《嵌入式 LINUX 应用程序开发标准教程》

作者：华清远见

第 2 章　Linux 基础命令

本章目标

Linux 是一个高可靠、高性能的系统，而所有这些优越性只有在直接使用 Linux 命令行时（shell 环境）才能充分地体现出来。在本章将帮助读者学会如下内容。

- 掌握 shell 基本概念
- 熟练使用 Linux 中用户管理命令
- 熟练使用 Linux 中系统相关命令
- 熟练使用 Linux 中文件目录相关命令
- 熟练使用 Linux 中打包压缩相关命令
- 熟练使用 Linux 中文件比较合并相关命令
- 熟练使用 Linux 中网络相关命令
- 了解 Linux 的启动过程
- 深入了解 init 进程及其配置文件
- 能够独立完成在 Linux 中解压缩软件
- 学会添加环境变量
- 能够独立定制 Linux 中的系统服务

2.1 Linux 常用命令

在安装完 Linux 再次启动之后，就可以进入到与 Windows 类似的图形化界面了。这个界面就是 Linux 图形化界面 X 窗口系统（简称 X）的一部分。要注意的是，X 窗口系统仅仅是 Linux 上面的一个软件（或者也可称为服务），它不是 Linux 自身的一部分。虽然现在的 X 窗口系统已经与 Linux 整合得相当好了，但毕竟还不能保证绝对的可靠性。另外，X 窗口系统是一个相当耗费系统资源的软件，它会大大地降低 Linux 的系统性能。因此，若是希望更好地享受 Linux 所带来的高效及高稳定性，建议读者尽可能地使用 Linux 的命令行界面，也就是 shell 环境。

当用户在命令行下工作时，不是直接同操作系统内核交互信息的，而是由命令解释器接受命令，分析后再传给相关的程序。shell 是一种 Linux 中的命令行解释程序，就如同 command.com 是 DOS 下的命令解释程序一样，为用户提供使用操作系统的接口。它们之间的关系如图 2.1 所示。用户在提示符下输入的命令都由 shell 先解释然后传给 Linux 内核。

小知识

- shell 是命令语言、命令解释程序及程序设计语言的统称。它不仅拥有自己内建的 shell 命令集，同时也能被系统中其他应用程序所调用。
- shell 的一个重要特性是它自身就是一个解释型的程序设计语言，shell 程序设计语言支持绝大多数在高级语言中能见到的程序元素，如函数、变量、数组和程序控制结构。shell 编程语言简单易学，任何在提示符中能键入的命令都能放到一个可执行的 shell 程序中。关于 shell 编程的详细讲解，感兴趣的读者可以参见其他相关书籍。

Linux 中运行 shell 的环境是“系统工具”下的“终端”，读者可以单击“终端”以启动 shell 环境。这时屏幕上显示类似“[david@localhost home]$”的信息，其中，david 是指系统用户，localhost 是计算机名，而 home 是指当前所在的目录。

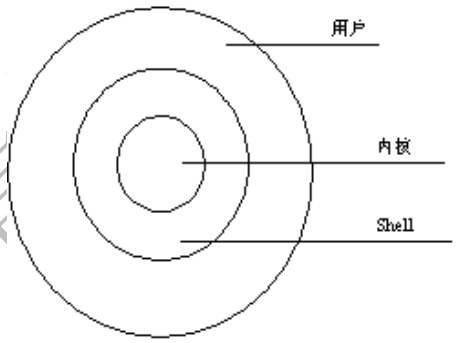

图 2.1 内核、shell 和用户的关系

由于 Linux 中的命令非常多，要全部介绍几乎是不可能的。因此，在本书按照命令的用途进行分类讲解，并且对每一类中最常用的命令详细讲解，同时列出同一类中的其他命令。由于同一类的命令都有很大的相似性，因此，读者通过学习本书中所列命令，可以很快地掌握其他命令。

命令格式说明。

- 格式中带[]的表明为可选项，其他为必选项。
- 选项可以多个连带写入。
- 本章后面选项参数列表中加粗的含义是：该选项是非常常用的选项。

2.1.1 用户系统相关命令

Linux 是一个多用户的操作系统，每个用户又可以属于不同的用户组，下面，首先来熟悉一下 Linux 中的用户切换和用户管理的相关命令。

1．用户切换（su）

（1）作用。

变更为其他使用者的身份，主要用于将普通用户身份转变为超级用户，而且需输入相应用户密码。

（2）格式。

su [选项] [使用者]

其中的使用者为要变更的对应使用者。

（3）常见参数。

主要选项参数如表 2.1 所示。

表 2.1　　su 命令常见参数列表

选　　项	参 数 含 义
-，-l，--login	为该使用者重新登录，大部分环境变量（如 HOME、SHELL 和 USER 等）和工作目录都是以该使用者（USER）为主。若没有指定 USER，缺省情况是 root
-m，-p	执行 su 时不改变环境变量
-c，--command	变更账号为 USER 的使用者，执行指令（command）后再变回原来使用者

（4）使用示例。

```
[david@localhost ~]$ su - root
Password:
[root@localhost ~]#
```

示例通过 su 命令将普通用户变更为 root 用户，并使用选项“-”携带 root 环境变量。

（5）使用说明。

- 在将普通用户变更为 root 用户时建议使用“-”选项，这样可以将 root 的环境变量和工作目录同时带入，否则在以后的使用中可能会由于环境变量的原因而出错。
- 在转变为 root 权限后，提示符变为#。

小知识

环境变量实际上就是用户运行环境的参数集合。Linux 是一个多用户的操作系统。而且在每个用户登录系统后，都会有一个专有的运行环境。通常每个用户默认的环境都是相同的，而这个默认环境实际上就是一组环境变量的定义。用户可以对自己的运行环境进行定制，其方法就是修改相应的系统环境变量。

常见的环境变量如下。

☆PATH 是系统路径。

☆HOME 是系统根目录。

☆HISTSIZE 是指保存历史命令记录的条数。

☆LOGNAME 是指当前用户的登录名。

☆HOSTNAME 是指主机的名称，若应用程序要用到主机名，通常是从这个环境变量中来取得的。

☆SHELL 是指当前用户用的是哪种 shell。

☆LANG/LANGUGE 是和语言相关的环境变量，使用多种语言的用户可以修改此环境变量。

☆MAIL 是指当前用户的邮件存放目录。

设置环境变量方法如下。

- ✓ 通过 echo 显示字符串（指定环境变量）。
- ✓ 通过 export 设置新的环境变量。
- ✓ 通过 env 显示所有环境变量。
- ✓ 通过 set 命令显示所有本地定义的 shell 变量。
- ✓ 通过 unset 命令来清除环境变量。

读者可以试着用“env”命令查看“su - root”（或“su –”）和“su root”的区别。

2．用户管理（useradd 和 passwd）

Linux 中常见用户管理命令如表 2.2 所示，本书仅以 useradd 和 passwd 为例进行详细讲解，其他命令类似，请读者自行学习使用。

表 2.2 Linux 常见用户管理命令

命　令	命 令 含 义	格　式
useradd	添加用户账号	useradd [选项] 用户名
usermod	设置用户账号属性	usermod [选项] 属性值
userdel	删除对应用户账号	userdel [选项] 用户名
groupadd	添加组账号	groupadd [选项] 组账号
groupmod	设置组账号属性	groupmod [选项] 属性值
groupdel	删除对应组账号	groupdel [选项] 组账号
passwd	设置账号密码	passwd [对应账号]
id	显示用户 ID、组 ID 和用户所属的组列表	id [用户名]
groups	显示用户所属的组	groups [组账号]
who	显示登录到系统的所有用户	who

（1）作用。

① useradd：添加用户账号。

② passwd：更改对应用户的账号密码。

（2）格式。

① useradd：useradd [选项] 用户名。

② passwd：passwd [选项] [用户名]。

其中的用户名为修改账号密码的用户，若不带用户名，缺省为更改当前使用者的密码。

（3）常用参数

① useradd 主要选项参数如表 2.3 所示。

表 2.3 useradd 命令常见参数列表

选　项	参 数 含 义
-g	指定用户所属的群组
-m	自动建立用户的登入目录
-n	取消建立以用户名称为名的群组

② passwd：一般很少使用选项参数。

（4）使用实例。

```
[root@localhost ~]# useradd david
[root@localhost ~]# passwd david
New password:  （输入密码）
Retype new password:  （再输入一次密码，以确认输入的正确性）
passwd: all authentication tokens updated successfully
[root@localhost ~]# su - david
[david@localhost ~]$
[david@localhost ~]$ pwd（查看当前目录）
/home/david （该用户的工作目录）
```

实例中先添加了用户名为 david 的用户，接着又为该用户设置了账号密码。从 su 的命令可以看出，该用户添加成功，其工作目录为“/home/david”。

（5）使用说明。

- 在添加用户时，这两个命令是一起使用的，其中，useradd 必须用 root 的权限。而且 useradd 指令所建立的账号，实际上是保存在“/etc/passwd”文本文件中，文件中每一行包含一个账号信息。
- 在缺省情况下，useradd 所做的初始化操作包括在“/home”目录下为对应账号建立一个同名的主目录，并且还为该用户单独建立一个与用户名同名的组。
- adduser 只是 useradd 的符号链接（关于符号链接的概念在本节后面会有介绍），两者是相同的。
- passwd 还可用于普通用户修改账号密码，Linux 并不采用类似 Windows 的密码回显（显示为*号），所以输入的这些字符用户是看不见的。密码最好包括字母、数字和特殊符号，并且设成 6 位以上。

3．系统管理命令（ps 和 kill）

Linux 中常见的系统管理命令如表 2.4 所示，本书以 ps 和 kill 为例进行讲解。

表 2.4　　Linux 常见系统管理命令

命　令	命 令 含 义	格　式
ps	显示当前系统中由该用户运行的进程列表	ps [选项]
top	动态显示系统中运行的程序（一般为每隔 5s）	top
kill	输出特定的信号给指定 PID（进程号）的进程	kill [选项] 进程号（PID）
uname	显示系统的信息（可加选项-a）	uname [选项]
setup	系统图形化界面配置	setup
crontab	循环执行例行性命令	crontab [选项]
shutdown	关闭或重启 Linux 系统	shutdown [选项] [时间]
uptime	显示系统已经运行了多长时间	uptime
clear	清除屏幕上的信息	clear

