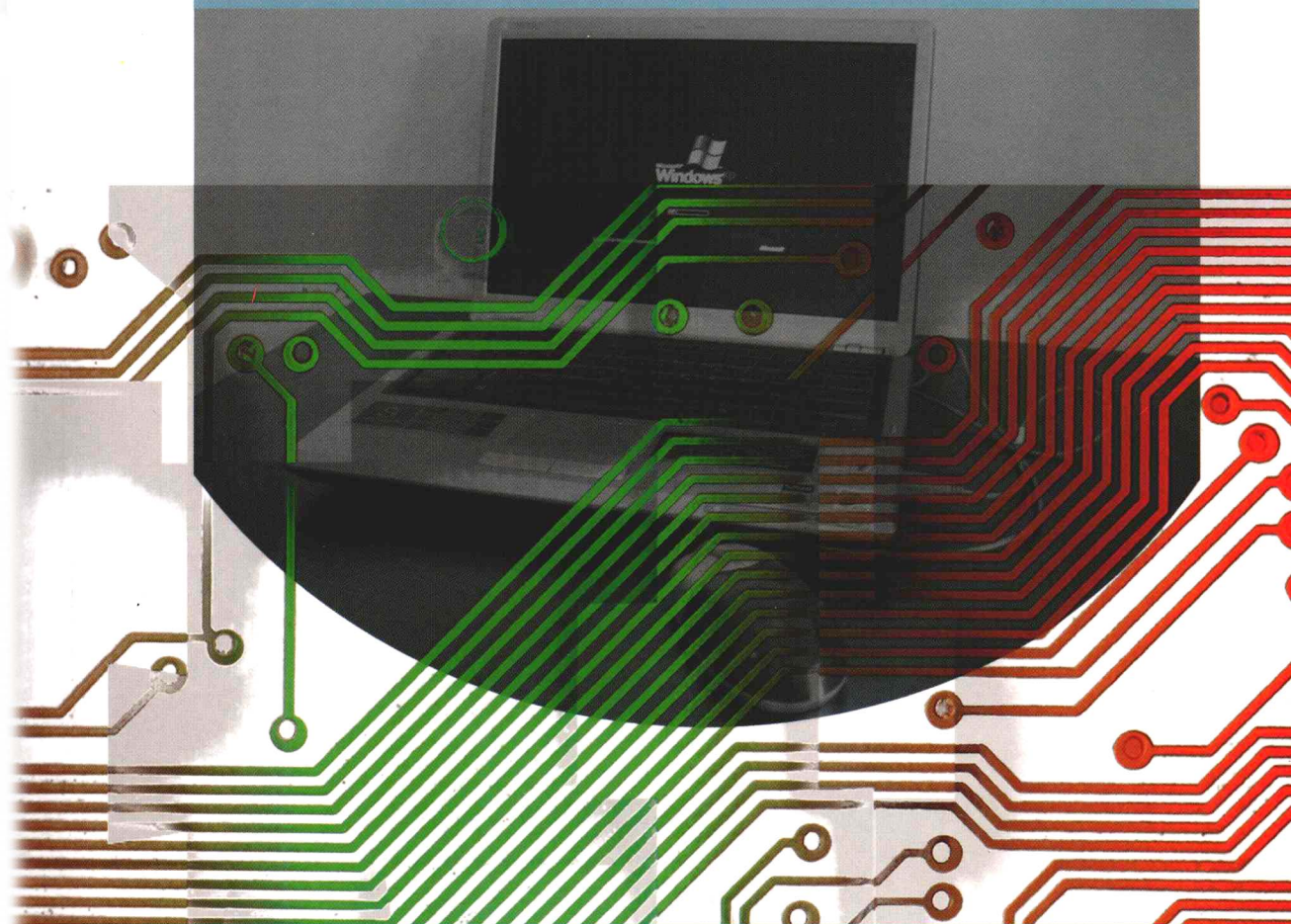



操作系统

(本科)

