

第 3 章 系统的基本结构

Linux 操作系统从结构上划分为内核、运行期库和系统程序、SHELL 及实用工具程序四部分。本章主要从整体上讲述 Linux 系统的基本结构，主要包括 Linux 控制台的概念与使用、Linux 系统与硬件的关系、系统的文件与目录结构、系统运行级的概念与含义、系统启动与关闭过程等。通过对本章内容的学习，读者可以对 Linux 系统有一个清晰的概念和全新的认识。

3.1 Linux 控制台的使用

Linux 中的所有管理任务都可以在控制台中完成。许多情况下，在控制台下使用程序比使用图形化的程序更快捷，而且还可以实现额外的功能。此外，所有的控制台任务都可以写到脚本中，这样就可以自动地执行相应的任务。为了真正地驾驭 Linux 系统，必须掌握在控制台环境下如何工作。

3.1.1 进入控制台

如果系统引导进入文本模式（为了降低服务开销，服务器通常是这样配置的），那么用户在以文本形式登录后就已经在控制台环境中了。在 Linux 系统中，通常可以使用快捷键 Alt+（F1~F6）切换到另外的控制台。

每一个控制台都是系统中一个完全独立的会话，每一个控制台都可以分别由完全不同的用户来控制。例如，某用户可能在控制台 1 中以 root 身份登录，而在控制台 2 中以 joeuser 身份登录。两个控制台在各自的用户空间中运行不同的程序。用户可以在某一虚拟控制台上运行的工作尚未结束时，切换到另一虚拟控制台来开始另一项工作。虚拟控制台可使同一用户同时在多个不同的控制台上工作，真正体现 Linux 系统多用户的特性。

3.1.2 Linux 控制台命令

在控制台可以使用的命令很多，其中有一些实际上只在编写脚本时才会用到。在 Linux 中，所有的命令和选项都区分大小写，控制台命令几乎全都是小写的。常用的控制台命令有以下几类：安装和登录命令、文件处理命令、系统管理命令、网络操作命令、系统安全相关命令及其他命令。

1. login

(1) 作用

login 的作用是登录系统，它的使用权限是所有用户。

(2) 格式

```
login [name] [-p] [-h 主机名称]
```

(3) 主要参数

-p: 通知 `login` 保持现在的环境参数。

-h: 用来向远程登录的主机传输用户名。

如果选择用命令行模式登录 Linux，那么看到的第一个 Linux 命令就是 `login`。

一般界面如下：

```
Mandrake Linux release 9.1(Bamboo) for i586
kernel 2.4.21-0.13mdk on i686 / tty1
localhost login:root
password:
```

上面代码中，第一行是 Linux 发行版本号，第二行是内核版本号和登录的虚拟控制台，用户在第三行输入登录名，按 Enter 键在 Password 后输入账户密码，即可登录系统。出于安全考虑，输入账户密码时字符不会在屏幕上回显，光标也不移动。登录后会看到下面这个界面（以超级用户为例）：

```
[root@localhost root]#
last login: Tue, Nov 18 10:00:55 on vc/1
```

上面显示的是登录星期、月、日、时间和使用的虚拟控制台。

2. shutdown

(1) 作用

`shutdown` 命令的作用是关闭计算机，它的使用权限是超级用户。

(2) 格式

```
shutdown [-h] [-i] [-k] [-m] [-t]
```

(3) 主要参数

-h: 关机后关闭电源。

-i: 关机时显示系统信息。

-k: 并不真正关机，只是发送警告信号给每位登录者。

-m: 将系统改为单用户模式。

-t: 在改变到其他运行级别之前，告诉 `init` 程序多久以后关机。

(4) 使用说明

`shutdown` 命令可以安全地将系统关闭。有些用户会使用直接断掉电源的方式来关闭 Linux 系统，这是十分危险的。因为 Linux 系统与 Windows 系统不同，其后台运行着许多进程，强制关机可能会导致进程的数据丢失，使系统处于不稳定的状态，甚至在有的系统中会损坏硬件设备（硬盘）。在系统关机前使用 `shutdown` 命令，系统管理员会通知所有登录的用户系统将要关闭，并且 `login` 指令会被冻结，即新的用户不能再登录。

3. halt

(1) 作用

`halt` 命令的作用是关闭系统，它的使用权限是超级用户。



(2) 格式

```
halt [-n] [-w] [-d] [-f] [-i] [-p]
```

(3) 主要参数

-n: 防止 sync 系统调用, 在使用 fsck (file system check) 命令修补根分区之后, 以阻止内核用老版本的超级块覆盖修补过的超级块。

-w: 并不是真正的重启或关机, 只是写 wtmp (/var/log/wtmp) 记录。

-d: 关闭系统, 但不留下记录。

-f: 没有调用 shutdown, 而强制关机或重启。

-i: 关机 (或重启) 前, 关掉所有的网络接口。

-p: 当关机的时候顺便做关闭电源的动作。

(4) 使用说明

Halt 命令就是调用 shutdown -h。halt 命令执行时, 杀死应用进程, 执行 sync (将存于 buffer 中的资料强制写入硬盘中) 系统调用, 文件系统写操作完成后就会停止内核。若系统的运行级别为 0 或 6, 则关闭系统; 否则以 shutdown 指令 (加上 -h 参数) 来取代。

4. reboot

(1) 作用

reboot 命令的作用是重新启动计算机, 它的使用权限是系统管理者。

(2) 格式

```
reboot [-n] [-w] [-d] [-i]
```

(3) 主要参数

-n: 在重开机前不做将记忆体资料写回硬盘的动作。

-w: 并不会真的重开机, 只是把记录写到 /var/log/wtmp 文件里。

-d: 不把记录写到 /var/log/wtmp 文件里 (-n 这个参数包含了 -d)。

-i: 在重开机之前先把所有与网络相关的装置停止。

5. exit

(1) 作用

exit 命令的作用是退出系统, 它的使用权限是所有用户。

(2) 格式

```
exit
```

(3) 参数

exit 命令没有参数, 运行后退出系统进入登录界面。

6. last

(1) 作用

last 命令的作用是显示近期用户或终端的登录情况, 它的使用权限是所有用户。通过 last 命令查看该程序的 log, 管理员可以获知谁曾经或企图连接系统。

(2) 格式

```
last[-n][-f file][-t tty] [-h 节点] [-i -IP] [-y] [-x]
```

(3) 主要参数

- n: 指定输出记录的条数。
- f file: 指定用文件 file 作为查询用的 log 文件。
- t tty: 只显示指定的虚拟控制台上的登录情况。
- h 节点: 只显示指定的节点上的登录情况。
- i IP: 只显示指定的 IP 上登录的情况。
- y: 显示记录的年、月、日。
- x: 显示系统关闭、用户登录和退出的历史。

7. file

(1) 作用

内容判断文件类型，使用权限是所有用户。

(2) 格式

```
file [options] 文件名
```

(3) 主要参数

- v: 在标准输出后显示版本信息，并且退出。
- z: 探测压缩过的文件类型。
- L: 允许符合连接。
- f name: 从文件 namefile 中读取要分析的文件名列表。

(4) 使用说明

使用 file 命令可以知道某个文件究竟是二进制（ELF 格式）的可执行文件，还是 ShellScript 文件，或者是其他的什么格式。file 能识别的文件类型有目录、Shell 脚本、英文文本、二进制可执行文件、C 语言源文件、文本文件、DOS 的可执行文件。

8. mkdir

(1) 作用

mkdir 命令的作用是建立名称为 dirname 的子目录，与 MS DOS 下的 md 命令类似，它的使用权限是所有用户。

(2) 格式

```
mkdir [options] 目录名
```

(3) 主要参数

- m, --mode=模式: 设定权限<模式>，与 chmod 命令类似。
- p, --parents: 需要时创建上层目录；如果目录早已存在，则不当作错误。
- v, --verbose: 每次创建新目录都显示信息。
- version: 显示版本信息后离开。

(4) 使用说明

在进行目录创建时可以设置目录的权限，此时使用的参数是“-m”。假设要创建的目录名是“tsk”，让所有用户都有 rwx 权限（即读、写、执行的权限），那么可以使用以下命令：

```
$ mkdir -m 777 tsk
```



9. grep

(1) 作用

grep 命令可以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。grep 全称是 Global Regular Expression Print，表示全局正则表达式版本，它的使用权限是所有用户。

(2) 格式

```
grep [options]
```

(3) 主要参数

- 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。

10. dd

(1) 作用

dd 命令用来复制文件，并根据参数将数据转换和格式化。

(2) 格式

```
dd [options]
```

(3) 主要参数

- bs=字节：同时设置读/写缓冲区的字节数（等于设置 ibs 和 obs）。
- cbs=字节：每次转换指定的<字节>。
- conv=关键字：根据以逗号分隔的关键字表示的方式来转换文件。
- count=块数目：只复制指定<块数目>的输入数据。
- ibs=字节：每次从缓冲区读取指定的<字节>。
- if=文件：读取<文件>内容，而非标准输入的数据。
- obs=字节：每次写入指定的<字节>到缓冲区。
- of=文件：将数据写入<文件>中而不在标准输出显示。
- seek=块数目：先略过以 obs 为单位的指定<块数目>的输出数据。
- skip=块数目：先略过以 ibs 为单位的指定<块数目>的输入数据。

(4) 使用说明

dd 命令常常用来制作 Linux 启动盘。先找一个可引导内核，使它的根设备指向正确的根分区，然后使用 dd 命令将其写入软盘：

```
$ rdev vmlinuz /dev/hda  
$ dd if=vmlinuz of=/dev/fd0
```

上面代码是使用 rdev 命令将可引导内核 vmlinuz 中的根设备指向/dev/hda，用户使用时把“hda”换成自己的根分区，接下来用 dd 命令将该内核写入软盘。

11. find

(1) 作用

`find` 命令的作用是在目录中搜索文件，它的使用权限是所有用户。

(2) 格式

```
find [path][options][expression]
```

`path` 指定目录路径，系统从此处开始沿着目录树向下查找文件。它是一个路径列表，相互用空格分离，如果不写 `path`，那么默认为当前目录。

(3) 主要参数

[options]参数：

-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。

-maxdepth levels: 表示至多查找到开始目录的第 level 层子目录。level 是一个非负数，如果 level 为 0 表示仅在当前目录中查找。