（1）作用。

① ps：显示当前系统中由该用户运行的进程列表。

② kill：输出特定的信号给指定 PID（进程号）的进程，并根据该信号完成指定的行为。其中可能的信号有进程挂起、进程等待、进程终止等。

（2）格式。

① ps：ps [选项]。

② kill：kill [选项] 进程号（PID）。

kill 命令中的进程号为信号输出的指定进程的进程号，当选项是缺省时为输出终止信号给该进程。

（3）常见参数。

① ps 主要选项参数如表 2.5 所示。

表 2.5　　ps 命令常见参数列表

选　项	参 数 含 义
-ef	查看所有进程及其 PID（进程号）、系统时间、命令详细目录、执行者等
-aux	除可显示-ef 所有内容外，还可显示 CPU 及内存占用率、进程状态
-w	显示加宽并且可以显示较多的信息

② kill 主要选项参数如表 2.6 所示。

表 2.6　　kill 命令常见参数列表

选　项	参 数 含 义
-s	将指定信号发送给进程
-p	打印出进程号（PID），但并不送出信号

-l	列出所有可用的信号名称

（4）使用实例。

```
[root@localhost root]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0  2005 ?        00:00:05 init
root         2     1  0  2005 ?        00:00:00 [keventd]
root         3     0  0  2005 ?        00:00:00 [ksoftirqd_CPU0]
root         4     0  0  2005 ?        00:00:00 [ksoftirqd_CPU1]
root      7421     1  0  2005 ?        00:00:00 /usr/local/bin/ntpd -c /etc/ntp.
root     21787 21739  0 17:16 pts/1     00:00:00 grep ntp
[root@localhost root]# kill -9 7421 （杀死进程）
[root@localhost root]# ps -ef|grep ntp
root     21789 21739  0 17:16 pts/1    00:00:00 grep ntp
```

该实例中首先查看所有进程，并终止进程号为 7421 的 ntp 进程，之后再次查看时已经没有该进程号的进程。

（5）使用说明。

- ps 在使用中通常可以与其他一些命令结合起来使用，主要作用是提高效率。
  - ps 选项中的参数 w 可以写多次，通常最多写 3 次，它的含义为加宽 3 次，这足以显示很长的命令行了。例如：ps –auxwww。

小知识

管道是 Linux 中信息通信的重要方式。它是把一个程序的输出直接连接到另一个程序的输入，而不经过任何中间文件。管道线是指连接两个或更多程序管道的通路。在 shell 中字符”|”表示管道线。如前例子中的 ps –ef|grep ntp 所示，ps –ef 的结果直接输入到 grep ntp 的程序中（关于 grep 命令在后面会有详细的介绍）。grep、pr、sort 和 wc 都可以在上述管道线上工作。读者可以灵活地运用管道机制

4．磁盘相关命令（fdisk）

Linux 中与磁盘相关的命令如表 2.7 所示，本书仅以 fdisk 为例进行讲解。

表 2.7　　Linux 常见系统管理命令

选　项	参 数 含 义	格　　式
free	查看当前系统内存的使用情况	free [选项]
df	查看文件系统的磁盘空间占用情况	df [选项]
du	统计目录（或文件）所占磁盘空间的大小	du [选项]
fdisk	查看硬盘分区情况及对硬盘进行分区管理	fdisk [-l]

（1）作用。

fdisk 可以查看硬盘分区情况，并可对硬盘进行分区管理，这里主要介绍如何查看硬盘分区情况，另外，fdisk 也是一个非常好的硬盘分区工具，感兴趣的读者可以另外查找资料学习如何使用 fdisk 进行硬盘分区。

（2）格式。

fdisk [-l]

（3）使用实例。

```
[root@localhost ~]# fdisk -l
Disk /dev/hda: 40.0 GB, 40007761920 bytes
240 heads, 63 sectors/track, 5168 cylinders
Units = cylinders of 15120 * 512 = 7741440 bytes
   Device Boot      Start         End      Blocks   Id  System
```

```
/dev/hda1   *          1        1084     8195008+   c  W95 FAT32 (LBA)
/dev/hda2            1085        5167    30867480    f  W95 Ext'd (LBA)
/dev/hda5            1085        2439    10243768+   b  W95 FAT32
/dev/hda6            2440        4064    12284968+   b  W95 FAT32
/dev/hda7            4065        5096     7799526   83  Linux
/dev/hda8            5096        5165      522081   82  Linux swap
Disk /dev/sda: 999 MB, 999816704 bytes
4 heads, 8 sectors/track, 61023 cylinders
Units = cylinders of 32 * 512 = 16384 bytes
Disk identifier: 0x00000000

Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *          1       61024      976379+   b  W95 FAT32
```

可以看出，使用“fdisk –l”列出了文件系统的分区情况。

（4）使用说明

- 使用 fdisk 必须拥有 root 权限。
- IDE 硬盘对应的设备名称分别为 hda、hdb、hdc 和 hdd，SCSI 硬盘对应的设备名称则为 sda、sdb、…。此外，hda1 代表 hda 的第一个硬盘分区，hda2 代表 hda 的第二个分区，依此类推。
- 通过查看/var/log/messages 文件，可以找到 Linux 系统已辨认出来的设备代号。

5．文件系统挂载命令（mount）

（1）作用。

挂载文件系统，它的使用权限是超级用户或/etc/fstab 中允许的使用者。正如 1.2.1 节中所述，挂载是指在分区和目录之间建立映射关系的过程，而挂载点是指挂载在文件树中的位置。使用 mount 命令可以把文件系统挂载到相应的目录下，并且由于 Linux 中把设备都当成文件一样使用，因此，mount 命令也可以挂载不同的设备。

通常，在 Linux 下“/mnt”目录是专门用于挂载不同的文件系统的，它可以在该目录下新建不同的子目录来挂载不同的设备文件系统。

（2）格式。

mount [选项] [类型] 设备文件名 挂载点目录

其中的类型是指设备文件的类型。

（3）常见参数

mount 常见参数如表 2.8 所示。

表 2.8　　mount 命令选项常见参数列表

选　项	参 数 含 义
-a	依照/etc/fstab 的内容装载所有相关的硬盘
-l	列出当前已挂载的设备、文件系统名称和挂载点
-t 类型	将后面的设备以指定类型的文件格式装载到挂载点上。常见的类型有前面介绍过的几种：vfat、ext3、ext2、iso9660、nfs 等
-f	通常用于除错。它会使 mount 不执行实际挂上的动作，而是模拟整个挂上的过程，通常会和-v 一起使用

（4）使用实例。

使用 mount 命令主要通过以下几个步骤。

① 确认是否为 Linux 可以识别的文件系统，Linux 可识别的文件系统只要是以下几种。

- Windows 95/98 常用的 FAT32 文件系统：vfat。
- WindowsNT/2000 的文件系统：ntfs。
- OS/2 用的文件系统：hpfs。
- Linux 用的文件系统：ext2、ext3、nfs。
- CD-ROM 光盘用的文件系统：iso9660。

② 确定设备的名称，可通过使用命令“fdisk -l”查看。

③ 查找挂载点。

必须确定挂载点已经存在，也就是在“/mnt”下的相应子目录已经存在，一般建议在“/mnt”下新建几个如“/mnt/windows”，“/mnt/usb”的子目录，现在有些新版本的 Linux（如 Fedora、Ubuntu、红旗 Linux、中软 Linux、MandrakeLinux）都可自动挂载文件系统，Red Hat Linux 仅可自动挂载光驱。

④ 挂载文件系统如下所示。

```
[root@locaohost ~]# mkdir -p /mnt/win/c
[root@locaohost ~]# mount -t vfat /dev/hda1 /mnt/win/c
[root@localhost ~]# cd /mnt/win/c
24.s03e01.pdtv.xvid-sfm.rmvb  Documents and Settings  Program Files
24.s03e02.pdtv.xvid-sfm.rmvb  Downloads                Recycled
...
```

C 盘是原先笔者 Windows 系统的启动盘。可见，在挂载了 C 盘之后，可直接访问 Windows 下的 C 盘的内容。

⑤ 在使用完该设备文件后可使用命令 umount 将其卸载。

```
[root@localhost ~]# umount /mnt/win/c
[root@localhost ~]# cd /mnt/win/c
[root@localhost ~]# ls /mnt/win/c
```

可见，此时目录“/mnt/win/c”下为空。Windows 下的 C 盘已被成功卸载。

小知识

- 在 Linux 下如何使用 U 盘呢？

一般 U 盘为 SCSI 格式的硬盘，其格式为 vfat 格式，其设备号可通过“fdisk –l”进行查看，假若设备名为“/dev/sda1”，则可用如下命令将其挂载：

mount -t vfat /dev/sda1 /mnt/usb

- 若想设置在开机时自动挂载，可在文件“/etc/fstab”中加入相应的设置行即可。

2.1.2 文件相关命令

Linux 中有关文件的操作非常重要，也非常常用，本节将对 Linux 系统的文件操作命令进行详细讲解。

1．cd

（1）作用。

改变当前工作目录。

（2）格式。

cd [路径]

其中的路径为要改变的工作目录，可为相对路径或绝对路径。

（3）使用实例。

```
[root@localhost ~]# cd /home/david/
[root@localhost david]# pwd
```

```
[root@localhost david]# /home/david/
```

该实例中变更工作目录为“/home/david/”，在后面的“pwd”（显示当前目录）的结果中可以看出。

（4）使用说明。

- 该命令将当前目录改变至指定路径的目录。若没有指定路径，则回到用户的主目录（例如：“/home/david”为用户 david 的主目录）。为了改变到指定目录，用户必须拥有对指定目录的执行和读权限。
- 该命令可以使用通配符。
- 使用“cd –”可以回到前次工作目录。
- “./”代表当前目录，“../”代表上级目录。

2．ls

（1）作用

列出目录和文件的信息。

（2）格式。

ls [选项] [文件]

其中文件选项为指定查看指定文件的相关内容，若未指定文件，默认查看当前目录下的所有文件。

（3）常见参数。

ls 主要选项参数见表 2.9 所示。

表 2.9　　ls 命令常见参数列表

选　　项	参 数 含 义
-1，--format=single-column	一行输出一个文件（单列输出）
-a，-all	列出目录中所有文件，包括以“.”开头的隐藏文件
-d	将目录名和其他文件一样列出，而不是列出目录的内容
-l,--format=long, --format=verbose	除每个文件名外，增加显示文件类型、权限、硬链接数、所有者名、组名、大小（Byte）及时间信息（如未指明是其他时间即指修改时间）
-f	不排序目录内容，按它们在磁盘上存储的顺序列出

（4）使用实例。

```
[david@localhost test]$ ls -l
total 220
drwxr-xr-x    2 root    root        4096 Mar 31  2005 bin
drwxr-xr-x    3 root    root        4096 Apr  3  2005 boot
-rw-r--r--    1 root    root           0 Apr 24  2002 test.run
…
```