实验指南

张茂林 孟庆昌 编



 中央广播电视大学出版社

操作系统（本科）实验指南

张茂林 孟庆昌 编

中央广播电视大学出版社
北 京

图书在版编目 (CIP) 数据

操作系统 (本科) 实验指南/张茂林, 孟庆昌编. —北京: 中央广播电视大学出版社, 2008.5

ISBN 978-7-304-04073-4

I. 操… II. ①张…②孟… III. 操作系统—高等学校—教学参考资料 IV. TP316

中国版本图书馆 CIP 数据核字 (2008) 第 075302 号

版权所有, 翻印必究。

操作系统 (本科) 实验指南

张茂林 孟庆昌 编

出版·发行: 中央广播电视大学出版社

电话: 发行部 010-58840200

总编室 010-68182524

网址: <http://www.crtvup.com.cn>

地址: 北京市海淀区西四环中路 45 号 邮编: 100039

经销: 新华书店北京发行所

策划编辑: 何勇军

版式设计: 韩建冬

责任编辑: 申敏

责任校对: 王亚

责任印制: 赵联生

印刷: 北京博图彩色印刷有限公司

印数: 0001~5000

版本: 2008 年 6 月第 1 版

2008 年 6 月第 1 次印刷

开本: 787×1092 1/16

印张: 4.5 字数: 108 千字

书号: ISBN 978-7-304-04073-4

定价: 9.00 元

(如有缺页或倒装, 本社负责退换)

中央广播电视大学开放教育
计算机科学与技术系列规划教材编委会

顾 问：葛道凯

主 任：李林曙 于二元

副主任：陶水龙 杨孝堂 刘 臣

成 员：李伟生 胡 俊 崔 林 郭 鸿

袁 薇 何晓新 王春风 顾静相

前 言

操作系统是各大专院校计算机科学与技术专业的一门必修课程，又是一门理论和实践结合十分密切的课程。在教学过程中，除了讲授基本理论外，还必须重视教学实验环节的设计，使学生能够通过教学实验增强对操作系统的感性认识，加深对操作系统理论知识的理解，达到进一步巩固学习成果的目的。

本书是《操作系统（本科）》的配套实验教材，全书分为3个部分。第1部分基础知识分为两章：第1章介绍Linux系统使用的基本知识；第2章介绍shell程序设计的相关内容。第2部分实验指南是本书的主体部分，共分为5章，分别介绍了Linux系统安装、Linux系统使用及shell编程、进程管理、存储管理、文件管理5个实验。第3部分附录为一些实验相关参考资料，包括Linux常用命令列表和vi编辑器使用的详细指导等内容，供教师和学生参考。

本书由北京航空航天大学张茂林副教授、北京信息科技大学孟庆昌教授编写。孟庆昌编写第1部分，张茂林编写第2、3部分。感谢袁薇对本书的编写和出版给予的热情支持和帮助。

本书可作为大学本科计算机科学与技术专业的教科书，也可作为其他相关专业或成人教育的参考书，以及计算机工作者的自学用书。

由于编者水平有限，书中难免有不妥甚至错误之处，希望广大读者给予批评指正。

编 者

2008年4月于北京

目 录

第 1 部分 基础知识	(1)
1 Linux 使用	(1)
1.1 Linux 简单命令	(1)
1.2 Linux 命令格式	(2)
1.3 文件操作	(3)
1.4 目录操作	(4)
1.5 文件存取权限	(7)
2 shell 程序设计	(9)
2.1 shell 概述	(9)
2.2 shell 变量和算术运算	(10)
2.3 shell 特殊字符	(15)
2.4 shell 程序控制结构	(18)
2.5 shell 函数和内置命令	(25)
第 2 部分 实验指南	(27)
3 Linux 系统安装实验	(27)
3.1 实验目的和实验要求	(27)
3.2 实验步骤	(27)
4 Linux 使用及 shell 编程实验	(29)
4.1 实验目的和实验要求	(29)
4.2 实验步骤	(30)
5 进程管理实验	(36)
5.1 实验目的和实验要求	(36)
5.2 实验步骤	(36)
6 存储管理实验	(42)
6.1 实验目的和实验要求	(42)
6.2 实验步骤	(43)
7 文件管理实验	(46)
7.1 实验目的和实验要求	(46)

7.2 实验步骤	(46)
第3部分 附录	(51)
附录 A Linux 常用命令	(51)
A1 登录和退出系统	(51)
A2 文件操作命令	(51)
A3 目录操作命令	(54)
A4 信息显示命令	(55)
A5 进程管理命令	(56)
A6 程序编译运行	(56)
附录 B vi 编辑器的使用	(57)
B1 进入和退出 vi	(57)
B2 光标移动	(57)
B3 正文输入	(59)
B4 正文检索和替换	(60)
B5 正文移动和复制	(61)
B6 重复和取消操作	(62)
B7 其他命令	(62)
参考文献	(63)

第 1 部分 基础知识

1 Linux 使用

Linux 系统提供的命令需要在 shell 环境下运行。Linux 系统中默认的 shell 是 bash，它提供了几百条系统命令。虽然这些命令的功能不同，但它们的使用方式和规则都是统一的。