-mindepth levels: 表示至少查找到开始目录的第 level 层子目录。

-mount: 不在其他文件系统（如 Msdos、Vfat 等）的目录和文件中查找。

-version: 打印版本。

[expression]是匹配表达式，是 `find` 命令接受的表达式，`find` 命令的所有操作都是针对表达式的。它的参数非常多，这里只介绍一些常用的参数：

-name: 支持通配符*和?。

-atime n: 搜索在过去 n 天读取过的文件。

-ctime n: 搜索在过去 n 天修改过的文件。

-group groupname: 搜索组名为 groupname 的文件。

-user 用户名: 搜索文件属主为用户名（ID 或名称）的所有文件。

-size n: 搜索文件大小是 n 个 BLOCK 的文件。

-print: 输出搜索结果，并且打印。

(4) 使用说明

`find` 命令查找文件的几种方法：

① 根据文件名查找

例如，我们想要查找一个文件名是 `lilo.conf` 的文件，可以使用如下命令：

```
find / -name lilo.conf
```

`find` 命令后的“/”表示搜索整个硬盘。

② 快速查找文件

根据文件名查找文件会遇到一个实际问题，就是要花费相当长的一段时间，特别是大型 Linux 文件系统和大容量硬盘文件放在很深的子目录中时。如果我们知道了这个文件存放在某个目录中，那么只要在这个目录中往下寻找就能节省很多时间。比如 `smb.conf` 文件，从它的文件后缀“`.conf`”可以判断这是一个配置文件，那么它应该在 `/etc` 目录内，此时可以使用下面命令：

```
find /etc -name smb.conf
```

这样，使用“快速查找文件”方式可以缩短时间。



③ 根据部分文件名查找方法

有时我们知道某个文件包含有 `abvd` 这 4 个字,那么可以输入下面命令来查找系统中所有包含这 4 个字符的文件:

```
find / -name '*abvd*'
```

输入这个命令以后, Linux 系统会在 / 目录中查找所有的包含有 `abvd` 这 4 个字符的文件(其中 * 是通配符), 比如 `abvdrmz` 等符合条件的文件都能显示出来。

④ 使用混合查找方式查找文件

`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` 把两个查找参数链接起来组合成一个混合的查找方式。

```
find /etc -size +500000c -and -mtime +1
```

12. mv

(1) 作用

`mv` 命令用来为文件或目录改名, 或者将文件由一个目录移入另一个目录中, 它的使用权限是所有用户。该命令如同 DOS 命令中的 `ren` 和 `move` 的组合。

(2) 格式

```
mv [options] 源文件或目录 目标文件或目录
```

(3) 主要参数

`-i`: 交互方式操作。如果 `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, 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。`mv` 操作要覆盖某个已有的目标文件时不给任何指示, 指定此参数后 `i` 参数将不再起作用。

13. ls

(1) 作用

`ls` 命令用于显示目录内容, 类似 DOS 下的 `dir` 命令, 它的使用权限是所有用户。

(2) 格式

```
ls [options][filename]
```

(3) 主要参数

`-a, --all`: 不隐藏任何以 “.” 字符开始的项目。

`-A, --almost-all`: 列出除了 “.” 及 “..” 以外的任何项目。

`--author`: 印出每个文件著作者。

`-b, --escape`: 以八进制溢出序列表示不可打印的字符。

`--block-size=大小`: 指定 `size` 个字节作为块的单位。

`-i, --inode`: 列出每个文件的 `inode` 号。

`-l, --ignore=样式`: 不打印出任何符合 Shell 万用字符 <样式> 的项目。

`-k`: 即 `block-size=1K`。

`-l`: 使用较长格式列出信息。

-L, --dereference: 当显示符号链接的文件信息时, 显示符号链接所指示的对象, 而非符号链接本身的信息。

-m: 所有项目以逗号分隔, 并填满整行。

-n, --numeric-uid-gid: 类似-l, 但列出 UID 及 GID 号。

-N, --literal: 列出未经处理的项目名称, 例如不特别处理控制字符。

-p, --file-type: 加上文件类型的指示符号。

-Q, --quote-name: 用双引号标明项目名称。

-r, --reverse: 依相反次序排列。

-R, --recursive: 同时列出所有子目录层。

-s, --size: 以块大小为序。

(4) 使用说明

ls 命令是 Linux 系统使用频率最多的命令, 它的参数也是 Linux 命令中最多的。使用 ls 命令时会有几种不同的颜色, 其中蓝色表示是目录, 绿色表示是可执行文件, 红色表示是压缩文件, 浅蓝色表示是链接文件, 加粗的黑色表示符号链接, 灰色表示是其他格式文件。

14. diff

(1) 作用

diff 命令用于两个文件之间的比较, 并指出两者的不同, 它的使用权限是所有用户。

(2) 格式

```
diff [options] 源文件目标文件
```

(3) 主要参数

-a: 将所有文件当作文本文件来处理。

-b: 忽略空格造成的不同。

-B: 忽略空行造成的不同。

-c: 使用纲要输出格式。

-H: 利用试探法加速对大文件的搜索。

-I: 忽略大小写的变化。

-n --rcs: 输出 RCS 格式。

15. cmp

(1) 作用

cmp (“compare”的缩写) 命令用来简要指出两个文件是否存在差异, 它的使用权限是所有用户。

(2) 格式

```
cmp[options] 文件名
```

(3) 主要参数

-l: 将字节以十进制的方式输出, 并方便将两个文件中不同的以八进制的方式输出。

16. cat

(1) 作用

cat (“concatenate”的缩写) 命令用于连接并显示指定的一个和多个文件的有关信息,



它的使用权限是所有用户。

(2) 格式

```
cat [options] 文件1 文件2.....
```

(3) 主要参数

- 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。
- s: 当遇到有连续两行以上的空白行时, 就代换为一行的空白行。

(4) 使用说明

cat 命令一个最简单的用处是显示文本文件的内容。例如, 我们想在命令行看一下 README 文件的内容, 可以使用命令:

```
$ cat README
```

有时需要将几个文件处理成一个文件, 并将这种处理的结果保存到一个单独的输出文件。cat 命令在其输入上接受一个或多个文件, 并将它们作为一个单独的文件打印到它的输出。例如, 把 README 和 INSTALL 的文件内容加上行号(空白行不加)之后, 将内容附加到一个新文本文件 File1 中:

```
$ cat README INSTALL File1
```

cat 还有一个重要的功能就是可以对行进行编号, 这种功能对于程序文档的编制以及法律和科学文档的编制很方便, 打印在左边的行号使得参考文档的某一部分变得容易, 这些在编程、科学研究、业务报告甚至是立法工作中都是非常重要的。

17. ln

(1) 作用

ln 命令用来在文件之间创建链接, 它的使用权限是所有用户。

(2) 格式

```
ln [options] 源文件 [链接名]
```

(3) 主要参数

- f: 链接时先将源文件删除。
- d: 允许系统管理者硬链接自己的目录。
- s: 进行软链接 (Symbolic Link)。
- b: 将在链接时会被覆盖或删除的文件进行备份。

18. df

(1) 作用

df 命令用来检查文件系统的磁盘空间占用情况, 使用权限是所有用户。

(2) 格式

```
df [options]
```

(3) 主要参数

- s: 对每个 Names 参数只给出占用的数据块总数。

-a: 递归地显示指定目录中各文件及子目录中各文件占用的数据块数。若既不指定-s, 也不指定-a, 则只显示 Names 中的每一个目录及其中的各子目录所占的磁盘块数。

-k: 以 1024 字节为单位列出磁盘空间使用情况。

-x: 跳过在不同文件系统上的目录不予统计。

-l: 计算所有的文件大小, 对硬链接文件则计算多次。

-i: 显示 inode 信息而非块使用量。

-h: 以容易理解的格式印出文件系统大小, 例如 136KB、254MB、21GB。

-P: 使用 POSIX 输出格式。

-T: 显示文件系统类型。

(4) 使用说明

df 命令被广泛地用来生成文件系统的使用统计数据, 它能显示系统中所有的文件系统的信息, 包括总容量、可用的空闲空间、目前的安装点等。超级权限用户使用 df 命令时会发现这样的情况: 某个分区的容量超过了 100%。这是因为 Linux 系统为超级用户保留了 10% 的空间, 由其单独支配。也就是说, 对于超级用户而言, 他所见到的硬盘容量将是 110%。这样的安排对于系统管理而言是有好处的, 当硬盘被使用的容量接近 100% 时系统管理员还可以正常工作。

19. top

(1) 作用

top 命令用来显示执行中的程序进程, 使用权限是所有用户。

(2) 格式

```
top [-] [d delay] [q] [c] [S] [s] [i] [n]
```

(3) 主要参数

d: 指定更新的间隔, 以秒计算。

q: 没有任何延迟的更新。如果使用者有超级用户, 则 top 命令将会以最高的优先序执行。

c: 显示进程完整的路径与名称。

S: 累积模式, 会将已完成或消失的子行程的 CPU 时间累积起来。

s: 安全模式。

i: 不显示任何闲置 (Idle) 或无用 (Zombie) 的行程。

n: 显示更新的次数, 完成后将会退出 top。

20. free

(1) 作用

free 命令用来显示内存的使用情况, 使用权限是所有用户。

(2) 格式

```
free [-b|-k|-m] [-o] [-s delay] [-t] [-V]
```

(3) 主要参数

-b -k -m: 分别以字节 (KB、MB) 为单位显示内存使用情况。

-s delay: 显示每隔多少秒数来显示一次内存使用情况。

-t: 显示内存总和列。



-o: 不显示缓冲区调节列。

(4) 使用说明

`free` 命令是用来查看内存使用情况的主要命令。和 `top` 命令相比，它的优点是使用简单，并且只占用很少的系统资源。通过 `-S` 参数可以使用 `free` 命令不间断地监视有多少内存在使用，这样可以把它当作一个方便的实时监控器。

```
#free -b -s5
```