该实例查看当前目录下的所有文件，并通过选项“-l”显示出详细信息。

显示格式说明如下。

文件类型与权限 链接数 文件属主 文件属组 文件大小 修改的时间 名字

（5）使用说明。

- 在 ls 的常见参数中，-l（长文件名显示格式）的选项是最为常见的。可以详细显示出各种信息。
- 若想显示出所有“.”开头的隐藏文件，可以使用-a，这在嵌入式开发中很常用。

注意	Linux 中的可执行文件不是与 Windows 一样通过文件扩展名来标识的，而是通过设置文件相应的可执行属性来实现的。

3．mkdir

（1）作用。
创建一个目录。
（2）格式。
mkdir [选项] 路径
（3）常见参数。
mkdir 主要选项参数如表 2.10 所示。

表 2.10 mkdir 命令常见参数列表

选　项	参 数 含 义
-m	对新建目录设置存取权限，也可以用 chmod 命令（在本节后会有详细说明）设置
-p	可以是一个路径名称。此时若此路径中的某些目录尚不存在，在加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录

（4）使用实例。

```
[david@localhost ~]$ mkdir -p ./hello/my
[david@localhost ~]$ cd hello/my
[david@localhost my]$ pwd（查看当前目录命令）
/home/david/hello/my
```

该实例使用选项“-p”一次创建了./hello/my 多级目录。

```
[david@localhost my]$ mkdir -m 777 ./why
[david@localhost my]$ ls -l
total 4
drwxrwxrwx    2 root    root       4096 Jan 14 09:24 why
```

该实例使用改选项“-m”创建了相应权限的目录。对于“777”的权限在本节后面会有详细的说明。
（5）使用说明。
该命令要求创建目录的用户在创建路径的上级目录中具有写权限，并且路径名不能是当前目录中已有的目录或文件名称。

4．cat

（1）作用。
连接并显示指定的一个或多个文件的有关信息。
（2）格式。
cat[选项]文件 1 文件 2…
其中的文件 1、文件 2 为要显示的多个文件。
（3）常见参数。
cat 命令的常见参数如表 2.11 所示。

表 2.11 cat 命令常见参数列表

选　项	参 数 含 义
-n	由第一行开始对所有输出的行数编号
-b	和-n 相似，只不过对于空白行不编号

（4）使用实例。

```
[david@localhost ~]$ cat -n hello1.c hello2.c
     1  #include <stdio.h>
     2  void main()
     3  {
     4         printf("Hello!This is my home!\n");
     5  }
     6  #include <stdio.h>
     7  void main()
     8  {
     9         printf("Hello!This is your home!\n");
    10  }
```

在该实例中，指定对 hello1.c 和 hello2.c 进行输出，并指定行号。

5．cp、mv 和 rm

（1）作用。

① cp：将给出的文件或目录复制到另一文件或目录中。

② mv：为文件或目录改名或将文件由一个目录移入另一个目录中。

③ rm：删除一个目录中的一个或多个文件或目录。

（2）格式。

① cp：cp [选项] 源文件或目录 目标文件或目录

② mv：mv [选项] 源文件或目录 目标文件或目录

③ rm：rm [选项] 文件或目录

（3）常见参数。

① cp 主要选项参数如表 2.12 所示。

表 2.12　　cp 命令常见参数列表

选　项	参 数 含 义
-a	保留链接、文件属性，并复制其子目录，其作用等于 dpr 选项的组合
-d	复制时保留链接
-f	删除已经存在的目标文件而不提示
-i	在覆盖目标文件之前将给出提示要求用户确认。回答 y 时目标文件将被覆盖，而且是交互式复制
-p	此时 cp 除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中
-r	若给出的源文件是一个目录文件，此时 cp 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名

② mv 主要选项参数如表 2.13 所示。

表 2.13　　mv 命令常见参数列表

选　项	参 数 含 义
-i	若 mv 操作将导致对已存在的目标文件的覆盖，此时系统询问是否重写，并要求用户回答 y 或 n，这样可以避免误覆盖文件
-f	禁止交互操作。在 mv 操作要覆盖某已有的目标文件时不给任何指示，在指定此选项后，i 选项将不再起作用

③ rm 主要选项参数如表 2.14 所示。

表 2.14　　rm 命令常见参数列表

选　项	参 数 含 义
-i	进行交互式删除
-f	忽略不存在的文件，但从不给出提示
-r	指示 rm 将参数中列出的全部目录和子目录均递归地删除

（4）使用实例。

① cp

```
[root@www hello]# cp -a ./my/why/ ./
[root@www hello]# ls
my  why
```

该实例使用-a 选项将“/my/why”目录下的所有文件复制到当前目录下。而此时在原先目录下还有原有的文件。

② mv

```
[root@www hello]# mv -i ./my/why/ ./
[root@www hello]# ls
my  why
```

该实例中把“/my/why”目录下的所有文件移至当前目录，则原目录下文件被自动删除。

③ rm

```
[root@www hello]# rm –r -i ./why
rm: descend into directory './why'? y
rm: remove './why/my.c'? y
rm: remove directory './why'? y
```

该实例使用“-r”选项删除“./why”目录下所有内容，系统会进行确认是否删除。

（5）使用说明。

① cp：该命令把指定的源文件复制到目标文件，或把多个源文件复制到目标目录中。

② mv

- 该命令根据命令中第二个参数类型的不同（是目标文件还是目标目录）来判断是重命名还是移动文件，当第二个参数类型是文件时，mv 命令完成文件重命名，此时，它将所给的源文件或目录重命名为给定的目标文件名；
- 当第二个参数是已存在的目录名称时，mv 命令将各参数指定的源文件均移至目标目录中；
- 在跨文件系统移动文件时，mv 先复制，再将原有文件删除，而连至该文件的链接也将丢失。

③ rm

- 如果没有使用- r 选项，则 rm 不会删除目录；
- 使用该命令时一旦文件被删除，它是不能被恢复的，所以最好使用-i 参数。

6．chown 和 chgrp

（1）作用。

① chown：修改文件所有者和组别。

② chgrp：改变文件的组所有权。

（2）格式。

① chown：chown [选项]...文件所有者[所有者组名] 文件

其中的文件所有者为修改后的文件所有者。

② chgrp：chgrp [选项]... 文件所有组 文件

其中的文件所有组为改变后的文件组拥有者。

（3）常见参数。

chown 和 chgrp 的常见参数意义相同，其主要选项参数如表 2.15 所示。

表 2.15　　chown 和 chgrp 命令常见参数列表

选　项	参 数 含 义
-c，-changes	详尽地描述每个 file 实际改变了哪些所有权
-f，--silent，--quiet	不打印文件所有权就不能修改的报错信息

（4）使用实例。

在笔者的系统中一个文件的所有者原先是这样的。

```
[root@localhost test]#$ ls -l
-rwxr-xr-x   15 apectel  david        4096  6月  4  200X uClinux-dist.tar
```

可以看出，这是一个文件，文件拥有者是 apectel，具有可读写和执行的权限，它所属的用户组是 david，具有可读和执行的权限，但没有可写的权限，同样，系统其他用户对其也只有可读和执行的权限。

首先使用 chown 将文件所有者改为 root。

```
[root@localhost test]# chown root uClinux-dist.tar
[root@localhost test]# ls –l
-rwxr-xr-x   15 root     david        4096  6月  4  200X uClinux-dist.tar
```

可以看出，此时，该文件拥有者变为了 root，它所属文件用户组不变。

接着使用 chgrp 将文件用户组变为 root。

```
[root@localhost test]# chgrp root uClinux-dist.tar
[root@localhost test]# ls –l
-rwxr-xr-x   15 root     root         4096  6月  4  200X uClinux-dist.tar
```

（5）使用说明。

■　使用 chown 和 chgrp 必须拥有 root 权限。

小知识	在进行有关文件的操作时，若想避免输入冗长的文件，在文件名没有重复的情况下可以使用输入文件前几个字母＋<Tab>键的方式，即：cd /uC<tab>会显示 cd /uClinux-list

7．chmod

（1）作用。

改变文件的访问权限。

（2）格式。

chmod 可使用符号标记进行更改和八进制数指定更改两种方式，因此它的格式也有两种不同的形式。

① 符号标记：chmod [选项]…符号权限[符号权限]…文件

其中的符号权限可以指定为多个，也就是说，可以指定多个用户级别的权限，但它们中间要用逗号分开表示，若没有显式指出则表示不作更改。

② 八进制数：chmod [选项] …八进制权限 文件…

其中的八进制权限是指要更改后的文件权限。

（3）选项参数。

chmod 主要选项参数如表 2.16 所示。

表 2.16　　chmod 命令常见参数列表

选　项	参 数 含 义
-c	若该文件权限确实已经更改，才显示其更改动作
-f	若该文件权限无法被更改也不要显示错误信息
-v	显示权限变更的详细资料

（4）使用实例。

chmod 涉及文件的访问权限，在此对相关的概念进行简单的回顾。

在 1.3.1 节中已经提到，文件的访问权限可表示成：**- rwx rwx rwx**。在此设有 3 种不同的访问权限：读（r）、写（w）和运行（x）。3 个不同的用户级别：文件拥有者（u）、所属的用户组（g）和系统里的其他用户（o）。在此，可增加一个用户级别 a（all）来表示所有这 3 个不同的用户级别。

① 第一种符号连接方式的 chmod 命令中，用加号“+”代表增加权限，用减号“–”代表删除权限，等于号“=”代表设置权限。

例如，原先笔者系统中有文件 uClinux20031103.tgz，其权限如下所示。

```
[root@localhost test]# ls -l
-rw-r--r--    1 root     root     79708616 Mar 24  2005 uClinux20031103.tgz
[root@localhost test]# chmod a+rx,u+w uClinux20031103.tgz
[root@localhost test]# ls -l
-rwxr-xr-x    1 root     root     79708616 Mar 24  2005 uClinux20031103.tgz
```

可见，在执行了 chmod 之后，文件拥有者除拥有所有用户都有的可读和执行的权限外，还有可写的权限。

② 对于第二种八进制数指定的方式，将文件权限字符代表的有效位设为“1”，即“rw-”、“rw-”和“r--”的八进制表示为“110”、“110”、“100”，把这个二进制串转换成对应的八进制数就是 6、6、4，也就是说该文件的权限为 664（三位八进制数）。这样对于转化后八进制数、二进制及对应权限的关系如表 2.17 所示。

表 2.17　　转化后八进制数、二进制及对应权限的关系

转换后八进制数	二　进　制	对 应 权 限	转换后八进制数	二　进　制	对 应 权 限
0	000	没有任何权限	1	001	只能执行
2	010	只写	3	011	只写和执行
4	100	只读	5	101	只读和执行
6	110	读和写	7	111	读、写和执行