在桌面环境下，可以利用终端程序进入传统的命令行操作界面，进入方式如下：在“开始”菜单中选择“应用程序→附件→系统终端”。Konsole 终端程序的窗口如图 1-1 所示。要退出终端程序，可以单击窗口右上角的“关闭”按钮，或在 shell 提示符下执行 exit 命令，也可按快捷键 [Ctrl-D]。

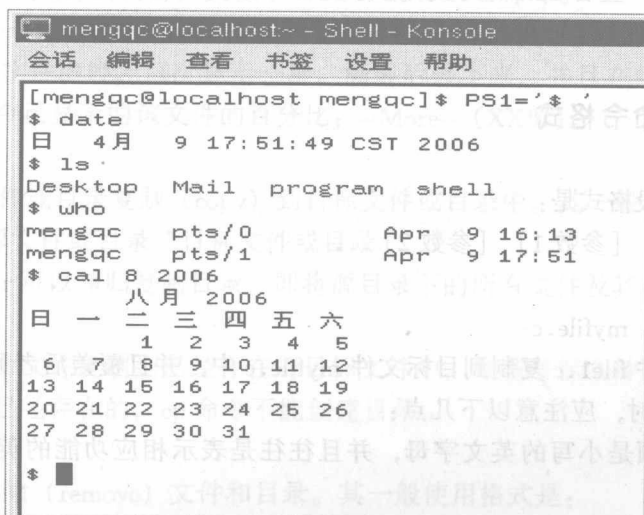


图 1-1 Konsole 终端程序的窗口

1.1 Linux 简单命令

下面列出几个常用的简单命令，包括使用格式和说明。

1. who 命令

who 命令列出所有正在使用系统的用户、所用终端名和注册到系统的时间。而 who am i 命令将列出使用该命令的当前用户的相关信息。

2. echo 命令

echo 命令可以将命令行中的参数显示到标准输出（即屏幕）上。

3. date 命令

date 命令在屏幕上显示或设置系统的日期和时间。如果没有选项和参数，将直接显示系统当前的日期和时间。

4. cal 命令

cal 命令可以显示公元 1—9999 年中任意一年或者任意一个月的日历。如果使用该命令时不带任何参数，则显示当前月份的日历。如果在 cal 命令后只有一个参数，则该参数被解释为年份，而不是月份。当在 cal 命令后有两个参数时，则第一个参数表示月份，第二个参数表示年份。

5. clear 命令

clear 命令清除屏幕上的信息，它类似于 DOS 中的 CLS 命令。

6. man 命令

Linux 系统中有大量的命令，而且许多命令又有众多选项或参数，要想全部记住它们相当困难。对大多数用户来说，也没有必要这样做，因为用户常用的命令是整个命令集合中的一个子集。幸运的是，Linux 提供了很有用的联机帮助命令 man，利用它可以方便地查看所有命令的完整说明，包括命令语法、各选项的意义以及相关命令等。例如：

```
$ man date    查看 date 命令的用法。
```

```
$ man grep    查看 grep 命令的用法。
```

1.2 Linux 命令格式

bash 命令的一般格式是：

```
命令名 [选项] [参数 1] [参数 2] ...
```

例如：

```
cp -f file1.c myfile.c
```

该命令将源文件 file1.c 复制到目标文件 myfile.c 中，并且覆盖后者原有内容。

使用 bash 命令时，应注意以下几点：

(1) 命令名必须是小写的英文字母，并且往往是表示相应功能的英文单词或单词的缩写。

(2) 由方括号括起来的部分是可选的，即可有可无，依具体情况而定。

(3) 选项以“-”开始，多个选项可用“-”连起来，如 ls -l -a 与 ls -la 相同。

(4) 在命令、选项和参数之间要用空格隔开。连续的空格会被 shell 解释为单个空格。

(5) 命令行的参数提供命令运行的信息或者命令执行过程中所使用的文件名。通常，参数是一些文件名，告诉命令从哪里可以得到输入，以及把输出送到什么地方。

(6) 如果命令行中没有提供参数，则命令将从标准输入文件（即键盘）上接收数据，输出结果显示在标准输出文件（即显示器）上，而错误信息则显示在标准错误输出文件（即显示器）上。可使用重定向功能对这些文件进行重定向。

(7) 命令在正常执行后返回一个 0 值，表示执行成功；如果命令在执行过程中出错，没有完成全部工作，则返回一个非 0 值（在 shell 中可用变量 \$? 查看）。在 shell 脚本中可用此