使用这个命令后终端会连续不断地报告内存使用情况（以字节为单位），每 5 秒更新一次。

21. quota

(1) 作用

`quota` 命令用来显示磁盘使用情况和限制情况，使用权限为超级用户。

(2) 格式

```
quota [-g][-u][-v][-p] 用户名 组名
```

(3) 参数

-g: 显示用户所在组的磁盘使用限制。

-u: 显示用户的磁盘使用限制。

-v: 显示没有分配空间的文件系统的分配情况。

-p: 显示简化信息。

(4) 使用说明

在企业应用中磁盘配额非常重要，普通用户要学会看懂自己的磁盘使用情况。要查询磁盘配额可以使用下面命令（下例中用户账号是 `caojh`）：

```
#quota caojh
Disk quotas for user caojh(uid 502):
Filesystem blocks quota limit grace files quota limit grace
/dev/hda3 58 200000 400000 41 500 1000
```

以上显示 ID 号为 502 的 `caojh` 账号，文件个数设置为 500~1000 个，硬盘空间限制设置为 200MB~400MB。一旦磁盘配额要用完时，就需要删除一些垃圾文件或向系统管理员请求追加配额。

22. at

(1) 作用

`at` 命令用来在指定时刻执行指定的命令序列。

(2) 格式

```
at [-V] [-q] [-f file] [-m] time
```

(3) 主要参数

-V: 显示标准错误输出。

-q: 许多队列输出。

-f: 从文件中读取作业。

-m: 执行完作业后发送电子邮件到用户。

time: 设定作业执行的时间。**time** 格式有严格的要求, 由小时、分钟、日期和时间的偏移量组成, 其中日期的格式为 **MM.DD.YY**, **MM** 是分钟, **DD** 是日期, **YY** 是指年份。偏移量的格式为时间+偏移量, 单位是 **minutes**、**hours** 和 **days**。

(4) 使用说明

```
#at -f data 15:30 +2 days
```

该命令表示让系统在两天后的 17:30 执行文件 **data** 中指定的作业。

23. lp

(1) 作用

lp 是打印文件的命令, 使用权限是所有用户。

(2) 格式

```
lp [-c] [-d] [-m] [-number] [-title] [-p]
```

(3) 主要参数

- c: 先拷贝文件再打印。
- d: 打印队列文件。
- m: 打印结束后发送电子邮件到用户。
- number: 打印份数。
- title: 打印标题。
- p: 设定打印的优先级别, 最高为 100。

24. useradd

(1) 作用

useradd 命令用来建立用户账号和创建用户的起始目录, 使用权限是超级用户。

(2) 格式

```
useradd [options] name
```

(3) 主要参数

- c: 加上备注文字, 备注文字保存在 **passwd** 的备注栏中。
- d: 指定用户登录时的起始目录。
- D: 变更预设值。
- e: 指定账号的有效期限, 缺省表示永久有效。
- f: 指定在密码过期后多少天即关闭该账号。
- g: 指定用户所属的群组。
- G: 指定用户所属的附加群组。
- m: 自动建立用户的登录目录。
- M: 不要自动建立用户的登录目录。
- n: 取消建立以用户名称为名的群组。
- r: 建立系统账号。
- s: 指定用户登录后所使用的 **shell**。
- u: 指定用户 ID 号。



(4) 使用说明

`useradd` 可用来建立用户账号, 它和 `adduser` 命令是相同的。账号建好之后, 再用 `passwd` 设定账号的密码。使用 `useradd` 命令所建立的账号, 实际上是保存在 `/etc/passwd` 文本文件中。例如, 建立一个新用户账户, 并设置 ID:

```
#useradd caojh -u 544
```

需要说明的是, 设定 ID 值时尽量要大于 500, 以免冲突。因为 Linux 安装后会建立一些特殊用户, 一般 0~499 之间的值留给 `bin`、`mail` 这样的系统账号。

25. groupadd

(1) 作用

`groupadd` 命令用于将新组加入系统。

(2) 格式

```
groupadd [-g gid] [-o] [-r] [-f] groupname
```

(3) 主要参数

`-g gid`: 指定组 ID 号。

`-o`: 允许组 ID 号, 不必唯一。

`-r`: 加入组 ID 号, 低于 499 系统账号。

`-f`: 加入已经有的组时, 发展程序退出。

(4) 使用说明

建立一个新组, 并设置组 ID 加入系统:

```
#groupadd -g 344 cjh
```

此时在 `/etc/passwd` 文件中产生一个组 ID (GID) 是 344 的项目。

26. kill

(1) 作用

`kill` 命令用来中止一个进程。

(2) 格式

```
kill [ -s signal | p ] pid ...  
kill -l [ signal ]
```

(3) 主要参数

`-s`: 指定发送的信号。

`-p`: 模拟发送信号。

`-l`: 指定信号的名称列表。

`pid`: 要中止进程的 ID 号。

`Signal`: 表示信号。

(4) 使用说明

进程是 Linux 系统中一个非常重要的概念, Linux 是一个多任务的操作系统, 系统经常同时运行着多个进程。用户不关心这些进程究竟是如何分配的, 或者是内核如何管理分配时间片的, 关心的是如何去控制这些进程, 让它们能够很好地为用户服务。`kill` 命令的工作原理是向 Linux 系统的内核发送一个系统操作信号和某个程序的进程标识号, 然后系

统内核就可以对进程标识号指定的进程进行操作。比如在 `top` 命令中，我们看到系统运行许多进程，有时就需要使用 `kill` 命令中止某些进程来提高系统资源。

27. ifconfig

(1) 作用

`ifconfig` 用于查看和更改网络接口的地址和参数，包括 IP 地址、网络掩码、广播地址，使用权限是超级用户。

(2) 格式

```
ifconfig -interface [options] address
```

(3) 主要参数

`-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。

`pointtopoint`: 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) 使用说明

`ifconfig` 是用来设置和配置网卡的命令行工具。为了手工配置网络，这是一个必须掌握的命令，使用该命令的好处是无须重新启动机器。

28. ping

(1) 作用

`ping` 检测主机网络接口状态，使用权限是所有用户。

(2) 格式

```
ping [options] IP 地址
```

(3) 主要参数

`-d`: 使用 `Socket` 的 `SO_DEBUG` 功能。

`-c`: 设置完成要求回应的次数。

`-f`: 极限检测。

`-i`: 指定收发信息的间隔秒数。

`-I`: 网络界面使用指定的网络界面送出数据包。

`-l`: 前置载入，设置在送出要求信息之前，先行发出的数据包。

`-n`: 只输出数值。

`-p`: 设置填满数据包的范本样式。

`-q`: 不显示指令执行过程，开头和结尾的相关信息除外。

`-r`: 忽略普通的 `Routing Table`，直接将数据包送到远端主机上。

`-R`: 记录路由过程。

`-s`: 设置数据包的大小。

`-t`: 设置存活数值 `TTL` 的大小。

`-v`: 详细显示指令的执行过程。



(4) 使用说明

ping 命令是使用最多的网络指令,通常使用它检测网络是否连通,它使用 ICMP 协议。但是有时会有这样的情况,用户可以浏览查看一个网页,但是使用 ping 命令却无法连通,这是因为一些网站出于安全考虑安装了防火墙。另外,也可以通过下面的命令使系统对 ping 命令没有反应:

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

29. finger

(1) 作用

finger 用来查询一台主机上的登录账号的信息,通常会显示用户名、主目录、停滞时间、登录时间、登录 Shell 等信息,使用权限为所有用户。

(2) 格式

```
finger [选项] [使用者] [用户@主机]
```

(3) 主要参数

-s: 显示用户注册名、实际姓名、终端名称、写状态、停滞时间、登录时间等信息。

-l: 除了-s 选项显示的信息外,还显示用户主目录、登录 Shell、邮件状态等信息,以及用户主目录下的.plan、.project 和.forward 文件的内容。

-p: 除了不显示.plan 文件和.project 文件以外,与-l 选项相同。

(4) 使用说明

如果要查询远程机上的用户信息,需要在用户名后面接“@主机名”,采用[用户名@主机名]的格式,不过要查询的网络主机需要运行 finger 守护进程的支持。

30. nslookup

(1) 作用

nslookup 命令的功能是查询一台机器的 IP 地址和其对应的域名,使用权限是所有用户。它通常需要一台域名服务器来提供域名服务。如果用户已经设置好域名服务器,就可以用这个命令查看不同主机的 IP 地址对应的域名。

(2) 格式

```
nslookup [IP 地址/域名]
```

(3) 使用说明

在本地计算机上使用 nslookup 命令:

```
$ nslookup
Default Server: name.cao.com.cn
Address: 192.168.1.9
>
```

在符号“>”后面输入要查询的 IP 地址域名,按回车键即可。如果要退出该命令,输入“exit”,并按回车键即可。

31. passwd

(1) 作用

passwd 命令原来修改账户的登录密码,使用权限是所有用户。

(2) 格式

```
passwd [选项] 账户名称
```

(3) 主要参数

-l: 锁定已经命名的账户名称，只有具备超级用户权限的使用者方可使用。

-u: 解开账户锁定状态，只有具备超级用户权限的使用者方可使用。

-x, --maximum=DAYS: 最大密码使用时间（天），只有具备超级用户权限的使用者方可使用。

-n, --minimum=DAYS: 最小密码使用时间（天），只有具备超级用户权限的使用者方可使用。

-d: 删除使用者的密码，只有具备超级用户权限的使用者方可使用。

-S: 检查指定使用者的密码认证种类，只有具备超级用户权限的使用者方可使用。

32. su

(1) 作用

su 的作用是变更为其他使用者的身份，超级用户除外，需要键入该使用者的密码。

(2) 格式

```
su [选项]... [-] [USER [ARG]...]
```

(3) 主要参数

-f, --fast: 不必读启动文件（如 `csh`、`cshrc` 等），仅用于 `csh` 或 `tcsh` 两种 Shell。

-l, --login: 加了这个参数之后，就好像是重新登录为该使用者一样，大部分环境变量（例如 `HOME`、`SHELL` 和 `USER` 等）都是以该使用者（`USER`）为主，并且工作目录也会改变。如果没有指定 `USER`，缺省情况下是 `root`。

-m, -p, --preserve-environment: 执行 `su` 时不改变环境变数。

-c command: 变更账号为 `USER` 的使用者，并执行指令（`command`）后再变回原来的使用者。

USER: 打算变更的使用者账号，`ARG` 传入新的 Shell 参数。

33. umask

(1) 作用

`Umask` 为用户文件和目录的文件创建缺省屏蔽值，若将此命令放入 `profile` 文件，就可控制该用户后续所建文件的存取许可。它告诉系统在创建文件时不给谁存取许可。使用权限是所有用户。