同上例，原先笔者系统中有文件 genromfs-0.5.1.tar.gz，其权限如下所示。

```
[root@localhost test]# ls -l
-rw-rw-r--    1 david    david       20543 Dec 29  2004 genromfs-0.5.1.tar.gz
[root@localhost test]# chmod 765 genromfs-0.5.1.tar.gz
[root@localhost test]# ls -l
-rwxrw-r-x    1 david    david       20543 Dec 29  2004 genromfs-0.5.1.tar.gz
```

可见，在执行了 chmod 765 之后，该文件的拥有者权限、文件组权限和其他用户权限都恰当地对应了。

（5）使用说明

- 使用 chmod 必须具有 root 权限。

	想一想	chmod o+x uClinux20031103.tgz 是什么意思？它所对应的八进制数指定更改应如何表示？

8. grep

（1）作用。

在指定文件中搜索特定的内容，并将含有这些内容的行标准输出。

（2）格式。

grep [选项] 格式 [文件及路径]

其中的格式是指要搜索的内容格式，若缺省“文件及路径”则默认表示在当前目录下搜索。

（3）常见参数。

grep 主要选项参数如表 2.18 所示。

表 2.18　　grep 命令常见参数列表

选　项	参 数 含 义
-c	只输出匹配行的计数
-I	不区分大小写（只适用于单字符）
-h	查询多文件时不显示文件名
-l	查询多文件时只输出包含匹配字符的文件名
-n	显示匹配行及行号
-s	不显示不存在或无匹配文本的错误信息
-v	显示不包含匹配文本的所有行

（4）使用实例。

```
    [root@localhost test]# grep "hello" / -r
    Binary file ./iscit2005/备份/iscit2004.sql matches
    ./ARM_TOOLS/uClinux-Samsung/linux-2.4.x/Documentation/s390/Debugging390.txt:hello
world$2 = 0
    ...
```

在本例中，“hello”是要搜索的内容，“/ -r”是指定文件，表示搜索根目录下的所有文件。

（5）使用说明。

- 在缺省情况下，“grep”只搜索当前目录。如果此目录下有许多子目录，“grep”会以如下形式列出：“grep:sound:Is a directory”。这会使“grep”的输出难以阅读。但有以下两种解决的方法。

① 明确要求搜索子目录：grep –r（正如上例中所示）；

② 忽略子目录：grep -d skip。

- 当预料到有许多输出时，可以通过管道将其转到“less”（分页器）上阅读：如 grep "h" ./ -r |less 分页阅读。

- grep 特殊用法。

grep pattern1|pattern2 files：显示匹配 pattern1 或 pattern2 的行；

grep pattern1 files|grep pattern2：显示既匹配 pattern1 又匹配 pattern2 的行；

小知识

在文件命令中经常会使用 pattern 正则表达式，它是可以描述一类字符串的模式（Pattern），如果一个字符串可以用某个正则表达式来描述，就称这个字符和该正则表达式匹配。这和 DOS 中用户可以使用通配符“*”代表任意字符类似。在 Linux 系统上，正则表达式通常被用来查找文本的模式，以及对文本执行“搜索－替换”操作等。

正则表达式的主要参数有如下

- \: 忽略正则表达式中特殊字符的原有含义；
- ^: 匹配正则表达式的开始行；
- $: 匹配正则表达式的结束行；
- <: 从匹配正则表达式的行开始；
- >: 到匹配正则表达式的行结束；
- [ ]: 单个字符，如[A]即 A 符合要求；

- [-]：范围，如[A-Z]，即 A、B、C 一直到 Z 都符合要求；
- 。：所有的单个字符；
- *：所有字符，长度可以为 0。

9．find

（1）作用。
在指定目录中搜索文件，它的使用权限是所有用户。
（2）格式。
find [路径][选项][描述]
其中的路径为文件搜索路径，系统开始沿着此目录树向下查找文件。它是一个路径列表，相互用空格分离。若缺省路径，那么默认为当前目录。
其中的描述是匹配表达式，是 find 命令接受的表达式。
（3）常见参数。
[选项]主要参数如表 2.19 所示。

表 2.19　　find 选项常见参数列表

选　　项	参 数 含 义
-depth	使用深度级别的查找过程方式，在某层指定目录中优先查找文件内容
-mount	不在其他文件系统（如 Msdos、Vfat 等）的目录和文件中查找

[描述]主要参数如表 2.20 所示。

表 2.20　　find 描述常见参数列表

选　　项	参 数 含 义
-name	支持通配符*和?
-user	用户名：搜索文件属主为用户名（ID 或名称）的文件
-print	输出搜索结果，并且打印

（4）使用实例。

```
[root@localhost test]# find ./ -name hello*.c
./hello1.c
./iscit2005/hello2.c
```

在该实例中使用了-name 的选项支持通配符。
（5）使用说明。

- 若使用目录路径为“/”，通常需要查找较多的时间，可以指定更为确切的路径以减少查找时间。
- find 命令可以使用混合查找的方法，例如，想在/etc 目录中查找大于 500000 字节，并且在 24 小时内修改的某个文件，则可以使用-and（与）把两个查找参数链接起来组合成一个混合的查找方式，如“find /etc -size +500000c -and -mtime +1”。

10．locate

（1）作用。

用于查找文件。其方法是先建立一个包括系统内所有文件名称及路径的数据库，之后当寻找时就只需查询这个数据库，而不必实际深入档案系统之中了。因此其速度比 find 快很多。

（2）格式。

locate [选项]

（3）locate 主要选项参数如表 2.21 所示。

表 2.21　　locate 命令常见参数列表

选　项	参 数 含 义
-u	从根目录开始建立数据库
-U	在指定的位置开始建立数据库
-f	将特定的文件系统排除在数据库外，例如 proc 文件系统中的文件
-r	使用正则运算式做寻找的条件
-o	指定数据库的名称

（4）使用实例。

```
[root@localhost test]# locate issue -U ./
[root@localhost test]# updatedb
[root@localhost test]# locate -r issue*
./ARM_TOOLS/uClinux-Samsung/lib/libpam/doc/modules/pam_issue.sgml
./ARM_TOOLS/uClinux-Samsung/lib/libpam/modules/pam_issue
./ARM_TOOLS/uClinux-Samsung/lib/libpam/modules/pam_issue/Makefile
./ARM_TOOLS/uClinux-Samsung/lib/libpam/modules/pam_issue/pam_issue.c
...
```

实例中首先在当前目录下建立了一个数据库，并且在更新了数据库之后进行正则匹配查找。通过运行可以发现 locate 的运行速度非常快。

（5）使用说明。

locate 命令所查询的数据库由 updatedb 程序来更新，而 updatedb 是由 cron daemon 周期性建立的，但若所找到的档案是最近才建立或刚改名的，可能会找不到，因为 updatedb 默认每天运行一次，用户可以由修改 crontab 配置（etc/crontab）来更新周期值。

11．ln

（1）作用。

为某一个文件在另外一个位置建立一个符号链接。当需要在不同的目录用到相同的文件时，Linux 允许用户不用在每一个需要的目录下都存放一个相同的文件，而只需将其他目录下的文件用 ln 命令链接即可，这样就不必重复地占用磁盘空间。

（2）格式。

ln[选项] 目标 目录

（3）常见参数。

- –s 建立符号链接（这也是通常惟一使用的参数）。

（4）使用实例。

```
[root@localhost test]# ln -s ../genromfs-0.5.1.tar.gz ./hello
[root@localhost test]# ls -l
total 77948
lrwxrwxrwx   1 root   root    24 Jan 14 00:25 hello -> ../genromfs-0.5.1.tar.gz
```

该实例建立了当前目录的 hello 文件与上级目录之间的符号链接，可以看见，在 hello 的 ls –l 中的第一位为“l”，表示符号链接，同时还显示了链接的源文件。

（5）使用说明。

- ln 命令会保持每一处链接文件的同步性，也就是说，不论改动了哪一处，其他的文件都会发生相同的变化。
- ln 的链接分软链接和硬链接两种。

软链接就是上面所说的 ln -s ** **，它只会在用户选定的位置上生成一个文件的镜像，不会重复占用磁盘空间，平时使用较多的都是软链接。

硬链接是不带参数的 ln ** **，它会在用户选定的位置上生成一个和源文件大小相同的文件，无论是软链接还是硬链接，文件都保持同步变化。

2.1.3 压缩打包相关命令

Linux 中打包压缩的相关命令如表 2.22 所示，本书以 gzip 和 tar 为例进行讲解。

表 2.22 Linux 常见系统管理命令

命　令	命 令 含 义	格　式
bzip2	.bz2 文件的压缩（或解压缩）程序	bzip2[选项] 压缩（解压缩）的文件名
bunzip2	.bz2 文件的解压缩程序	bunzip2[选项] .bz2 压缩文件
bzip2recover	修复损坏的.bz2 文件	bzip2recover .bz2 压缩文件
gzip	.gz 文件的压缩程序	gzip [选项] 压缩（解压缩）的文件名
gunzip	解压缩被 gzip 压缩过的文件	gunzip [选项] .gz 文件名
unzip	解压缩 winzip 压缩的.zip 文件	unzip [选项] .zip 压缩文件
compress	早期的压缩或解压缩程序(压缩后文件名为.Z)	compress [选项] 文件
tar	对文件目录进行打包或解压缩	tar [选项] [打包后文件名]文件目录列表

1．gzip

（1）作用。

对文件进行压缩和解压缩，而且 gzip 根据文件类型可自动识别压缩或解压。

（2）格式。

gzip [选项] 压缩（解压缩）的文件名。

（3）常见参数。

gzip 主要选项参数如表 2.23 所示。

表 2.23 gzip 命令常见参数列表

选　项	参 数 含 义
-c	将输出信息写到标准输出上，并保留原有文件
-d	将压缩文件解压
-l	对每个压缩文件，显示下列字段：压缩文件的大小、未压缩时文件的大小、压缩比、未压缩时文件的名字
-r	查找指定目录并压缩或解压缩其中的所有文件
-t	测试，检查压缩文件是否完整
-v	对每一个压缩和解压的文件，显示文件名和压缩比

（4）使用实例。

```
[root@localhost test]# gzip portmap-4.0-54.i386.rpm
[root@localhost test]# ls
portmap-4.0-54.i386.rpm.gz
[root@localhost test]# gzip -l portmap-4.0-54.i386.rpm
compressed   uncompressed  ratio uncompressed_name
21437          25751   16.9% portmap-4.0-54.i386.rpm
```