返回值作为控制逻辑流程的一部分。

另外，Linux 操作系统的联机帮助对每个命令的准确语法都作了说明。

1.3 文件操作

1. Linux 文件类型

Linux 操作系统支持以下文件类型：普通文件、目录文件、特殊文件。

(1) 普通文件：也称为常规文件，如文本文件、数据文件、可执行的二进制程序。

(2) 目录文件：类似于 Windows 系统中的文件夹，其中包含许多普通文件和子目录。

(3) 特殊文件：特指各种外部设备的文件，为了便于统一管理，Linux 把所有 I/O 设备都按文件格式供用户使用。特殊文件分为字符特殊文件（如终端、打印机）和块特殊文件（如硬盘、软盘和磁带）。

2. 常用文件操作命令

Linux 常用文件操作命令包括查看文件的内容、复制文件、删除文件、移动文件等。（注意：下面给出一些命令的功能，在附录 A 中列出 Linux 常用命令的较详细的说明。）

(1) cat 命令

cat 命令用来显示文件的内容，类似于 DOS 下的 TYPE 命令。

(2) more 命令

more 命令显示文件内容，每次显示一屏。满屏后停下来，并且在屏幕的底部出现一个提示信息，给出至今已显示的该文件的百分比：--More-- (XX%)。

(3) cp 命令

cp 命令将源文件或目录复制 (copy) 到目标文件或目录中。其一般使用格式是：

cp [选项] 源文件或目录 目标文件或目录

利用选项 -R, -r 可以递归复制目录，即将源目录下的所有文件及其各级子目录都复制到目标位置。

注意：若把一个文件复制到已经存在的目标文件中，则该目标文件的内容将被破坏。所有目标目录必须是已经存在的，cp 命令不能创建目录。

(4) rm 命令

rm 命令可以删除 (remove) 文件和目录。其一般使用格式是：

rm [选项] 文件列表

应注意：使用 rm 命令要格外小心。因为一旦删除了一个文件，就无法再恢复它。所以，在删除文件之前，最好再看一下文件的内容，确定是否真要删除。

(5) mv 命令

mv 命令用来对文件或目录重新命名，或者将文件从一个目录移到 (move) 另一个目录中。如果目标文件已经存在，则目标文件的内容将被覆盖。其一般使用格式是：

mv [选项] 源文件或目录 目标文件或目录

(6) wc 命令

wc 命令用来统计指定文件的字节数、字数、行数，并将统计结果显示出来，并且总是按以下格式显示，每项只占一列：

行数 字数 字节数 文件名

(7) pr 命令

pr 命令按给定格式输出文件。通常，在不带选项的情况下，会输出指定文件的内容，并在开头加上标题行：日期、时间、页码。

(8) grep 命令

grep 命令用来在文本文件中查找指定模式的词或短语，并在标准输出上显示包括给定字符串模式的所有行。其一般使用格式是：

```
grep [选项] 查找模式 [文件名 1, 文件名 2, ...]
```

例如，在密码文件/etc/passwd 中查找包含“mengqc”的所有行：

```
$ grep "mengqc" /etc/passwd
```

```
mengqc:x:500:100:mengqc:/home/mengqc:/bin/bash
```

grep 命令在指定文件中搜索特定模式及定位特定主题等方面用途很大。要搜索的模式被看做是一些关键词，查看指定的文件中是否包含这些关键词。如果没有指定文件，它们就从标准输入中读取。在正常情况下，每个匹配的行被显示到标准输出上。如果要搜索的文件不止一个，则在每一行输出之前加上文件名。

1.4 目录操作

1. 目录结构

(1) Linux 文件系统采用树状目录结构，即只有一个根目录（通常用“/”表示），其中含有下级子目录或文件的信息，子目录中又可含有更下级的子目录或者文件的信息……这样一层一层地延伸下去，构成一棵倒置的树，如图 1-2 所示。