(2) 格式

```
umask [-p] [-S] [mode]
```

(3) 主要参数

-S: 确定当前的 `umask` 设置。

-p: 修改 `umask` 设置。

[mode]: 修改数值。

34. chmod

(1) 作用

`chmod` 命令非常重要，用于改变文件或目录的访问权限，用户可以用它控制文件或目



录的访问权限，使用权限是超级用户。

(2) 格式

`chmod` 命令有两种用法：一种是包含字母和操作符表达式的字符设定法（相对权限设定）；另一种是包含数字的数字设定法（绝对权限设定）。

① 字符设定法

```
chmod [who] [+ | - | =] [mode] 文件名
```

操作对象 `who` 可以是下述字母中的任一个或它们的组合：

- `u` 表示用户，即文件或目录的所有者。
- `g` 表示同组用户，即与文件属主有相同组 ID 的所有用户。
- `o` 表示其他用户。
- `a` 表示所有用户，它是系统默认值。

操作符号：

- `+` 添加某个权限。
- `-` 取消某个权限。
- `=` 赋予给定权限，并取消其他所有权限。

设置 `mode` 的权限可用下述字母的任意组合：

- `r` 可读。
- `w` 可写。
- `x` 可执行。
- `X` 只有目标文件对某些用户是可执行的或该目标文件是目录时才追加 `x` 属性。
- `s` 文件执行时把进程的属主或组 ID 设置为该文件的文件属主，“`u+s`”设置文件的用户 ID 位，“`g+s`”设置组 ID 位。
- `t` 保存程序的文本到交换设备上。
- `u` 与文件属主拥有一样的权限。
- `g` 与和文件属主同组的用户拥有一样的权限。
- `o` 与其他用户拥有一样的权限。

文件名：以空格分开的要改变权限的文件列表，支持通配符。一个命令行中可以给出多个权限方式，其间用逗号隔开。

② 数字设定法

```
chmod [mode] 文件名
```

数字属性的格式应为 3 个 0~7 的八进制数，其顺序是(u)(g)(o)文件名，以空格分开要改变权限的文件列表，支持通配符。数字表示的权限的含义如下：0001 为所有者的执行权限；0002 为所有者的写权限；0000 为所有者的读权限；0010 为组的执行权限；0020 为组的写权限；0040 为组的读权限；0100 为其他人的执行权限；0200 为其他人的写权限；0400 为其他人的读权限；1000 为粘贴位置位；2000 表示假如这个文件是可执行文件，则组 ID 为位置位，否则其中文件锁定位位置位；4000 表示假如这个文件是可执行文件，则用户 ID 为位置位。

35. chown

(1) 作用

更改一个或多个文件或目录的属主和属组。使用权限是超级用户。

(2) 格式

```
chown [选项] 用户或组文件
```

(3) 主要参数

--dereference: 受影响的是符号链接所指示的对象，而非符号链接本身。

-h, --no-dereference: 会影响符号链接本身，而非符号链接所指示的目的地（当系统支持更改符号链接的所有者，此选项才有效）。

--from=目前所有者：目前组只当每个文件的所有者和组符合选项所指定的，才会更改所有者和组。其中一个可以省略，这已省略的属性就不需要符合原有的属性。

-f, --silent, --quiet: 去除大部分的错误信息。

-R, --recursive: 递归处理所有的文件及子目录。

-v, --verbose: 处理任何文件都会显示信息。

(4) 使用说明

chown 将指定文件的拥有者改为指定的用户或组，用户可以是用户名或用户 ID；组可以是组名或组 ID；文件是以空格分开的要改变权限的文件列表，支持通配符。系统管理员经常使用 **chown** 命令，在将文件拷贝到另一个用户的目录下以后，让用户拥有使用该文件的权限。

36. sudo

(1) 作用

sudo 是一种以限制配置文件中的命令为基础，在有限时间内给用户使用，并且记录到日志中的命令，权限是所有用户。

(2) 格式

```
sudo [options]
```

(3) 主要参数

-b: 在后台执行命令。

-h: 显示帮助。

-H: 将 HOME 环境变量设为新身份的 HOME 环境变量。

-k: 结束密码的有效期，即下次将需要输入密码。

-l: 列出当前用户可以使用的命令。

-p: 改变询问密码的提示符号。

-s <shell>: 执行指定的 Shell。

-u <用户>: 以指定的用户为新身份，不使用时默认为 root。

-v: 延长密码有效期 5 分钟。

(4) 使用说明

sudo 命令的配置在 `/etc/sudoers` 文件中。当用户使用 **sudo** 命令时，需要输入口令以验证使用者身份。随后的一段时间内可以使用定义好的命令，当使用配置文件中没有的命令时，将会有报警的记录。**sudo** 是系统管理员用来允许某些用户以 root 身份运行部分/全部系统命令的程序。一个通用的用途是增强了站点的安全性，如果需要每天以超级用户的身



份做一些日常工作，经常执行一些固定的只有超级用户身份才能执行的命令，那么用 `sudo` 是非常适合的。

37. ps

(1) 作用

`ps` 显示瞬间进程（process）的动态，使用权限是所有使用者。

(2) 格式

```
ps [options] [--help]
```

(3) 主要参数

`ps` 的参数非常多，此处仅列出几个常用的参数。

`-A`: 列出所有的进程。

`-l`: 显示长列表。

`-m`: 显示内存信息。

`-w`: 显示加宽可以显示较多的信息。

`-e`: 显示所有进程。

`a`: 显示终端上的所有进程，包括其他用户的进程。

`-au`: 显示较详细的信息。

`-aux`: 显示所有包含其他使用者的进程。

(4) 使用说明

要对进程进行监测和控制，首先要了解当前进程的情况，也就是需要查看当前进程。

`ps` 命令就是最基本、也是非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行、运行的状态、进程是否结束、进程有没有僵尸、哪些进程占用了过多的资源等，大部分信息都可以通过执行该命令得到。

38. who

(1) 作用

`who` 显示系统中有哪些用户登录了系统，显示的资料包含使用者 ID、使用的登录终端、上线时间、空闲时间、CPU 占用，以及做了些什么。使用权限为所有用户。

(2) 格式

```
who [options] [user]
```

(3) 主要参数

`-h`: 不要显示标题列。

`-u`: 不要显示使用者的动作/工作。

`-s`: 使用简短的格式来显示。

`-f`: 不要显示使用者的上线位置。

`-V`: 显示程序版本。

(4) 使用说明

该命令主要用于查看当前在线上的用户情况，系统管理员希望监视每个登录的用户此时此刻的所作所为，也要使用 `who` 命令。`who` 命令应用起来非常简单，可以比较准确地掌握用户的情况，所以使用非常广泛。

39. tar

(1) 作用

tar 命令是 Linux 系统中备份文件的可靠方法，几乎可以工作于任何环境中，它的使用权限是所有用户。

(2) 格式

```
tar [主选项+可选项] 文件或目录
```

(3) 主要参数

使用该命令时，主选项是必须要有的，可选项是辅助使用的，可以选用。

主选项：

- **-c** 创建新的档案文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。
- **-r** 把要存档的文件追加到档案文件的末尾。例如用户已经做好备份文件，又发现还有一个目录或是一些文件忘记备份了，这时可以使用该选项，将目录或文件追加到备份文件中。
- **-t** 列出档案文件的内容，查看已经备份了哪些文件。
- **-u** 更新文件。即用新增的文件取代原备份文件，如果在备份文件中找不到要更新的文件，则把它追加到备份文件的最后。
- **-x** 从档案文件中释放文件。

可选项：

- **-b** 该选项是为磁带机设定的，其后跟一数字，用来说明区块的大小，系统预设值为 20 (20 × 512 bytes)。
- **-f** 使用档案文件或设备，这个选项通常是必选的。
- **-k** 保存已经存在的文件。例如把某个文件还原，在还原的过程中遇到相同的文件，不会进行覆盖。
- **-m** 在还原文件时，把所有文件的修改时间设定为现在。
- **-M** 创建多卷的档案文件，以便在几个磁盘中存放。
- **-v** 详细报告 tar 处理的文件信息。如无此选项，tar 不报告文件信息。
- **-w** 每一步都要求确认。
- **-z** 用 gzip 来压缩/解压缩文件，加上该选项后可以将档案文件进行压缩，还原时也一定要使用该选项进行解压缩。

40. unzip

(1) 作用

unzip 命令位于 /usr/bin 目录中，和 MS DOS 下的 pkzip、pkunzip 及 MS Windows 中的 winzip 软件功能一样，将文件压缩成 .zip 文件，以节省硬盘空间，当需要的时候再将压缩文件用 unzip 命令解开。该命令使用权限是所有用户。

(2) 格式

```
unzip [-cflptuvz] [-agCjLMnoqsVX] [-P <密码>] [.zip 文件] [文件] [-d <目录>] [-x <文件>]
```

(3) 主要参数

-c：将解压缩的结果显示到屏幕上，并对字符做适当的转换。



- f: 更新现有的文件。
 - l: 显示压缩文件内所包含的文件。
 - p: 与-c 参数类似, 会将解压缩的结果显示到屏幕上, 但不会执行任何的转换。
 - t: 检查压缩文件是否正确。
 - u: 与-f 参数类似, 但除了更新现有的文件外, 也会将压缩文件中的其他文件解压缩到目录中。
 - v: 执行时显示详细的信息。
 - z: 仅显示压缩文件的备注文字。
 - a: 对文本文件进行必要的字符转换。
 - b: 不对文本文件进行字符转换。
 - C: 压缩文件中的文件名称区分大小写。
 - j: 不处理压缩文件中原有的目录路径。
 - L: 将压缩文件中的全部文件名改为小写。
 - M: 将输出结果送到 more 程序处理。
 - n: 解压缩时不覆盖原有的文件。
 - o: 不询问用户, unzip 命令执行后覆盖原有文件。
 - P<密码>: 使用 zip 的密码选项。
 - q: 执行时不显示任何信息。
 - s: 将文件名中的空白字符转换为底线字符。
 - V: 保留 VMS 的文件版本信息。
 - X: 解压缩的同时回存文件原来的 UID/GID。
- [.zip 文件]: 指定.zip 压缩文件。
[文件]: 指定要处理.zip 压缩文件中的哪些文件。
-d<目录>: 指定文件解压缩后所要存储的目录。
-x<文件>: 指定不要处理.zip 压缩文件中的哪些文件。
-Z unzip: -Z 等于执行 zipinfo 指令。在 Linux 中, 还提供了一个叫 zipinfo 的工具, 能够察看 zip 压缩文件的详细信息。