该实例将目录下的“hello.c”文件进行压缩，选项“-l”列出了压缩比。

（5）使用说明。

- 使用 gzip 压缩只能压缩单个文件，而不能压缩目录，其选项“-d”是将该目录下的所有文件逐个进行压缩，而不是压缩成一个文件。

2．tar

（1）作用。

对文件目录进行打包或解包。

在此需要对打包和压缩这两个概念进行区分。打包是指将一些文件或目录变成一个总的文件，而压缩则是将一个大的文件通过一些压缩算法变成一个小文件。为什么要区分这两个概念呢？这是由于在 Linux 中的很多压缩程序（如前面介绍的 gzip）只能针对一个文件进行压缩，这样当想要压缩较多文件时，就要借助它的工具将这些堆文件先打成一个包，然后再用原来的压缩程序进行压缩。

（2）格式。

tar [选项] [打包后文件名]文件目录列表。

tar 可自动根据文件名识别打包或解包动作，其中打包后文件名为用户自定义的打包后文件名称，文件目录列表可以是要进行打包备份的文件目录列表，也可以是进行解包的文件目录列表。

（3）主要参数。

tar 主要选项参数如表 2.24 所示。

表 2.24　　tar 命令常见参数列表

选　项	参 数 含 义
-c	建立新的打包文件
-r	向打包文件末尾追加文件
-x	从打包文件中解出文件
-o	将文件解开到标准输出
-v	处理过程中输出相关信息
-f	对普通文件操作
-z	调用 gzip 来压缩打包文件，与-x 联用时调用 gzip 完成解压缩
-j	调用 bzip2 来压缩打包文件，与-x 联用时调用 bzip2 完成解压缩
-Z	调用 compress 来压缩打包文件，与-x 联用时调用 compress 完成解压缩

（4）使用实例。

```
[root@localhost home]# tar -cvf david.tar  david
./david/
./david/.bash_logout
./david/.bash_profile
./david/.bashrc
```

```
./david/.bash_history
./david/my/
./david/my/1.c.gz
./david/my/my.c.gz
./david/my/hello.c.gz
./david/my/why.c.gz
[root@localhost home]# ls -l david.tar
-rw-r--r--    1 root     root       10240 Jan 14 15:01 david.tar
```

该实例将“david”目录下的文件加以打包，其中选项“-v”在屏幕上输出了打包的具体过程。

```
[david@localhost david]# tar -zxvf linux-2.6.11.tar.gz
linux-2.6.11/
linux-2.6.11/drivers/
linux-2.6.11/drivers/video/
linux-2.6.11/drivers/video/aty/
…
```

该实例用选项“-z”调用 gzip，与“-x”联用时完成解压缩。

（5）使用说明。

tar 命令除了用于常规的打包之外，使用更为频繁的是用选项“-z”或“-j”调用 gzip 或 bzip2（Linux 中另一种解压工具）完成对各种不同文件的解压。

表 2.25 对 Linux 中常见类型的文件解压命令做一个总结。

表 2.25　　Linux 常见类型的文件解压命令一览表

文 件 后 缀	解 压 命 令	示　　例
.a	tar xv	tar xv hello.a
.z	Uncompress	uncompress hello.Z
.gz	Gunzip	gunzip hello.gz
.tar.Z	tar xvZf	tar xvZf hello.tar.Z
.tar.gz/.tgz	tar xvzf	tar xvzf hello.tar.gz
tar.bz2	tar jxvf	tar jxvf hello.tar.bz2
.rpm	安装：rpm –i	安装：rpm -i hello.rpm
	解压缩：rpm2cpio	解压缩：rpm2cpio hello.rpm
.deb（Debain 中的文件格式）	安装：dpkg –i	安装：dpkg -i hello.deb
	解压缩：dpkg-deb --fsys-tarfile	解 压 缩 ： dpkg-deb --fsys-tarhello hello.deb
.zip	Unzip	unzip hello.zip

2.1.4　文件比较合并相关命令

1．diff

（1）作用。

比较两个不同的文件或不同目录下的两个同名文件功能，并生成补丁文件。

（2）格式。

diff[选项] 文件 1 文件 2

diff 比较文件 1 和文件 2 的不同之处，并按照选项所指定的格式加以输出。diff 的格式分为命令格式和上下文格式，其中上下文格式又包括了旧版上下文格式和新版上下文格式，命令格式分为标准命令格式、简单命令格式及混合命令格式，它们之间的区别会在使用实例中进行详细讲解。当选项缺省时，diff 默认使用混合命令格式。

（3）主要参数。

diff 主要选项参数如表 2.26 所示。

表 2.26　　diff 命令常见参数列表

选　项	参 数 含 义
-r	对目录进行递归处理
-q	只报告文件是否有不同，不输出结果
-e，-ed	命令格式
-f	RCS（修订控制系统）命令简单格式
-c，--context	旧版上下文格式
-u，--unified	新版上下文格式
-Z	调用 compress 来压缩归档文件，与-x 联用时调用 compress 完成解压缩

（4）使用实例。

以下有两个文件 hello1.c 和 hello2.c。

```
/* hello1.c */
#include <stdio.h>
void main()
{
        printf("Hello!This is my home!\n");
}
/* hello2.c */
#include <stdio.h>
void main()
{
        printf("Hello!This is your home!\n");
}
```

以下实例主要讲解了各种不同格式的比较和补丁文件的创建方法。

① 主要格式比较。

首先使用旧版上下文格式进行比较。

```
[root@localhost david]# diff -c hello1.c hello2.c
*** hello1.c    Sat Jan 14 16:24:51 2006
--- hello2.c    Sat Jan 14 16:54:41 2006
***************
*** 1,5 ****
  #include <stdio.h>
  void main()
  {
!       printf("Hello!This is my home!\n");
  }
--- 1,5 ----
  #include <stdio.h>
```

```
 void main()
 {
!       printf("Hello!This is your home!\n");
 }
```

可以看出，用旧版上下文格式进行输出时，在显示每个有差别行的同时还显示该行的上下 3 行，区别的地方用“!”加以标出，由于示例程序较短，上下 3 行已经包含了全部代码。
接着使用新版的上下文格式进行比较。

```
[root@localhost david]# diff -u hello1.c hello2.c
--- hello1.c    Sat Jan 14 16:24:51 2006
+++ hello2.c    Sat Jan 14 16:54:41 2006
@@ -1,5 +1,5 @@
 #include <stdio.h>
 void main()
 {
-       printf("Hello!This is my home!\n");
+       printf("Hello!This is your home!\n");
 }
```

可以看出，在新版上下文格式输出时，仅把两个文件的不同之处分别列出，而相同之处没有重复列出，这样大大方便了用户的阅读。
接下来使用命令格式进行比较。

```
[root@localhost david]# diff -e hello1.c hello2.c
4c
       printf("Hello!This is your home!\n");
```

可以看出，命令符格式输出时仅输出了不同的行，其中命令符“4c”中的数字表示行编号，字母的含义为：**a** 表示**添加**，**b** 表示**删除**，**c** 表示**更改**。因此，-e 选项的命令符表示：若要把 hello1.c 变为 hello2.c，就需要把 hello1.c 的第 4 行改为显示出的“printf（“Hello!This is your home!\n”）;”。
选项“-f”和选项“-e”显示的内容基本相同，就是数字和字母的顺序相交换了，从以下的输出结果可以看出。

```
[root@localhost david]# diff -f hello1.c hello2.c
c4
       printf("Hello!This is your home!\n");
```

在 diff 选项缺省的情况下，输出结果如下所示。

```
[root@localhost david]# diff hello1.c hello2.c
4c4
<       printf("Hello!This is my home!\n");
---
>       printf("Hello!This is your home!\n");
```

可以看出，diff 缺省情况下的输出格式充分显示了如何将 hello1.c 转化为 hello2.c，即通过“4c4”实现。
② 创建补丁文件（也就是差异文件）是 diff 的功能之一，不同的选项格式可以生成与之相对应的补丁文件，如下面扔例子所示。

```
[root@localhost david]# diff hello1.c hello2.c >hello.patch
[root@localhost david]# vi hello.patch
4c4
<       printf("Hello!This is my home!\n");
```

```
---
>       printf("Hello!This is your home!\n");
```

可以看出，使用缺省选项创建补丁文件的内容和前面使用缺省选项的输出内容是一样的。

小知识

上例中所使用的”>“是输出重定向。通常在 Linux 上执行一个 shell 命令行时，会自动打开 3 个标准文件：标准输入文件（stdin），即通常对应终端的键盘；标准输出文件（stdout）和标准错误输出文件（stderr），前两个文件都对应终端的屏幕。进程将从标准输入文件中得到输入数据，并且将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。这就是通常使用的标准输入/输出方式。

直接使用标准输入/输出文件存在以下问题：首先，用户输入的数据只能使用一次。当下次希望再次使用这些数据时就不得不重新输入。同样，用户对输出信息不能做更多的处理，只能等待程序的结束。

为了解决上述问题，Linux 系统为输入、输出的信息传送引入了两种方式：输入/输出重定向机制和管道（在 1.3.1 的小知识中已有介绍）。其中，输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以不来自键盘，而来自一个指定的文件。同样，输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就可以不显示在屏幕上，而是写入到指定文件中。就如上述例子中所用到的把“diff hello1.c hello2.c”的结果重定向到 hello.patch 文件中。这就大大增加了输入/输出的灵活性。

2．patch

（1）作用。

命令跟 diff 配合使用，把生成的补丁文件应用到现有代码上。

（2）格式。

patch [选项] [待 patch 的文件[patch 文件]]。

常用的格式为：patch -pnum [patch 文件]，其中的-pnum 是选项参数，在后面会详细介绍。

（3）常见参数。

patch 主要选项参数如表 2.27 所示。

表 2.27　　patch 命令常见参数列表

选　项	参 数 含 义
-b	生成备份文件
-d	把 dir 设置为解释补丁文件名的当前目录
-e	把输入的补丁文件看作是 ed 脚本
-pnum	剥离文件名中的前 NUM 个目录部分
-t	在执行过程中不要求任何输入
-v	显示 patch 的版本号

以下对-punm 选项进行说明。

首先查看以下示例（对分别位于 xc.orig/config/cf/Makefile 和 xc.bsd/config/cf/Makefile 的文件使用 patch 命令）。

```
diff -ruNa xc.orig/config/cf/Makefile xc.bsd/config/cf/Makefile
```

以下是 patch 文件的头标记。

```
--- xc.orig/config/cf/Imake.cf Fri Jul 30 12:45:47 1999
+++ xc.new/config/cf/Imake.cf Fri Jan 21 13:48:44 2000
```