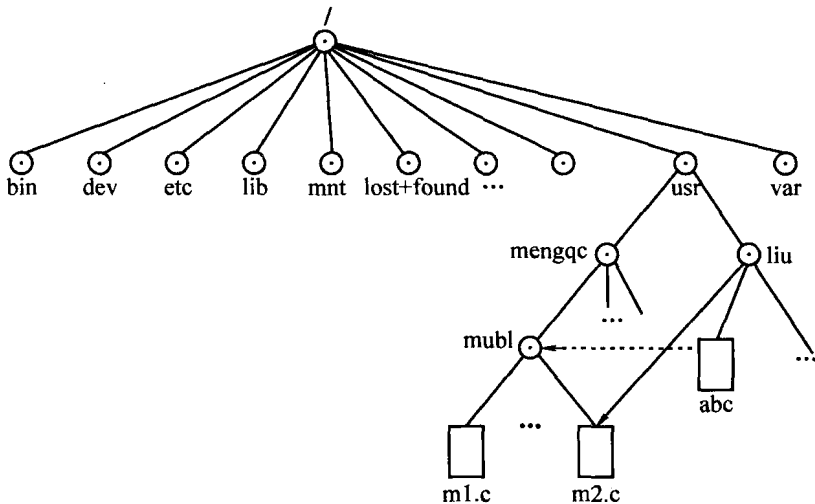


图 1-2 Linux 树状目录结构

在目录树中，根节点和中间节点（用圆圈表示）都必须都是目录，而文件只能作为“叶子”出现。当然，目录也可以是叶子。在 Linux 系统中，除根目录（root）外，所有文件和目录都包含在相应的目录文件中。

用户登录进入系统时，其主目录就是用户当前工作目录。主目录往往位于/home 或者 /usr 目录之下，并且与注册名相同，如 /usr/mengqc。

(2) 为了访问文件，必须告诉系统：它们在什么地方，即文件保存在哪个目录下。路径名描述了文件系统中从某个节点通向指定文件的路径。有两种路径名：绝对路径名和相对路径名。

在 Linux 文件系统中，每一个文件有唯一的绝对路径名，它是从根目录 (/) 开始，沿着目录层次结构向下、直到指定文件的所有目录名连接而成的字符串，各目录名之间以斜线字符 (/) 隔开，如 /usr/mengqc/mub1/m1.c。绝对路径名也称全路径名。

相对路径名是相对当前工作目录的路径名。如果当前工作目录是 /usr/mengqc，则文件 /usr/mengqc/mub1/m1.c 的相对路径名是：mub1/m1.c。相对路径名不能以斜线 (/) 字符开头。

为了访问在当前工作目录中和当前工作目录之上的文件，可以在相对路径名中使用特殊目录名——点 (.) 和点点 (..)。点 (.) 表示当前工作目录本身，而点点 (..) 代表当前工作目录的父目录。例如，当前工作目录为 /usr/mengqc/mub1，要列出 /usr/liu 目录的内容，可使用命令：

```
$ ls ../../liu
```

另外，还应注意：

① 在每个目录中都有点点目录文件 (..)。如 /usr/mengqc/mub1 的父目录是 /usr/mengqc，而后者的父目录是 /usr。

② 利用 ../ 表示父目录的形式可以连续使用，直至根目录，如上例所示。系统中的每个文件都可以利用相对路径名来命名。

(3) Linux 具有为一个文件起多个名字的功能，称为链接。被链接的文件可以存放在同一目录下，但是必须有不同的文件名；被链接的文件也可以存放在不同的目录下，有相同的文件名。

文件链接有两种形式，即硬链接和符号链接。

硬链接是在别的目录或本目录中增加目标文件的一个目录项，即登记在多个目录中。如图 1-2 所示的 m2.c 文件就在目录 mub1 和 liu 中都建立了目录项。一个文件的硬链接数可以在目录的长列表格式的第 2 列中看到，无额外链接的文件的链接数为 1。

应注意：不能对目录文件作硬链接；不能在不同的文件系统之间作硬链接。就是说，链接文件和被链接文件必须位于同一个文件系统中。

符号链接也称为软链接，是将一个路径名链接到一个文件，即建立一个小文本文件（如图 1-2 中的 abc），其中只包含所链接文件的全路径名。如图 1-2 中虚线箭头所指的那个文件是实际包含所有数据的文件。对符号链接文件进行读、写时，将沿着链接方向查访实际的文件。与硬链接不同的是，符号链接确实是一个新文件，当然它具有不同的 I 节点号；而硬链接并没有建立新文件。

符号链接没有硬链接的限制，可以对目录文件作符号链接，也可以在不同文件系统之间作符号链接。用 ln -s 命令建立符号链接时，源文件最好用绝对路径名，这样可以在任何工作目录下进行符号链接。而当源文件用相对路径名时，如果当前的工作路径与要创建的符号链接文件所在路径不同，则不能进行链接。