41. gunzip

(1) 作用

gunzip 命令的作用是解压文件, 使用权限是所有用户。

(2) 格式

```
gunzip [-acfhlLnNqrtvV][-s <压缩字尾字符串>][文件...] 或者 gunzip  
[-acfhlLnNqrtvV][-s <压缩字尾字符串>][目录]
```

(3) 主要参数

- a 或--ascii: 使用 ASCII 文字模式。
- c 或--stdout 或--to-stdout: 把解压后的文件输出到标准输出设备。
- f 或-force: 强行解开压缩文件, 不管文件名称或硬连接是否存在, 以及该文件是否为符号连接。
- h 或--help: 在线帮助。
- l 或--list: 列出压缩文件的相关信息。
- L 或--license: 显示版本与版权信息。

-n 或--no-name: 解压缩时, 若压缩文件内含有原来的文件名称及时间戳记, 则将其忽略不予处理。

-N 或--name: 解压缩时, 若压缩文件内含有原来的文件名称及时间戳记, 则将其回存到解开的文件上。

-q 或--quiet: 不显示警告信息。

-r 或--recursive: 递归处理, 将指定目录下的所有文件及子目录一并处理。

-S<压缩字尾字符串>或--suffix<压缩字尾字符串>: 更改压缩字尾字符串。

-t 或--test: 测试压缩文件是否正确无误。

-v 或--verbose: 显示指令执行过程。

-V 或--version: 显示版本信息。

(4) 使用说明

gunzip 是个使用广泛的解压缩程序, 它用于解开会用 gzip 命令压缩生成的文件, 这些压缩文件最后的扩展名为“.gz”。gunzip 命令是 gzip 命令的硬连接, 因此不论是压缩或解压缩, 都可通过 gzip 指令单独完成。

3.2 系统与硬件

操作系统必须与硬件系统密切协作, 它需要那些仅能由硬件提供的服务。为了全面理解 Linux 操作系统, 必须要懂得一些有关硬件的知识。本节将对硬件资源管理及外在设备的使用方法做一简要的介绍。

3.2.1 Linux 硬件资源管理

在 Linux 下查看系统的硬件信息没有在 Windows 系统下直观, 但 Linux 操作系统下的命令可以更加清楚地显示系统的硬件信息。在 Linux 中查看各硬件资源信息的命令如下。

1.

```
lspci | grep -i vga
```

2. 的

```
dmidecode | grep -i 'serial number'
```

3. CPU

```
#通过/proc文件系统
cat /proc/cpuinfo
dmesg | grep -i cpu
#通过查看开机信息
dmidecode -t processor
```

4.

```
fdisk -l 分区情况
df -h 大小情况
```



```
du -h 使用情况  
dmesg | grep sda
```

5.

```
cat /proc/meminfo  
dmesg | grep mem  
free -m  
vmstat  
dmldecode | grep -i mem
```

6.

```
dmesg | grep -i eth  
lspci | grep -i eth
```

7. USB

```
cat /proc/bus/input/devices 查看键盘和鼠标  
cat /proc/bus/usb/devices 查看 USB 设备  
cat /proc/interrupts
```

8.

```
dmesg | grep -i vga
```

9.

```
lspci (显示外设备信息, 如 USB、网卡等信息)
```

熟悉上述命令就可以方便地查看 Linux 系统的硬件资源信息了。

3.2.2 Linux 外在设备的使用方法

1. 与

在使用 Linux 系统时, 用户需要使用某些外部设备, 如软盘、U 盘、光驱和磁带等。在 Linux 中, 这些硬件设备都以文件的形式存在, 不同的设备文件有不同的文件类型。设备文件在 Linux 系统中存放在/dev 下面, 设备文件的命名方式是主设备号加次设备号, 主设备号说明设备类型, 次设备号说明具体指哪一个设备。

(1) 软盘

设备文件为/dev/fdx, fd (floppydisk) 是软盘驱动器, x 是软盘驱动器相应的编号。如:/dev/fd0 表示系统中的第一个软盘驱动器。

(2) U 盘

在 Linux 下将 U 盘识别为 SCSI 设备, 对应的设备文件为/dev/sdax。主设备号 sd 表示 SCSI disk, a 表示第一块 SCSI 设备, 若有第二块 SCSI 设备, 则对应的设备文件是/dev/sdb; x 表示 SCSI 设备的分区编号, 如:/dev/sda1 表示第一块 SCSI 设备的第一个分区, 而/dev/sdc5 表示第三块 SCSI 设备的第一个逻辑分区。

(3) 光驱

光驱是经常使用的外部设备之一。IDE光驱在Linux中对应的设备文件为/dev/had, SCSI光驱在Linux中对应的设备文件为/dev/srx。在/dev下还有一个cdrom设备文件,它是一个指向光驱的符号链接。

(4) 磁带

磁带是经常使用的外部存储设备。通常, Linux下SCSI磁带驱动器对应的设备文件为/dev/stx, st表示SCSI tape, x表示磁带驱动器的号码。如:/dev/st0表示系统第一个磁带驱动器的设备文件, /dev/st1则表示系统第二个磁带驱动器对应的设备文件, 以此类推。

2. 系统的

对于不同的外部设备, Linux提供了不同的文件类型, 常见的文件类型如表3-1所示。

3-1 Linux 系统

系统	
msdos	DOS 文件系统类型
vfat	支持长文件名的 DOS 分区文件系统类型
Iso9660	光盘格式文件系统类型
ext2/ext3	Linux 下的文件系统类型

了解了设备文件及其对应的文件系统类型后, 用户就可以在Linux下通过挂载来使用这些设备了。

3. 的

Linux操作系统下执行挂载功能的命令是mount, 格式如下:

```
mount -t 文件类型 设备名 挂载点
```

其中, 文件类型是上面讲到的几种文件系统格式, 设备名是对应的设备文件, 挂载点是Linux系统下指定的挂载目录。将设备挂载到这个指定目录后, 访问该挂载目录即相当于访问这个设备。

Linux系统中有一个/mnt目录, 专门用作挂载点。如果安装的系统中有软盘设备, 那么系统默认会在/mnt下创建一个目录/mnt/floppy, 用于软驱的挂载目录; 如果安装的系统中有光驱设备, 那么系统默认会在/mnt下创建一个目录/mnt/cdrom, 用于光盘的挂载目录。也就是说, 在实际应用中, 一般都将在设备挂载到/mnt目录的子目录中。

(1) 挂载软盘

命令格式为:

```
mount -t msdos /dev/fd0 /mnt/floppy
```

该命令可将采用DOS文件格式的一张软盘挂载到系统中, 以后就可以在/mnt/floppy目录下查看这张软盘中的内容了。

(2) 挂载U盘

命令格式为:

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



需要说明的是，挂载 U 盘设备前可在命令行输入“dmesg|more”来查看 U 盘的设备名，通常设备文件为/dev/sda1，建立挂载点 mkdir/mnt/usb，然后用上述 mount 命令进行挂载，然后就可以在/mnt/usb 目录下访问该 U 盘的内容了。

(3) 挂载光盘

命令格式为：

```
mount -t iso9660 /dev/hda /mnt/cdrom
```

或者

```
mount /dev/cdrom /mnt/cdrom
```

 用 mount 命令挂载的是软盘、光盘和 U 盘，而不是软驱和光驱，当需要换另外一张光盘的时候，必须先完成卸载操作，然后再完成装载新光盘的操作。

4. 的

命令格式为：

```
umount 挂载目录
```

例如，卸载软盘可输入如下命令：

```
umount /mnt/floppy
```

要卸载光盘则可以输入如下命令：

```
umount /mnt/cdrom
```

 在光盘没有卸载之前，光驱上的弹出键不起任何作用。

3.3 文件系统结构介绍

文件系统类型决定了向分区中存放、读取文件数据的方式和效率，在对分区进行格式化时需要选择所使用的文件系统类型。文件系统主要有以下几种类型。

1. 系统

包括硬盘、CD-ROM、DVD、USB 存储器与磁盘阵列等。常见文件系统的格式有：autofs、coda、Ext（Extended File system，扩展文件系统）、Ext2、Ext3、VFAT、ISO9660（通常是 CD-ROM）、UFS（Unix File System，Unix 文件系统）、FAT（File Allocation Table，文件分配表）、FAT16、FAT32、NTFS（New Technology File System）等。

2. 系统

网络文件系统是可以远程访问的文件系统，这种文件系统在服务器端仍是本地的磁盘文件系统，客户机通过网络来远程访问数据。常见文件系统格式有：NFS（Network File System，网络文件系统）、Samba（SMB/CIFS）、AFP（Apple Filling Protocol，Apple 文件归档协议）和 WebDAV 等。

3. / 系统

不驻留在磁盘上的文件系统。常见格式有：TMPFS（临时文件系统）、PROCFS（Process File System，进程文件系统）和 LOOPBACKFS（Loopback File System，回送文件系统）。

在 Windows 系统中，硬盘分区通常都是采用 FAT32 或 NTFS 文件系统，而在 Linux 系统中，硬盘分区大都采用 EXT3、EXT4 文件系统。最新的 RHEL6.4 采用的就是 EXT4 文件系统。EXT4 为第 4 代的扩展文件系统，是 RHEL6 中默认使用的文件系统类型，属于典型的日志型文件系统。其特点是可以保存磁盘存取记录的日志数据，便于需要的时候恢复，此外性能和稳定性更加出色。除了 EXT4 文件系统之外，Linux 中还有一个比较特殊的 swap 类型的文件系统，swap 文件系统是专门给交换分区使用的。交换分区类似于 Windows 系统中的虚拟内存，能够在一定程度上缓解物理内存不足的问题。不同的是，在 Windows 系统中采用的是一个名为 pagefile.sys 的系统文件作为虚拟内存来使用，而在 Linux 系统中则是划分了一个单独的分区作为虚拟内存来使用，这个分区被称为交换分区。交换分区的大小通常设置为主机物理内存容量的 2 倍，如主机的物理内存容量为 1GB，则交换分区的容量设置为 2GB 即可。