这个 patch 如果直接应用，那么它会去找“xc.orig/config/cf”目录下的 Makefile 文件，假如用户源码树的根目录是缺省的 xc 而不是 xc.orig，则除了可以把 xc.orig 移到 xc 处之外，还有什么简单的方法应用此 patch 吗？NUM 就是为此而设的：patch 会把目标路径名剥去 NUM 个“/”，也就是说，在此例中，-p1 的结果是 config/cf/Makefile，-p2 的结果是 cf/Makefile。因此，在此例中就可以用命令 cd xc；patch _p1 < /pathname/xxx.patch 完成操作。

（4）使用实例。

```
[root@localhost david]# diff hello1.c hello2.c >hello1.patch
[root@localhost david]# patch ./hello1.c < hello1.patch
patching file ./hello1.c
[root@localhost david]# vi hello1.c
#include <stdio.h>
void main()
{
      printf("Hello!This is your home!\n");
}
```

在该实例中，由于 patch 文件和源文件在同一目录下，因此直接给出了目标文件的目录，在应用了 patch 之后，hello1.c 的内容变为了 hello2.c 的内容。

（5）使用说明。

- 如果 patch 失败，patch 命令会把成功的 patch 行补上其差异，同时（无条件）生成备份文件和一个.rej 文件。.rej 文件里没有成功提交的 patch 行，需要手工打上补丁。这种情况在源码升级的时候有可能会发生。
- 在多数情况下，patch 程序可以确定补丁文件的格式，当它不能识别时，可以使用-c、-e、-n 或者-u 选项来指定输入的补丁文件的格式。由于只有 GNU patch 可以创建和读取新版上下文格式的 patch 文件，因此，除非能够确定补丁所面向的只是那些使用 GNU 工具的用户，否则应该使用旧版上下文格式来生成补丁文件。
- 为了使 patch 程序能够正常工作，需要上下文的行数至少是 2 行（即至少是有一处差别的文件）。

2.1.5 网络相关命令

Linux 下网络相关的常见命令如表 2.28 所示，本书仅以 ifconfig 和 ftp 为例进行说明。

表 2.28 Linux 下网络相关命令

选　项	参 数 含 义	常见选项格式
netstat	显示网络连接、路由表和网络接口信息	netstat [-an]
nslookup	查询一台机器的 IP 地址和其对应的域名	nslookup [IP 地址/域名]
finger	查询用户的信息	finger [选项] [使用者] [用户@主机]
ping	用于查看网络上的主机是否在工作	ping [选项] 主机名/IP 地址
ifconfig	查看和配置网络接口的参数	ifconfig [选项] [网络接口]
ftp	利用 ftp 协议上传和下载文件	在本节中会详细讲述
telnet	利用 telnet 协议访问主机	telent [选项] [IP 地址/域名]
ssh	利用 ssh 登录对方主机	ssh [选项] [IP 地址]

1．ifconfig

（1）作用。

用于查看和配置网络接口的地址和参数，包括 IP 地址、网络掩码、广播地址，它的使用权限是超级用户。

（2）格式。

ifconfig 有两种使用格式，分别用于查看和更改网络接口。

① ifconfig [选项] [网络接口]：用来查看当前系统的网络配置情况。

② ifconfig 网络接口 [选项] 地址：用来配置指定接口（如 eth0、eth1）的 IP 地址、网络掩码、广播地址等。

（3）常见参数。

ifconfig 第二种格式的常见选项参数如表 2.29 所示。

表 2.29　　ftp 命令选项的常见参数列表

选　项	参 数 含 义
-interface	指定的网络接口名，如 eth0 和 eth1
up	激活指定的网络接口卡
down	关闭指定的网络接口
broadcast address	设置接口的广播地址
poin to point	启用点对点方式
address	设置指定接口设备的 IP 地址
netmask address	设置接口的子网掩码

（4）使用实例。

首先，在本例中使用 ifconfig 的第一种格式来查看网络接口配置情况。

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:08:02:E0:C1:8A
          inet addr:192.168.1.70  Bcast:192.168.1.255
Mask:255.255.255.0
          inet6 addr: fe80::208:2ff:fee0:c18a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:26931 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3209 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6669382 (6.3 MiB)  TX bytes:321302 (313.7 KiB)
          Interrupt:11

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2537 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2537 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2093403 (1.9 MiB)  TX bytes:2093403 (1.9 MiB)
```

可以看出，使用 ifconfig 的显示结果中详细列出了所有活跃接口的 IP 地址、硬件地址、广播地址、子网掩码、回环地址等。

```
[root@localhost workplace]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:08:02:E0:C1:8A
```

```
          inet addr: 192.168.1.70  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::208:2ff:fee0:c18a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27269 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3212 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6698832 (6.3 MiB)  TX bytes:322488 (314.9 KiB)
          Interrupt:11
```

在此例中，通过指定接口显示出对应接口的详细信息。另外，用户还可以通过指定参数“-a”来查看所有接口（包括非活跃接口）的信息。

接下来的示例指出了如何使用 ifconfig 的第二种格式来改变指定接口的网络参数配置。

```
[root@localhost ~]# ifconfig eth0 down
[root@localhost ~]# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1931 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1931 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2517080 (2.4 MiB)  TX bytes:2517080 (2.4 MiB)
```

在此例中，通过将指定接口的状态设置为 DOWN，暂时停止该接口的工作。

```
[root@localhost ~]# ifconfig eth0 210.25.132.142 netmask 255.255.255.0
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:08:02:E0:C1:8A
          inet addr:210.25.132.142  Bcast:210.25.132.255  Mask:255.255.255.0
          inet6 addr: fe80::208:2ff:fee0:c18a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1722 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:147382 (143.9 KiB)  TX bytes:398 (398.0 b)
          Interrupt:11
...
```

从上例可以看出，ifconfig 改变了接口 eth0 的 IP 地址、子网掩码等，在之后的 ifconfig 查看中可以看出确实发生了变化。

（5）使用说明。

用 ifconfig 命令配置的网络设备参数不重启就可生效，但在机器重新启动以后将会失效，除非在网络接口配置文件中进行修改。

2．ftp

（1）作用。

该命令允许用户利用 ftp 协议上传和下载文件。

（2）格式。

ftp [选项] [主机名/IP]。

ftp 相关命令包括使用命令和内部命令，其中使用命令的格式如上所列，主要用于登录到 ftp 服务器。内部命令是指成功登录后进行的一系列操作，下面会详细列出。若用户缺省“主机名/IP”，则可在转入到 ftp 内部命令后继续选择登录。

（3）常见参数。

ftp 常见选项参数如表 2.30 所示。

表 2.30　　ftp 命令选项常见参数列表

选　　项	参 数 含 义
-v	显示远程服务器的所有响应信息
-n	限制 ftp 的自动登录
-d	使用调试方式
-g	取消全局文件名

ftp 常见内部命令如表 2.31 所示。

表 2.31　　ftp 命令常见内部命令

命　　令	命 令 含 义
account[password]	提供登录远程系统成功后访问系统资源所需的补充口令
ascii	使用 ASCII 类型传输方式，为缺省传输模式
bin/ type binary	使用二进制文件传输方式（嵌入式开发中的常见方式）
bye	退出 ftp 会话过程
cd remote-dir	进入远程主机目录
cdup	进入远程主机目录的父目录
chmod mode file-name	将远程主机文件 file-name 的存取方式设置为 mode
close	中断与远程服务器的 ftp 会话（与 open 对应）
delete remote-file	删除远程主机文件
debug[debug-value]	设置调试方式，　显示发送至远程主机的每条命令
dir/ls[remote-dir][local-file]	显示远程主机目录，并将结果存入本地文件 local-file
disconnection	同 close
get remote-file[local-file]	将远程主机的文件 remote-file 传至本地硬盘的 local-file
lcd[dir]	将本地工作目录切换至 dir
mdelete[remote-file]	删除远程主机文件
mget remote-files	传输多个远程文件
mkdir dir-name	在远程主机中建立一个目录
mput local-file	将多个文件传输至远程主机
open host[port]	建立与指定 ftp 服务器的连接，可指定连接端口
passive	进入被动传输方式（在这种模式下，数据连接是由客户程序发起的）
put local-file[remote-file]	将本地文件 local-file 传送至远程主机
reget remote-file[local-file]	类似于 get，但若 local-file 存在，则从上次传输中断处继续传输
size file-name	显示远程主机文件大小
system	显示远程主机的操作系统类型

（4）使用实例。

首先，在本例中使用 ftp 命令访问“ftp://study.byr.edu.cn”站点。

```
[root@localhost ~]# ftp study.byr.edu.cn
Connected to study.byr.edu.cn.
220 Microsoft FTP Service
500 'AUTH GSSAPI': command not understood
500 'AUTH KERBEROS_V4': command not understood
KERBEROS_V4 rejected as an authentication type
Name (study.byr.edu.cn:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 Anonymous user logged in.
Remote system type is Windows_NT.
```

注意 由于该站点可以匿名访问，因此，在用户名处输入 anonymous，在 Password 处输入任意一个 e-mail 地址即可登录成功。

```
ftp> dir
227 Entering Passive Mode (211,68,71,83,11,94).
125 Data connection already open; Transfer starting.
11-20-05  05:00PM       <DIR>          Audio
12-04-05  09:41PM       <DIR>          BUPT_NET_Material
01-07-06  01:38PM       <DIR>          Document
11-22-05  03:47PM       <DIR>          Incoming
01-04-06  11:09AM       <DIR>          Material
226 Transfer complete.
```

以上使用 ftp 内部命令 dir 列出了在该目录下文件及目录的信息。

```
ftp> cd /Document/Wrox/Wrox.Beginning.SQL.Feb.2005.eBook-DDU
250 CWD command successful.
ftp> pwd
257 "/Document/Wrox/Wrox.Beginning.SQL.Feb.2005.eBook-DDU" is current directory.
```

以上实例通过 cd 命令进入相应的目录，可通过 pwd 命令进行验证。

```
ftp> lcd /root/workplace
Local directory now /root/workplace
ftp> get d-wbsq01.zip
local: d-wbsq01.zip remote: d-wbsq01.zip
200 PORT command successful.
150 Opening ASCII mode data connection for d-wbsq01.zip(1466768 bytes).
WARNING! 5350 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
1466768 bytes received in 1.7 seconds (8.6e+02 Kbytes/s)
```

接下来通过 lcd 命令首先改变用户的本地工作目录，也就是希望下载或上传的工作目录，接着通过 get 命令进行下载文件。由于 ftp 默认使用 ASCII 模式，因此，若希望改为其他模式如“bin”，直接输入 bin 即可，如下所示：