在目录长列表中，符号链接作为一种特殊的文件类型显示出来，其第一个字母是 l。符号链接的大小是其链接文件的路径名中的字节数。当用 `ls -l` 命令列出文件时，可以看到符号链接名后有一个箭头指向源文件或目录，例如：

```
lrwxrwxrwx ... 14 Jun 20 10:20 /etc/motd ->/original_file
```

其中表示“文件大小”的数字“14”恰好说明源文件名“/original_file”由 14 个字符构成。

2. 常用目录操作命令

(1) mkdir 命令

`mkdir` 命令用来创建目录。例如：

```
$ mkdir /usr/mengqc/test      在目录/usr/mengqc下建立子目录test。
```

(2) rmdir 命令

`rmdir` 命令可以从一个目录中删除一个或多个空的子目录。另外，当前工作目录必须在被删除目录之上，不能是被删除目录本身，也不能是被删除目录的子目录。

应注意：目录被删除之前应该是空目录。就是说，删除该目录之前，其中的所有文件必须用 `rm` 命令全部删除。

虽然还可以用带有 `-r` 选项的 `rm` 命令递归删除一个目录中的所有文件和该目录本身，但是这样做存在很大的危险性。

(3) cd 命令

`cd` 命令用来改变工作目录。

(4) pwd 命令

`pwd` 命令显示出当前工作目录的绝对路径。

(5) ls 命令

`ls` 命令列出指定目录的内容。其一般使用格式是：

```
ls [选项] [目录或文件]
```

如果给出的参数是目录，则该命令将列出其中所有子目录与文件的信息；如果给出的参数是文件，则列出有关该文件属性的一些信息。在默认情况下，输出条目按字母顺序排列。

常用选项有：

`-a` 显示指定目录下所有子目录和文件，包括以“.”开头的隐藏文件（如 `.cshrc`）。

`-d` 如果参数是目录，则只显示它的名字（不显示其内容）。往往与 `-l` 选项一起使用，以得到目录的详细信息。

`-F` 在列出的文件名后面加上不同的符号，以区分不同类型的文件。附加的符号有：`/` 表示目录；`*` 表示可执行文件；`@` 表示符号链接文件；`|` 表示管道；`=` 表示 socket 文件。

`-R` 递归显示指定目录的各个子目录中的文件。

`-l` 以长格式显示文件的详细信息。输出的信息分成多列，它们依次是：

文件类型与权限 链接数 文件主 文件组 文件大小 建立或最近修改的时间 文件名

```
例如：-rw-r--r-- 2 mengqc group 198 Jun 23 2007 m2.c
```

它说明：`m2.c` 是普通文件，所有用户都可以读，只有文件主可以写，但都不可以执行；文件的链接数为 2；文件主是 `mengqc`；用户组是 `group`；文件大小为 198 字节；于 2007 年 6 月 23 日建立。

1.5 文件存取权限

每个文件都规定了权限，即允许用户对其进行什么操作。而用户对文件进行操作的前提就是拥有相应的权利。利用文件存取权限可以保护文件。

1. 用户和权限

(1) 文件主

Linux 为每个文件都分配了一个文件所有者，称为文件主，并赋予文件主唯一的注册名。对文件的控制取决于文件主或超级用户 (root)。

文件或目录的创建者对创建的文件或目录拥有特别使用权。当然，文件的所有关系是可以改变的，可以将文件或目录的所有权转让给其他用户，但只有文件主或 root 才有权改变文件的所有关系。文件的所有权的标志是用户 ID (UID)。

(2) 用户组

当系统管理员为用户建立账户之后，会分配一个组 ID 和一个特定的用户组名。通常，这些组名包含了有相同需求的用户，如一个开发部门的所有成员。采用组方式也有助于增强系统使用的安全性。

在 Linux 系统中，每个文件要隶属于一个用户组。当创建一个文件或目录时，系统会赋予它一个用户组关系，用户组的所有成员都可以使用此文件或目录。

文件用户组关系的标志是 GID。文件的 GID 只能由文件主或超级用户 (root) 来修改。利用 `chgrp` 命令可以改变文件的 GID。

(3) 存取权限

Linux 系统中的每个文件和目录都有存取许可权限，用它来确定谁可以通过何种方式对文件和目录进行访问和操作。

Linux 系统中规定了 4 种不同类型的用户：

- ① 文件主 (owner)；
- ② 同组用户 (group)；
- ③ 可以访问系统的其他用户 (others)；
- ④ 超级用户 (root)，具有管理系统的特权。