大多数由 Windows 平台转来的用户在使用 Linux 文件系统的时候都会感到困惑。Linux 文件系统与 Windows 文件系统有很大的差别。作为开端，只有一个单独的顶级目录结构，所有一切都从‘根’开始，用‘/’代表，并且延伸到子目录。DOS/Windows 有不同的分区，同时目录都存于分区上。Linux 则通过‘加载’的方式把所有分区都放置在“根”下指定的目录里。Windows 下最接近于‘根’的是“c:”。可以总结为在 Windows 环境下，目录结构属于分区；在 Linux 环境下，分区‘加载’于目录结构。

在 Windows 环境下，启动时检测不同的分区并赋予每个分区一个字母。在 Linux 环境下，除非用户加载一个分区或设备，否则系统不会知道哪个分区的存在。这看上去也许不是访问分区或设备的最方便的方式，但是这种方式提供了较大的机动性。这种构建的方式被称为统一的文件系统，超越了 Windows 系统所使用的方式。举个使用/usr 的例子，这个目录包含了大多数的系统可执行文件。在 Linux 文件系统下，用户可以选择把它加载为其他分区甚至网络上的其他计算机，而系统都不会感知其中的不同，因为它表现出的只是本地目录结构下众多目录中的一个而已。

Linux 文件系统目录说明及 Windows 文件系统的比较可总结如下：对于 Windows 系统，每个分区有一个盘符，每个分区使用独立的文件系统，在每一个分区都会有一个根目录，定位文件或目录位置时使用“\”进行分隔。而对于 Linux 系统，目录结构如图 3-1 所示，整个系统中只存在一个根目录，所有的分区、目录、文件都在同一个根目录下面。定位文件或目录位置时使用“/”进行分隔。根目录是 Linux 文件系统的起点，一个硬盘分区只有挂载到某个目录中才能被访问，这个指定的目录就被称为挂载点。如将分区“/dev/hda2”挂载到根目录“/”，那么通过访问根目录“/”就可以访问到“/dev/hda2”分区，这个分区也就称为根分区。需要注意的是 Linux 同样选择了大小写敏感，这意味着字母的大小写变得非常重要。

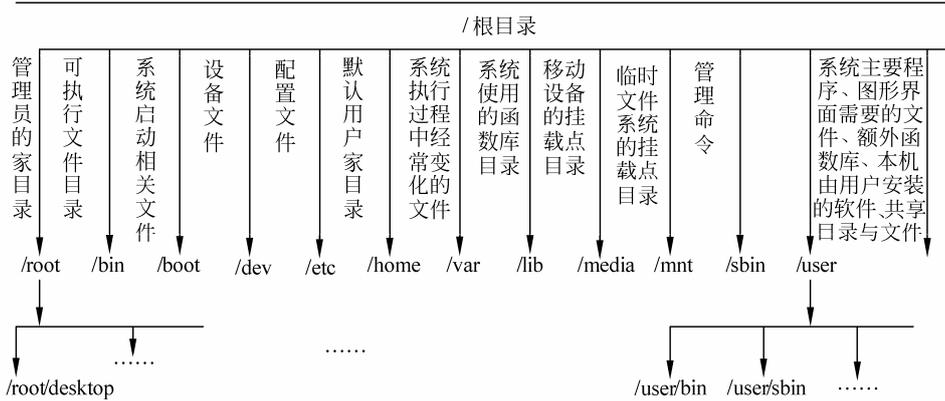


图 3-1 Linux 系统目录结构

3.4 运行机制介绍

Linux 系统的运行机制，主要包括系统的运行级别，系统的启动、引导与修复，root 密码重置及关机过程，下面讲述 Linux 的具体运行机制。

3.4.1 系统运行级

所谓运行级别，简单来说，就是操作系统当前正在运行的功能级别。级别为 0~6，分别具有不同的功能，这些级别定义保存在 `/etc/inittab` 文件中。这个文件是 `init` 程序寻找的主要文件，系统最先运行的服务是那些放在 `/etc/rc.d` 目录下的文件。

1. 理

Linux 下的 7 个运行级别如表 3-2 所示。

3-2 /etc/inittab

0	halt (系统停机)
1	single user mode (单用户维护模式)
2	multi user mode, without NFS (多用户模式, 不支持 NFS 功能)
3	multi user mode, text mode (多用户模式字符界面)
4	reserved (系统保留)
5	multi user mode, graphic mode (多用户模式, 图形化界面)
6	reboot (重启)

各级别说明如下。

0: 系统停机状态，系统默认运行级别不能设置为 0，否则不能正常启动，机器关闭。

1: 单用户工作模式，root 权限，用于系统维护，禁止远程登录，就像 Windows 下的安全模式登录。

- 2: 多用户模式, 不支持 NFS 功能。
- 3: 完整的多用户模式, 有 NFS 支持, 登录后进入控制台命令行模式。
- 4: 系统保留, 一般不用。在一些特殊情况下可以用它来做一些事情, 例如在笔记本电脑的电池用尽时, 可以切换到这个模式来做一些设置。
- 5: X11 控制台, 登录后进入图形 GUI 模式, X Window 系统。
- 6: 系统正常关闭并重启, 默认运行级别不能设为 6, 否则系统不能正常启动。运行 `init6` 机器就会重启。

 标准的 Linux 运行级别为 3 或 5。

关于运行级别的补充说明如下:

- ❑ 在目录 `/etc/rc.d/init.d` 下有许多服务器脚本程序, 一般称为服务 (service)。
- ❑ 在 `/etc/rc.d` 下有 7 个名为 `rcN.d` 的目录, 对应系统的 7 个运行级别。
- ❑ `rcN.d` 目录下都是一些符号链接文件, 这些链接文件都指向 `init.d` 目录下的 service 脚本文件, 命名规则为 `K+nn+服务名` 或 `S+nn+服务名`, 其中 `nn` 为两位数字。
- ❑ 系统会根据指定的运行级别进入到对应的 `rcN.d` 目录, 并按照文件名顺序检索目录下的链接文件: 对于以 `K` (Kill) 开头的文件, 系统将终止对应的服务; 对于以 `S` (Start) 开头的文件, 系统将启动对应的服务。
- ❑ 查看运行级别可用 `runlevel`。
- ❑ 进入其他运行级别可使用 `initN` 命令, 例如运行 `init3` 则进入终端模式, 运行 `init5` 则又登录到图形 GUI 模式。

标准的 Linux 运行级别为 3 或 5, 如果是 3 的话, 系统就运行在多用户状态下; 如果是 5 的话, 则运行在 X Window 系统下。不同的运行级别有不同的用处, 应该根据不同情形来设置。例如, 如果丢失了 `root` 口令, 那么可以让机器启动进入单用户状态来设置。在启动后的 `lilo` 提示符下输入:

```
init=/bin/shrw
```

这样就可以使机器进入运行级别 1, 并把 `root` 文件系统挂载为读写权限。它会跳过所有系统认证, 让用户使用 `passwd` 程序来改变 `root` 口令, 然后以一个新的运行级启动。

2. chkconfig

`chkconfig` 命令可以用来检查、设置系统的各种服务, 使用语法如下:

```
chkconfig [--add] [--del] [--list] [系统服务]
```

或

```
chkconfig [--level<等级代号>] [系统服务] [on/off/reset]
```

参数用法如下。

- ❑ **add** 增加指定的系统服务, 让 `chkconfig` 指令得以管理它, 并同时系统启动的叙述文件内增加相关数据。
- ❑ **del** 删除指定的系统服务, 不再由 `chkconfig` 指令来管理该任务, 并同时系统启动的叙述文件内删除相关数据。



□ `level< >` 指定读系统服务要在哪一个执行等级中开启或关闭。

使用范例：

```
chkconfig --list 列出所有的系统服务
chkconfig --add httpd 增加 httpd 服务
chkconfig --del httpd 删除 httpd 服务
chkconfig --level httpd 2345 on httpd 在运行级别为 2、3、4、5 的情况下设置为 on
(开启) 的状态。
```

`chkconfig` 命令提供了一种简单的方式来设置一个服务的运行级别。例如，为了设置 MySQL 服务器在运行级别为 3 和 4 上运行，用户必须首先将 MySQL 添加为受 `chkconfig` 管理的服务：

```
chkconfig --add mysql
```

现在，假设用户在级别 3 和 5 上设定服务为 “on”，可输入：

```
chkconfig --level 35 mysql on
```

在其他级别上设为 “off”，可输入：

```
chkconfig --level 01246 mysql off
```

为了确认用户的配置被正确地修改了，可以列出服务将会运行的运行级别，输入代码如下所示。

```
chkconfig --list mysql
```

3.4.2 系统启动过程及 GRUB 文件

RHEL 系统的启动按照如下步骤进行，其启动流程如图 3-2 所示。

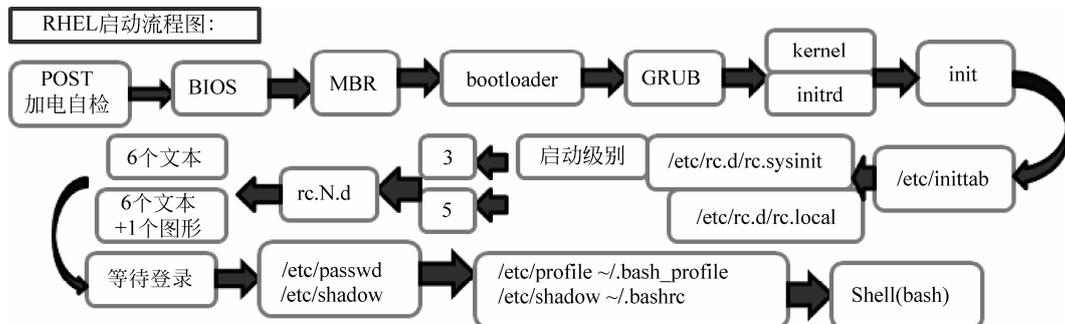


图 3-2 系统启动流程

(1) POST（加电自检）：检测硬件

在 CPU 的控制下，将 RAM 芯片中的某个程序映射到 ROM 地址空间，并执行其中的指令，完成系统硬件健康状况的检查，例如存储设备、网卡、CPU、声卡等硬件设备是否