```
ftp> bin
200 Type set to I.
```

```
ftp> bye
221
```

最后用 bye 命令退出 ftp 程序。

（5）使用说明

- 若是需要匿名登录，则在“Name (**.**.**.**):”处键入 anonymous，在“Password:”处键入自己的 E-mail 地址即可。
- 若要传送二进制文件，务必要把模式改为 bin。

2.2 Linux 启动过程详解

在了解了 Linux 的常见命令之后，下面详细讲解 Linux 的启动过程。Linux 的启动过程包含了 Linux 工作原理的精髓，而且在嵌入式开发过程中非常需要这方面的知识。

2.2.1 概述

用户开机启动 Linux 过程如下：

（1）当用户打开 PC（intel CPU）的电源时，CPU 将自动进入实模式，并从地址 0xFFFF0000 开始自动执行程序代码，这个地址通常是 ROM-BIOS 中的地址。这时 BIOS 进行开机自检，并按 BIOS 中设置的启动设备（通常是硬盘）进行启动，接着启动设备上安装的引导程序 lilo 或 grub 开始引导 Linux（也就是启动设备的第一个扇区），这时，Linux 才获得了启动权。

（2）第二阶段，Linux 首先进行内核的引导，主要完成磁盘引导、读取机器系统数据、实模式和保护模式的切换、加载数据段寄存器以及重置中断描述符表等。

（3）第三阶段执行 init 程序（也就是系统初始化工作），init 程序调用了 rc.sysinit 和 rc 等程序，而 rc.sysinit 和 rc 在完成系统初始化和运行服务的任务后，返回 init。

（4）第四阶段，init 启动 mingetty，打开终端供用户登录系统，用户登录成功后进入了 shell，这样就完成了从开机到登录的整个启动过程。

Linux 启动总体流程如图 2.2 所示，其中的 4 个阶段分别由同步棒隔开。第一阶段不涉及 Linux 自身的启动过程，下面分别对第二和第三阶段进行详细讲解。

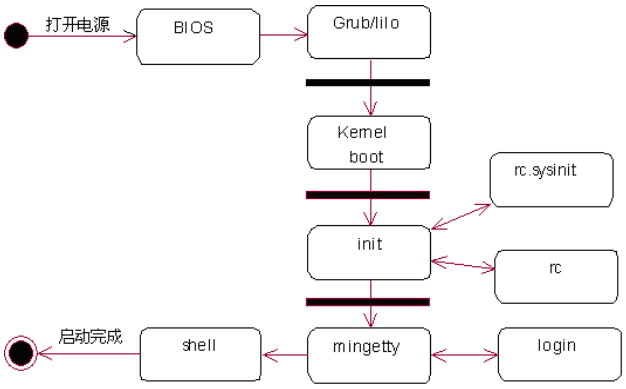

图 2.2　Linux 启动总体流程图

2.2.2 内核引导阶段

在 grub 或 lilo 等引导程序成功完成引导 Linux 系统的任务后，Linux 就从它们手中接管了 CPU 的控制权。用户可以从 www.kernel.org 上下载最新版本的源码进行阅读，其目录为：linux-2.6.*.*/arch/i386/boot。在启动过程中主要用到该目录下的几个文件：bootsect.S、setup.S 以及 compressed 子目录下的 head.S 等。

Linux 的内核通常是压缩过的，包括上述提到的那几个重要的汇编程序，它们都是在压缩内核 vmlinuz 中的。Linux 中提供的内核包含了众多驱动和功能，容量较大，压缩内核可以节省大量的空间，压缩的内核在启动时可以对自身进行解包。

（1）bootsect 阶段

当 grub 读入 vmlinuz 后，会根据 bootsect（512 字节）把它自身和 setup 程序段读到以不大于 0x90000 开始的的内存里（注意：在以往的引导协议里是放在 0x90000，但现在有所变化），然后 grub 会跳过 bootsect 那 512 字节的程序段，直接运行 setup 里的第一跳指令。就是说 bzImage 里 bootsect 的程序没有再被执行了，而 bootsect.S 在完成了指令搬移以后就退出了。之后执行权就转到了 setup.S 的程序中。

（2）setup 阶段。

setup.S 的主要功能是利用 ROM BIOS 中断读取机器系统数据，并将系统参数（包括内存、磁盘等）保存到以 0x90000～0x901FF 开始的内存中。

此外，setup.S 还将 video.S 中的代码包含进来，检测和设置显示器和显示模式。

最后，它还会设置 CPU 的控制寄存器 CR0（也称机器状态字），从而进入 32 位保护模式运行，并跳转到绝对地址为 0x100000（虚拟地址 0xC0000000+0x100000）的位置。当 CPU 跳到 0x100000 时，将执行“arch/i386/kernel/head.S”中的 startup_32。

（3）head.S 阶段。

当运行到 head.S 时，系统已经运行在保护模式，而 head.S 完成的一个重要任务就是将内核解压。内核是通过压缩的方式放在内存中的，head.S 通过调用 misc.c 中定义的 decompress_kernel()函数，将内核 vmlinuz 解压到 0x100000。

接下来 head.S 程序完成寄存器、分页表的初始化工作，但要注意的是，这个 head.S 程序与完成解压缩工作的 head.S 程序是不同的，它在源代码中的位置是 arch/i386/kernel/head.S。

在完成了初始化之后，head.S 就跳转到 start_kernel()函数中去了。

（4）main.c 阶段。

start_kernel()是“init/main.c”中定义的函数，start_kernel()调用了一系列初始化函数，进行内核的初始化工作。要注意的是，在初始化之前系统中断仍然是被屏蔽的，另外内核也处于被锁定状态，以保证只有一个 CPU 用于 Linux 系统的启动。

在 start_kernel()的最后，调用了 init()函数，也就是下面要讲述的 init 阶段。

2.2.3 init 阶段

在加载了内核之后，由内核执行引导的第一个进程是 init 进程，该进程号始终是“1”。init 进程根据其配置文件“/etc/inittab”主要完成系统的一系列初始化的任务。由于该配置文件是 init 进程执行的惟一依据，因此先对它的格式进行统一讲解。

inittab 文件中除了注释行外，每一行都有如下格式：

```
id:runlevels:action:process
```

（1）id。

id 是配置记录标识符，由 1～4 个字符组成，对于 getty 或 mingetty 等其他 login 程序项，要求 id 与 tty 的编号相同，否则 getty 程序将不能正常工作。

（2）runlevels。

runlevels 是运行级别记录符，一般使用 0～6 以及 S 和 s。其中，0、1、6 运行级别为系统保留：0 作为 shutdown 动作，1 作为重启至单用户模式，6 为重启；S 和 s 意义相同，表示单用户模式，且无需 inittab 文件，因此也不在 inittab 中出现。7～9 级别也是可以使用的，传统的 UNIX 系统没有定义这几个级别。

runlevel 可以是并列的多个值，对大多数 action 来说，仅当 runlevel 与当前运行级别匹配成功才会执行。

（3）action。

action 字段用于描述系统执行的特定操作，它的常见设置有：initdefault、sysinit、boot、bootwait、respawn 等。

initdefault 用于标识系统缺省的启动级别。当 init 由内核激活以后，它将读取 inittab 中的 initdefault 项，取得其中的 runlevel，并作为当前的运行级别。如果没有 inittab 文件，或者其中没有 initdefault 项，init 将在控制台上请求输入 runlevel。

sysinit、boot、bootwait 等 action 将在系统启动时无条件运行，忽略其中的 runlevel。

respawn 字段表示该类进程在结束后会重新启动运行。

（4）process。

process 字段设置启动进程所执行的命令。

以下结合笔者系统中的 inittab 配置文件详细讲解该配置文件完成的功能。

1．确定用户登录模式

在“/etc/inittab”中列出了如下所示的登录模式，主要有单人维护模式、多用户无网络模式、文字界面多用户模式、X-Windows 多用户模式等。其中的单人维护模式（run level 为 1）类似于 Windows 中的“安全模式”，在这种情况下，系统不加载复杂的模式从而使系统能够正常启动。在这些模式中最为常见的是 3 或 5，其中本系统中默认的为 5，也就是 X-Windows 多用户模式。以下是在“/etc/inittab”文件中设置系统启动模式的部分。

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode （文本界面启动模式）
#   4 - unused
#   5 - X11 （图形界面启动模式）
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

2．执行/etc/rc.d/rc.sysinit

在确定了登录模式之后，就要开始将 Linux 的主机信息读入系统，其过程是通过运行“/etc/rc.d/rc.sysinit”脚本而完成的。查看此文件可以看出，在这里确定了默认路径、主机名称、“/etc/sysconfig/network”中所记录的网络信息等。以下是在“/etc/inittab”文件中运行该脚本的部分。

```
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
```

3．加载内核的外挂模块，执行各运行级别的脚本以及进入用户登录界面

在此，主要是读取模块加载配置文件（/etc/modules.conf），以确认需要加载哪些模块。接下来会根据不同的运行级（run level），通过带参数（运行级）运行“/etc/rc.d/rc”脚本，加载不同的模块，启动系统服务。init 进程会等待（wait）“/etc/rc.d/rc”脚本的返回。系统还需要配置一些异常关机的处理部分，最后通过“/sbin/mingetty”打开几个虚拟终端（tty1~tty6），用于用户登录。如果运行级为 5（图形界面启动），则运行 xdm 程序，给用户提供 xdm 图形界面的登录方式。如果在本地打开一个虚拟终端，当这个终端超时没有用户登录或者太久没有用户击键时，该终端会退出执行，脚本中的“respawn”即告诉 init 进程重新打开该终端，否则在经过一段时间之后，我们会发现这个终端消失了，无法利用 ALT+Fn 切换。

以下是“/etc/inittab”文件中的相应部分。

```
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

2.3 Linux 系统服务

init 进程的作用是启动 Linux 系统服务（也就是运行在后台的守护进程）。Linux 的系统服务包括两种，第一种是独立运行的系统服务，它们常驻内存中，自开机后一直运行着（如 httpd），具有很快的响应速度；第二种是由 xinet 设定的服务。xinet 能够同时监听多个指定的端口，在接受用户请求时，它能够根据用户请求的端口不同，启动不同的网络服务进程来处理这些用户请求。因此，可以把 xinetd 看作一个启动服务的管理服务器，它决定把一个客户请求交给哪个程序处理，然后启动相应的守护进程。下面分别介绍这两种系统服务。