存取权限规定 3 种访问文件或目录的方式：读 (r)，写 (w)，可执行或查找 (x)。

当用 `ls -l` 命令显示文件或目录的详细信息时，最左边的一列为文件的存取权限。其中各位的含义如图 1-3 所示。

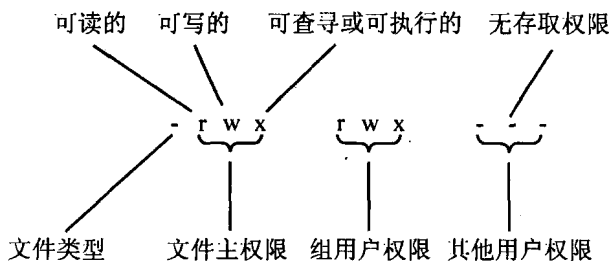


图 1-3 文件权限表示

对于普通文件：读权限（r）表示只允许相应用户读取该文件的内容，而禁止对它作任何的更改操作。凡是把文件作为输入参数的命令，都需要对该文件有读的权限，如 cat, more 等。写权限（w）表示允许相应用户打开并修改文件，如命令 vi, cp 等。执行权限（x）表示允许相应用户将该文件作为一个程序执行。

对于目录：读权限（r）表示可以列出在该目录中存放的文件，即读取目录内容列表。这一权限允许 shell 使用文件扩展名字符列出相匹配的文件名。写权限（w）表示允许从目录中删除或添加新的文件，通常只有目录主才有写权限。执行权限（x）表示允许在目录中进行查找，并能用 cd 命令将工作目录改为该目录。

2. 有关存取权限的命令

(1) chmod 命令

chmod 命令用于改变或设置文件或目录的存取权限。只有文件主或超级用户 root 才有权用 chmod 命令改变文件或目录的存取权限。

根据表示权限的方式，该命令有两种用法：符号法和数字法。

①符号法

一般使用格式是：chmod mode 文件名

其中，mode 的组成形式是：[who] [op] [perms]

这 3 部分必须按顺序输入。可以用多个 mode，但必须以逗号间隔。

who 表示操作对象，可以是下述字母中的任一个或者它们的组合：

u 表示“用户（user）”，即文件或目录的所有者。

g 表示“同组（group）用户”，即与文件属主有相同组 ID 的所有用户。

o 表示“其他（others）用户”。

a 表示“所有（all）用户”。它是系统默认值。

op 可以是：+ 表示添加某个权限；- 表示取消某个权限；= 表示赋予给定权限并取消其他所有权限（如果有的话）。

perms 表示权限，可用下述字母的任意组合：r 表示可读；w 表示可写；x 表示可执行。

例如：

```
$ chmod a+x ex1    将文件 ex1 的权限改为所有用户都有执行权限。
```

```
$ chmod u=r, ug=x ex1    将文件 ex1 的权限重新设置为文件主可以读和执行，组用户可以执行，其他用户无权访问。
```

②数字法

一般使用格式是：chmod mode 文件名

用八进制数字设置或改变文件的存取权限，就是用数字 1 和 0 表示图 1-3 中的 9 个权限位，置为 1 表示有相应权限，置为 0 表示没有相应权限。例如，某个文件的存取权限是：文件主有读、写和执行的权限，同组用户有读和执行的权限，其他用户仅有读的权限。用符号法表示是 rwxr-xr--，用二进制数字表示是 111 101 100。为了记忆和表示方便，通常用 3 个八进制数表示这 9 位二进制数。这样，上述二进制数就等价于八进制数 754。就是说，mode 是以 3 位八进制数字出现的，第 1 位表示文件主权限，第 2 位表示组用户权限，第 3 位表示其他用户权限。

例如：

`$ chmod 664 ex1` 使文件 `ex1` 的文件主和同组用户具有读、写权限，而其他用户只可读。

(2) `umask` 命令

`umask` 命令用来设置限制新建文件权限的掩码。一般使用格式是：`umask mode`

创建新文件时，由文件创建掩码决定其最初的权限。用户每次注册进入系统时，`umask` 命令都被执行，并自动设置掩码 `mode` 来限制新文件的权限。用户可以通过再次执行 `umask` 命令来改变默认值，新的权限将把旧的覆盖掉，即指定哪些权限将在以后新建文件的默认权限中被删除。

2 shell 程序设计

2.1 shell 概述

在前面介绍 Linux 命令时，shell 都作为命令解释程序出现，这是 shell 最常见的使用方式。除此以外，它还是一种高级程序设计语言，它有变量、关键字，有各种控制语句，如 `if`，`while`，`for` 等语句，支持函数模块，有自己的语法结构。利用 shell 程序设计语言可以编写出功能很强但代码简单的程序。特别是，它把相关的 Linux 命令有机地组合在一起，可大大提高编程的效率。充分利用 Linux 系统的开放性能，就能够设计出适合自己要求的命令。