完好。检查完成后，如果所有硬件或基本硬件、核心硬件没有问题时，则进入下一个启动流程启动 BIOS。

(2) BIOS (Boot Sequence): 决定启动介质

按照 BIOS 所设定的系统启动流程，根据引导次序 (Boot Sequence) 自上而下地寻找对应存储设备上操作系统的 MBR (全称为 Master Boot Record, 即硬盘的主引导记录)，如果 MBR 存在，则会读取 MBR 中的 bootloder。

(3) MBR (bootloder): 寻找 grub, 读取配置文件/etc/grub.conf, 决定默认启动项

根据 MBR 所指引的活动分区，寻找系统分区中的 bootloder。bootloder 是一段程序，占据 446 字节。在 bootloder 中配置了所要引导操作系统的内核所在的位置，因此当 BIOS 被载入内存并实现将控制权限转交给 bootloder 以后，bootloder 即接收整个系统的控制权限；而后根据用户的选择去读取相应操作系统的内核，并将内核装载到内存的某个空间位置，解压缩；这时 kernel 就可以在内存中开始工作，并根据 kernel 本身的功能在内存中搜索硬件，加载硬件驱动程序，完成内核初始化，bootloder 会将控制权限转交给内核。

(4) kernel (初始化): 内核会主动调用 init 进程，读取配置文件/etc/inittab, 决定启动级别，运行初始化脚本/etc/rc.d/rc.sysinit。

步骤是探测硬件→加载驱动 (initrd) →挂载根文件系统→rootfs (/sbin/init)。initrd 是一个虚拟的文件系统，里面有 /、lib、bin、sbin、usr、proc、sys、var、dev、boot 等目录，我们称之为虚拟的根文件系统，作用是将 kernel 和真实的根文件系统建立关联关系，让 kernel 去 initrd 中加载根文件系统所需要的驱动程序，以读写的方式挂载根文件系统，并执行用户当中第一个进程 init。

□ /etc/inittab

/sbin/init 启动会用到/etc/inittab 所定义的条目，如：默认登录级别为 id:3:initdefault:，/etc/inittab 有 0~6 个运行级别。

□ 系统 (/etc/rc.d/rc.sysinit)

- ① 检测，并以读写方式挂载根文件系统。
- ② 设定主机名。
- ③ 检测并挂载/etc/fstab 中的其他文件系统。
- ④ 启动 swap 分区。
- ⑤ 初始化外围硬件设备驱动。
- ⑥ 根据/etc/sysctl.conf 设定内核参数。
- ⑦ 激活 udev 和 seLinux。
- ⑧ 激活 LVM 和 RAID 设备。
- ⑨ 清理过期锁文件和 PID 文件。
- ⑩ 装载键映射 (键盘上每个键的功能)。

□ GRUB 配置 /etc/grub.conf

/etc/grub.conf 文件说明如图 3-3、图 3-4 所示。



```
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/sda2
#         initrd /initrd-[generic]-version.img
#boot=/dev/sda
default=0
#默认的启动项，也就是启动那个title，从0开始计数
timeout=5
#timeout=5，就是默认在启动选择界面停留的时间，等待5秒自动进入默认操作系统
splashimage=(hd0,0)/grub/splash.xpm.gz
#grub启动背景画面，这个可以不要
hiddenmenu
#是隐藏菜单只有按上下键才会出现
title Red Hat Enterprise Linux (2.6.32-279.el6.i686)
#title后面就是系统在启动时候显示的名字
    root (hd0,0)
#root指定你的内核所在的分区，hd0表示第一块硬盘设备，0表示第一个分区 也是/boot所在的分区
    kernel /vmlinuz-2.6.32-279.el6.i686 ro root=UUID=31d213fb-c46d-4b1d-9300-f43c0c7afb94 rd_NO_LUKS KEYBOARDT
YPE=pc KEYTABLE=us rd_NO_MD crashkernel=auto LANG=zh_CN.UTF-8 rd_NO_LVM rd_NO_DM rhgb quiet
#kernel 内核在哪里；ro 只读；root=UUID=,root=/LABEL 都是指定根分区位置；rhgb (redhat graphics boot) 图形化显示启动
过程；quiet (静默式启动，不显示dmesg信息)
    initrd /initramfs-2.6.32-279.el6.i686.img
#initrd内核镜像的名字
"/etc/grub.conf" 25L, 1444C                                     25,1  底端
```

图 3-3 /etc/grub.conf 文件说明

```
[root@localhost ~]# ll /etc/grub.conf
lrwxrwxrwx 1 root root 22 02-28 07:29 /etc/grub.conf -> ../boot/grub/grub.conf
[root@localhost ~]# vi /etc/grub.conf

# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
#         all kernel and initrd paths are relative to /, eg.
#         root (hd0,0)
#         kernel /boot/vmlinuz-version ro root=/dev/sda1
#         initrd /boot/initrd-version.img
#boot=/dev/sda
default=0  定义默认由哪个title来启动系统
timeout=5  定义默认等待时间，值为0不等待
splashimage=(hd0,0)/boot/grub/splash.xpm.gz  定义GRUB界面图片
hiddenmenu  隐藏菜单
password --md5 $1$rv2N/kdv$ZGDLj2NctCe64atkogse6/  启动时，进入GRUB的密码
title centos (2.6.18-194.el5)
    root (hd0,0)  内核所在分区
    kernel /boot/vmlinuz-2.6.18-194.el5 ro root=LABEL=/ rhgb quiet
    initrd /boot/initrd-2.6.18-194.el5.img  内核文件及镜像文件所在位置
title other
    rootnoverify (hd1,3)  当时用U盘安装时，
    chainloader +1  系统增加的启动项
```

图 3-4 /etc/grub.conf 文件说明

文件参数说明如下。

default=0: 默认启动第一个系统，后面的每一个 title 对应一个系统，第一个 title 对应第一个系统，用 0 表示；第二个 title 对应第二个系统，用 1 表示。

timeout=5: 默认启动选择界面时停留的时间，单位是秒。例如设置为 5，表示等待 5 秒后会进入默认操作系统。

splashimage=(hd0,0)/grub/splash.xpm.gz: grub 启动背景画面，配置文件中可以不包括此项内容。

hiddenmenu: 隐藏菜单。只有按上、下键才会出现。

title Red Hat Enterprise Linux (2.6.32-279.el6.i686): title 后面是系统在启动时显示的名字。

root (hd0,0): root 指定内核所在的分区，hd0 表示第一块硬盘设备，0 表示第一个分区，也就是 /boot 所在的分区。

kernel...: 内核所在位置和名字。ro 表示只读，root 用于指定根分区，rhgb 用图形化方式显示启动过程，quiet 表示用静默式方式启动，不显示 dmesg 信息。

initrd /initramfs-2.6.32-279.el6.i686.img: initrd 内核镜像的名字。

rhgb: 表示 redhat graphics boot, 就是用图片来代替启动过程中显示的文本信息。这些信息在启动后用 dmesg 可以看到。

quiet: 表示在启动过程中只显示重要信息, 类似硬件自检的消息则不会显示。

3.4.3 系统引导过程及修复

RHEL 系统启动的引导过程如图 3-5 所示。

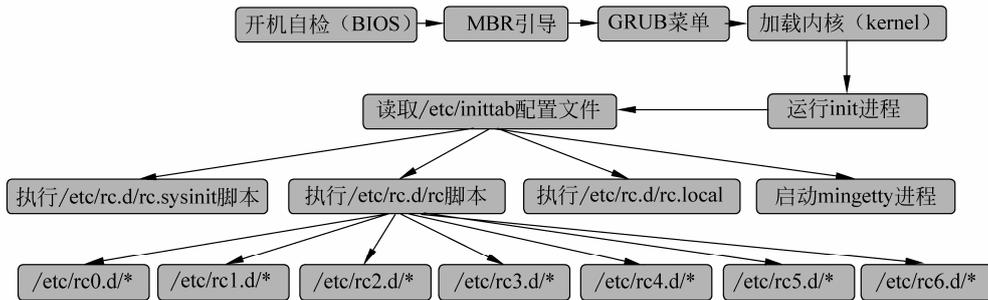


图 3-5 RHEL 系统启动引导过程

引导故障修复主要包含以下两种情况：`grub.conf` 文件损坏和 `/boot` 分区损坏。下面详细说明这两种故障产生的原因及解决方法。

3-1 重启系统停留在如图 3-6 所示的界面，分析故障原因并说明解决方法。

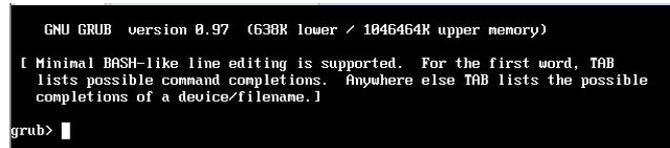


图 3-6 系统出错界面

故障原因：

GRUB 找到不启动文件。

解决步骤：

(1) 手动指定启动文件位置，运行 `root` 或者 `find /vmlinuz-2.6.32-279.el6.i686` (可以使用 `Tab` 键补全)，搜索 `vmlinuz` 所在分区，也就是 `/boot` 所在分区，如图 3-7 所示。

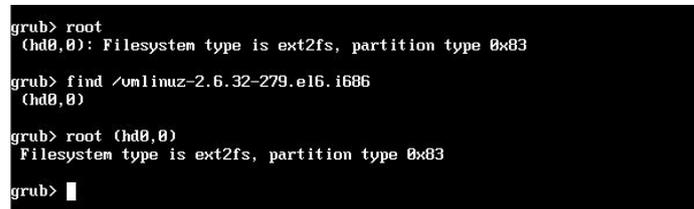


图 3-7 指定启动文件位置



(2) 指定 Linux 内核，执行 `kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet` (结合 Tab 键)，如图 3-8 所示。

(3) 指定 initrd 文件，执行 `initrd /initrd-2.6.32-279.el6.i686.img`，如图 3-9 所示。

```
grub> root
(hd0,0): Filesystem type is ext2fs, partition type 0x83

grub> find /vmlinuz-2.6.32-279.el6.i686
(hd0,0)

grub> root (hd0,0)
Filesystem type is ext2fs, partition type 0x83

grub> kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet
[Linux-bzImage, setup=0x3400, size=0x3aa5e0]

grub> |
```