2.3.1 独立运行的服务

独立运行的系统服务的启动脚本都放在目录“/etc/rc.d/init.d/”中。如笔者系统中的系统服务的启动脚本有：

```
[root@localhost init.d]# ls /etc/rc.d/init.d
acpid dc_client iptables named pand rpcsvcgssd tux
anacron dc_server irda netdump pcmcia saslauthd vncserver
apmd diskdump irqbalance netfs portmap sendmail vsftpd
arptables_jf dovecot isdn netplugd psacct single watchquagga
atd dund killall network rawdevices smartd winbind
autofs firstboot kudzu NetworkManager readahead smb xfs
…
```

为了指定特定运行级别服务的开启或关闭，系统的各个不同运行级别都有不同的脚本文件，其目录为“/etc/rc.d/rcN.d”，其中的 N 分别对应不用的运行级别。读者可以进入各个不同的运行级别目录，查看相应服务是在开启还是关闭状态，如进入“/rc3.d”目录中的文件如下所示：

```
[root@localhost rc3.d]# ls /etc/rc.d/rc3.d
K02NetworkManager K35winbind K89netplugd S10networ S28autofs S95anacron
K05saslauthd K36lisa K90bluetooth S12syslog S40smartd S95atd
K10dc_server K45named K94diskdump S13irqbalance S44acpid S97messagebus
K10psacct K50netdump K99microcode_ctl S13portmap S55cups S97rhnsd
…
```

可以看到，每个对应的服务都以“K”或“S”开头，其中的 K 代表关闭（kill），其中的 S 代表启动（start），用户可以使用命令“+start|stop|status|restart”来对相应的服务进行操作。
在执行完相应的 rcN.d 目录下的脚本文件后，init 最后会执行 rc.local 来启动本地服务，因此，用户若想把某些非系统服务设置为自启动，可以编辑 rc.local 脚本文件，加上相应的执行语句即可。
另外，读者还可以使用命令“service+系统服务+操作”来方便地实现相应服务的操作，如下所示：

```
[root@localhost xinetd.d]# service xinetd restart
停止 xinetd:                                          [  确定  ]
开启 xinetd:                                          [  确定  ]
```

2.3.2 xinetd 设定的服务

xinetd 管理系统中不经常使用的服务，这些服务程序只有在有请求时才由 xinetd 服务负责启动，一旦运行完毕服务自动结束。xinetd 的配置文件为“/etc/xinetd.conf”，它对 xinet 的默认参数进行了配置：

```
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
     instances              = 60
     log_type               = SYSLOG authpriv
     log_on_success           = HOST PID
     log_on_failure           = HOST
      cps                  = 25 30
```

```
}
includedir /etc/xinetd.d
```

从该配置文件的最后一行可以看出，xinetd 启动“/etc/xinetd.d”为其配置文件目录。在对应的配置文件目录中可以看到每一个服务的基本配置，如 tftp 服务的配置脚本文件如下：

```
service tftp
{
      socket_type  = dgram （数据报格式）
      protocol     = udp （使用 UDP 传输）
      wait         = yes
      user         = root
      server       = /usr/sbin/in.tftpd
      server_args  = -s /tftpboot
      disable      = yes （不启动）
      per_source   = 11
      cps          = 100 2
      flags        = IPv4
}
```

2.3.3 系统服务的其他相关命令

除了在本节中提到的 service 命令之外，与系统服务相关的命令还有 chkconfig，它也是一个很好的工具，能够为不同的系统级别设置不同的服务。

（1）chkconfig --list（注意在 list 前有两个小连线）：查看系统服务设定。

示例：

```
[root@localhost xinetd.d]# chkconfig --list
sendmail          0:关闭  1:关闭  2:打开  3:打开  4:打开  5:打开  6:关闭
snmptrapd         0:关闭  1:关闭  2:关闭  3:关闭  4:关闭  5:关闭  6:关闭
gpm               0:关闭  1:关闭  2:打开  3:打开  4:打开  5:打开  6:关闭
syslog            0:关闭  1:关闭  2:打开  3:打开  4:打开  5:打开  6:关闭
…
```

（2）chkconfig --level N [服务名称] 指定状态：将指定级别的某个系统服务配置为指定状态 （打开/关闭）。

```
[root@localhost xinetd.d]# chkconfig --list|grep ntpd
ntpd              0:关闭  1:关闭  2 关闭  3:关闭  4:关闭  5:关闭  6:关闭
[root@localhost ~]# chkconfig --level 3 ntpd on
[root@localhost ~]# chkconfig --list|grep ntpd
ntpd              0:关闭  1:关闭  2:关闭  3:打开  4:关闭  5:关闭  6:关闭
```

另外，在 2.1.1 节系统命令列表中指出的 setup 程序中也可以设定，而且是图形界面，操作较为方便，读者可以自行尝试。

2.4 实验内容

2.4.1 在 Linux 下解压常见软件

1．实验目的

在 Linux 下安装一个完整的软件（嵌入式 Linux 的必备工具——交叉编译工具），掌握 Linux 常见命令，学会设置环境变量，同时搭建起嵌入式 Linux 的交叉编译环境（关于交叉编译的具体概念在本书后面会详细讲解），为今后的实验打下良好的基础。

2．实验内容

在 Linux 中解压 cross-3.3.2.tar.bz2，并添加到系统环境变量中去。

3．实验步骤

（1）将光盘中的 cross-3.3.2.tar.bz2 复制到 Windows 下的任意盘中。
（2）重启机器转到 Linux 下，并用普通用户身份登录。
（3）打开“终端”，并切换到超级用户模式下。
命令为：su - root
（4）查看 cross-3.3.2.tar.bz2 所在的 Windows 下对应分区的格式，并记下其文件设备名称，如“/dev/hda1”等。
命令为：fdisk -l
（5）使用 mkdir 命令在“/mnt”新建子目录作为挂载点。
命令为：mkdir /mnt/win
（6）挂载 Windows 相应分区。
若是 vfat 格式，则命令为：mount –t vfat /dev/hda* /mnt/win。

注意	由于 ntfs 格式在 Linux 的早期版本中是不安全的，只能读，不能写，因此最好把文件放到 fat32 格式的文件系统中。

（7）进入挂载目录，查看是否确实挂载上。
命令为：cd /mnt/win；ls
（8）在/usr/local 下建一个名为 arm 的目录。
命令为：mkdir /usr/local/arm
（9）将 cross-3.3.2.tar.bz2 复制到刚刚创建的目录中。
命令为：cp　/mnt/win/cross-3.3.2.tar.bz2　/usr/local/arm

注意	若 cross-3.3.2.tar.bz2 在当前目录中，则可将命令简写为：cp ./cross-3.3.2.tar.bz2 /usr/local/arm

（10）将当前工作目录转到“/usr/local/arm”下。
命令为：cd　/usr/local/arm

想一想	为什么要将此目录创建在“/usr/local”下？

（11）解压缩该软件包。
命令为：tar　jxvf　cross-3.3.2.tar.bz2
（12）将此目录下的/bin 目录添加到环境变量中去。
命令为：export　PATH=/usr/local/arm/3.3.2/bin :$PATH

注意	用此方法添加的环境变量在掉电后会丢失，因此，可以在“/etc/bashrc”的最后一行添加以上命令。

（13）查看该路径是否已添加到环境变量中。
命令为：echo $PATH

4．实验结果

成功搭建了嵌入式 Linux 的交叉编译环境，熟悉 Linux 下常用命令，如 su、mkdir、mount、cp、tar 等，并学会添加环境变量，同时对 Linux 的目录结构有了更进一步的理解。

2.4.2 定制 Linux 系统服务

1．实验目的

通过定制 Linux 系统服务，进一步理解 Linux 的守护进程，能够更加熟练运用 Linux 操作基本命令，同时也加深对 init 进程的了解和掌握。

2．实验内容

查看 Linux 系统服务，并定制其系统服务。

3．实验步骤

（1）查看系统的默认运行级别。
命令为：cat　/etc/inittab（假设当前运行级别为 N）
（2）进入相应级别的服务脚本目录，查看哪些服务是系统启动的独立运行的服务，并做下记录。
命令为：cd　/etc/rc.d/rcN.d
（3）利用命令查看系统开机自启动服务，与上次查看结果进行比较，找出其中的区别，并思考其中的原因。
命令为：chkconfig　–list
（4）记录 chkconfig　–list 命令中由 xinet 管理的服务，并将其中启动的服务做下记录。
（5）进入 xinet 配置管理的相应目录，查看是否与 chkconfig　–list 所得的结果相吻合，并查看相应脚本文件。
命令为：cd　/etc/xinetd.d
（6）将 sshd 服务停止。
命令为：service　sshd　stop
（7）将 sshd 服务设置为开机不启动。
命令为：chkconfig　–level　N　sshd　stop
（8）查看该设置是否生效。
命令为：chkconfig　–list
（9）查看系统中所有服务及其端口号列表。
命令为：cat　/etc/services
（10）将 sshd 服务端口号改为 4022。
命令为：vi　/etc/services
（11）重启 sshd 服务，验证所改的端口号是否生效。
命令为：service　sshd　start
（12）重启 Linux 系统，验证所改的服务开机启动是否生效。

4．实验结果分析

本实验通过验证Linux系统服务的启动状态，进一步明确Linux系统服务启动的流程，更深入地理解了Linux系统操作。
本实验还通过定制 Linux 系统服务 sshd 的开机启动状态和端口号，熟悉了 Linux 的系统定制步骤。

2.5 本章小结

本章首先讲解了 Linux 操作的基本命令，这些命令是使用 Linux 的基础。Linux 基本命令包括用户系统相关命令、文件目录相关命令、压缩打包相关命令、比较合并相关命令以及网络相关命令。着重介绍了每一类命令中有代表性的重要命令及其用法，并给出了具体实例，对其他命令列出了其使用方法。希望读者能举一反三、灵活应用。
接下来，本章讲解了 Linux 启动过程，这部分的内容比较难，但对深入理解 Linux 系统是非常有帮助的，希望读者能反复阅读。
最后，本章还讲解了 Linux 系统服务，包括独立运行的服务和 xinetd 设定的服务，并且讲解了 Linux 系统中设定服务的常用方法。
本章安排了两个实验，实验一通过一个完整的操作使读者能够熟练使用 Linux 的基本命令，实验二讲解了如何定制 Linux 系统服务，希望读者能够认真动手实践。

2.6 思考与练习

1．更改目录的名称，如把/home/david 变为/home/john。
2．如何将文件属性变为-rwxrw-r--？
3．下载最新 Linux 源码，并解压缩至/usr/src 目录下。
4．修改 Telnet、FTP 服务的端口号。

联系方式

集团官网：www.hqyj.com　　嵌入式学院：www.embedu.org　　移动互联网学院：www.3g-edu.org

企业学院：www.farsight.com.cn　　物联网学院：www.topsight.cn　　研发中心：dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路银海大厦 A 座 8 层，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-22193762

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218