2.1.1 shell 的特点和类型

1. shell 的特点

shell 是一种高级程序设计语言，其功能强大，为用户开发程序提供了便捷的手段。它具有如下突出特点：

(1) 对已有命令进行简单组合，如建立 shell 脚本，就构成新命令。

(2) 可以直接使用 shell 的内置命令，而不需创建新的进程，减少系统调度开销。

(3) 允许灵活地使用数据流，提供通配符、输入/输出重定向、管道线等机制，方便了模式匹配、I/O 处理和数据传输。

(4) 结构化的程序模块，提供了顺序流程控制、条件控制、循环控制等。

(5) 提供了在后台 (&) 执行命令的能力。

(6) 提供了可配置的环境，允许用户创建和修改命令、命令提示符和其他的系统行为。

2. shell 的类型

Linux 系统提供多种不同的 shell。常用的有 Bourne shell (简称 `sh`)，C shell (简称 `csh`)，Korn shell (简称 `ksh`) 和 Bourne Again shell (简称 `bash`)。

(1) Bourne shell

Bourne shell 是 AT&T Bell 实验室的 Steven Bourne 为 AT&T 的 UNIX 开发的，它是 UNIX 的默认 shell，也是其他 shell 的开发基础。Bourne shell 在编程方面相当优秀，但在处理与用户的交互方面不如其他几种 shell。

(2) C shell

C shell 是加州伯克利大学的 Bill Joy 为 BSD UNIX 开发的，与 `sh` 不同，它的语法与 C

语言很相似。它提供了 Bourne shell 所不能处理的用户交互特征，如命令补全、命令别名、历史命令替换等。但是，C shell 与 Bourne shell 并不兼容。

(3) Korn shell

Korn shell 是 AT&T Bell 实验室的 David Korn 开发的，它集合了 C shell 和 Bourne shell 的优点，并且与 Bourne shell 向下完全兼容。Korn shell 的效率很高，其命令交互界面和编程交互界面都很好。

(4) Bourne Again shell

Bourne Again shell (即 bash) 是自由软件基金会 (GNU) 开发的一个 shell，它是 Linux 系统中一个默认的 shell。bash 不但与 Bourne shell 兼容，还继承了 C shell，Korn shell 的优点，如命令行历史、命令行编辑、命令补全、别名扩展等功能。

2.1.2 shell 脚本的建立和执行

1. 建立 shell 脚本

建立 shell 脚本的步骤与建立普通文本文件的方式相同，可以利用编辑器 (如 vi) 进行程序录入和编辑加工。

2. 执行 shell 脚本的方式

(1) 输入定向到 shell 脚本。其一般形式是：`$ bash < 脚本名`

例如：`$ bash < ex1`

shell 从文件 ex1 中读取命令行，并执行它们。当 shell 到达文件末尾时，就终止执行。

(2) 以脚本名作为 bash 参数。其一般形式是：`$ bash 脚本名 [参数]`

例如：`$ bash ex2 /usr/meng /usr/zhang`

其执行过程与上一种方式相同。其好处是，能在脚本名后面带有参数，从而将参数值传递给程序中的命令。如果用目前 shell (以 . 表示) 执行脚本文件，则可以使用如下简便形式：

`$. 脚本名 [参数]`

(3) 将 shell 脚本的权限设置为可执行，然后在提示符下直接输入脚本名，予以执行。

通常，直接编辑生成的脚本文件没有“执行”权限。如果想把它直接当做命令执行，就需要利用命令 `chmod` 将它设置为有“执行”权限。例如：

`$ chmod a+x ex2`

该命令执行后，脚本 ex2 就对所有用户都有“执行”权限。然后，将该脚本所在的目录添加到命令搜索路径 (PATH) 中。例如：

`$ PATH= $PATH:.`

就把当前工作目录 (以 “.” 表示) 添加到命令搜索路径中。这样，在提示符后输入脚本名 ex2 就可直接执行该文件：`$ ex2`

注意，在下面程序示例中都采用这种方式，即：预先将 shell 脚本的权限置为可执行。因此，shell 脚本名可直接作为命令名使用。

2.2 shell 变量和算术运算

2.2.1 shell 变量

shell 程序中利用变量存放字符串。shell 变量很简单，可以在使用时“边定义、边赋