图 3-8 指定内核

```
grub> initrd /initramfs-2.6.32-279.el6.i686.img
[Linux-initrd @ 0x37199000, 0xe56e83 bytes]

grub> |
```

图 3-9 指定 initrd 文件

(4) 执行 `boot`，如图 3-10 所示。

(5) 出现框中所标的 Red Hat 字样说明内核加载成功，如图 3-11 所示。

```
grub> initrd /initramfs-2.6.32-279.el6.i686.img
[Linux-initrd @ 0x37199000, 0xe56e83 bytes]

grub> boot |
```

图 3-10 执行 boot

```
Memory for crash kernel (0x0 to 0x0) not within permissible range
ACPI: Getting cpuindex for acpiid 0x1
ACPI: Getting cpuindex for acpiid 0x2
ACPI: Getting cpuindex for acpiid 0x3
Red Hat nash version 5.1.19.6 starting
sda: assuming drive cache: write through
sda: assuming drive cache: write through
Welcome to Red Hat Enterprise Linux Server
Press 'I' to enter interactive startup.
Setting clock (utc): Sun Dec 14 15:20:01 CST 2008 [ OK ]
Starting udev: [ OK ]
Loading default keymap (us): [ OK ]
Setting hostname rhe15: [ OK ]
No devices found
Setting up Logical Volume Management: No volume groups found
```

图 3-11 内核加载成功

📖 启动成功后需要在 `grub.conf` 中把刚才执行的三行命令加进去，否则下次启动时还是会出现系统出错界面。如果执行以上操作后还是无法正常进入系统，并提示内核恐慌等信息时，可以在第二步后面加上参数 `enforcing=0`。

3-2 若经过以上操作仍无法正常进入系统，提示内核恐慌，如图 3-12 所示，试说明原因及解决方法。

```
dracut: FATAL: No or empty root= argument
dracut: Refusing to continue

Kernel panic - not syncing: Attempted to kill init!
Pid: 1, comm: init Not tainted 2.6.32-279.el6.i686 #1
Call Trace:
[<c003bfbc>] ? panic+0x68/0x11c
[<c045a501>] ? do_exit+0x741/0x750
[<c045a54c>] ? do_group_exit+0x3c/0xa0
[<c045a5c1>] ? sys_exit_group+0x11/0x20
[<c0409a9f>] ? sysenter_do_call+0x12/0x28
```

图 3-12 内核恐慌提示

故障原因：

提示无法找到 `root` 的参数，也就是使用 `root=LABEL=/` 这个参数后无效。

解决步骤：

(1) 使用安装盘，以救援模式 `rescue` 启动，使用 `chroot` 命令切换到硬盘系统，修改根

目录的 LABEL 名称，如图 3-13 所示。

```

1 #df -h
2 Filesystem      Size  Used Avail Use% Mounted on
3 /dev/sda2       9.9G  2.4G  7.0G  26% /
4 tmpfs           504M  0     504M  0% /dev/shm
5 /dev/sda1       194M  27M  158M  15% /boot
6 /dev/sda5       7.7G  146M  7.2G  2% /home
7 # blkid -s LABEL
8 # blkid -s UUID
9 /dev/sda1: UUID="4e39be01-2b70-4248-938a-0c38cd244b44"
10 /dev/sda2: UUID="d8fe6590-fe20-451e-b5b9-dad15893db25"
11 /dev/sda3: UUID="207ca70d-d291-4e23-841a-1072213f30f2"
12 /dev/sda5: UUID="d344668f-5a57-41f6-9f20-befc25c021a0"
13 # e2label /dev/sda2 /
14 # mount LABEL=/
15 mount: /dev/sda2 already mounted or / busy
16 mount: according to mtab, /dev/sda2 is already mounted on /
17 # vim /etc/fstab
18 #
19 # /etc/fstab
20 # Created by anaconda on Wed Nov 13 17:03:47 2013
21 #
22 # Accessible filesystems, by reference, are maintained under '/dev/disk'
23 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
24 #
25 LABEL=/
26 UUID=4e39be01-2b70-4248-938a-0c38cd244b44 /boot ext4 defaults 1 1
27 UUID=d344668f-5a57-41f6-9f20-befc25c021a0 /home ext4 defaults 1 2
28 UUID=207ca70d-d291-4e23-841a-1072213f30f2 swap defaults 0 0
29 tmpfs /dev/shm tmpfs defaults 0 0
30 devpts /dev/pts devpts gid=5,mode=620 0 0
31 sysfs /sys sysfs defaults 0 0
32 proc /proc
33 # blkid
34 /dev/sda1: UUID="4e39be01-2b70-4248-938a-0c38cd244b44" TYPE="ext4"
35 /dev/sda2: UUID="d8fe6590-fe20-451e-b5b9-dad15893db25" TYPE="ext4" LABEL="/"
36 /dev/sda3: UUID="207ca70d-d291-4e23-841a-1072213f30f2" TYPE="swap"
37 /dev/sda5: UUID="d344668f-5a57-41f6-9f20-befc25c021a0" TYPE="ext4"
38 #

```

图 3-13 内核恐慌解决方案

(2) 再重复上面修复的过程即可，如果以上还是无法修复，可以尝试在【例 3-1】中解决方法的第二步指定内核 `kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet` 后面增加参数 `enforcing=0`。

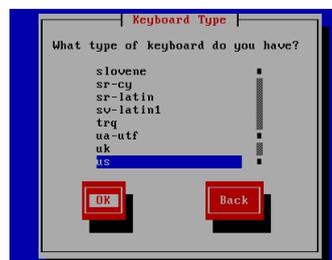
3-3 若/boot 分区损坏重启后出现的界面和上面介绍的 grub 文件损坏时的一样，如何判断是 grub 损坏，还是/boot 分区损坏？在指定内核时无法找到参数应如何解决？

在【例 3-1】中指定内核时发现找不到内核参数，此时，应按照如下方法解决：

(1) 放入安装盘从光盘启动，进入救援模式，执行步骤如图 3-14 所示。



(a)



(b)



(c)



(d)

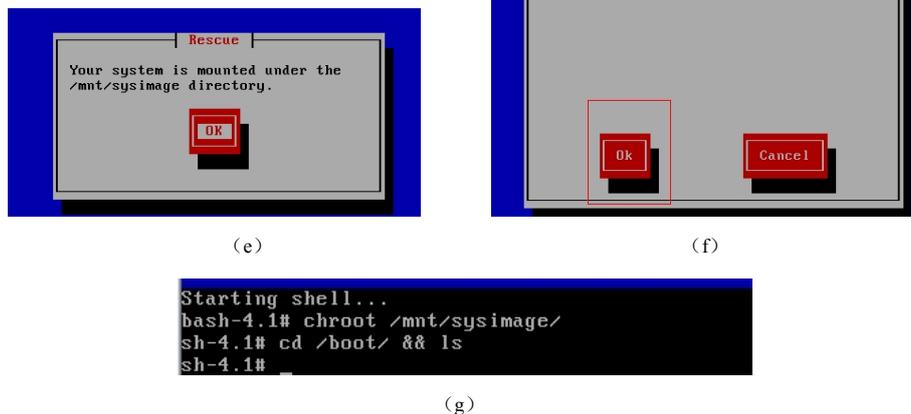


图 3-14 进入救援模式

(2) 当前 boot 目录下一个文件也没有 (boot 下面应主要包含 grub 和 kernel), 因此这里需要安装这两部分内容。先挂载安装盘, 命令为 `mount/dev/cdrom/mnt/cd/`, 如图 3-15 所示。

```
sh-4.1# rpm -ivh -force /mnt/cd/Packages/kernel-2.6.32-279.el6.i686.rpm
-force: unknown option
sh-4.1# rpm -ivh --force /mnt/cd/Packages/kernel-2.6.32-279.el6.i686.rpm
warning: /mnt/cd/Packages/kernel-2.6.32-279.el6.i686.rpm: Header U3 RSA/SHA256 S
ignature, key ID fd431d51: NOKEY
Preparing...
1:kernel
sh-4.1# ls
System.map-2.6.32-279.el6.i686      symvers-2.6.32-279.el6.i686.gz
config-2.6.32-279.el6.i686        vmlinuz-2.6.32-279.el6.i686
initramfs-2.6.32-279.el6.i686.img
```

图 3-15 挂载安装 grub

(3) 然后安装 kernel, 命令为: `rpm -ivh -force/mnt/cd/Packages/kernel-2.6.32-279.el6.i686.rpm`, 如图 3-16 所示。

```
sh-4.1# mount /dev/cdrom /mnt/cd/
mount: block device /dev/sr0 is write-protected, mounting read-only
sh-4.1# find /mnt/cd/ -name "*kernel*"
/mnt/cd/Packages/abrt-addon-kerneloops-2.0.8-6.el6.i686.rpm
/mnt/cd/Packages/dracut-kernel-004-283.el6.noarch.rpm
/mnt/cd/Packages/kernel-2.6.32-279.el6.i686.rpm
/mnt/cd/Packages/kernel-debug-2.6.32-279.el6.i686.rpm
/mnt/cd/Packages/kernel-debug-devel-2.6.32-279.el6.i686.rpm
/mnt/cd/Packages/kernel-devel-2.6.32-279.el6.i686.rpm
/mnt/cd/Packages/kernel-doc-2.6.32-279.el6.noarch.rpm
/mnt/cd/Packages/kernel-firmware-2.6.32-279.el6.noarch.rpm
/mnt/cd/Packages/kernel-headers-2.6.32-279.el6.i686.rpm
/mnt/cd/Packages/libreport-plugin-kerneloops-2.0.9-5.el6.i686.rpm
```

图 3-16 挂载安装 kernel

(4) 安装好 kernel 后还需要安装 grub, 命令为 `grub-install/dev/sda`, 如图 3-17 所示。

(5) 新安装的 grub 没有配置文件, 需要手动添加, 如图 3-18 所示。

(6) 输入两次 `exit` 后退出重启, 重启后会检测磁盘, 如图 3-19 所示。该过程可能会有点长。

```
sh-4.1# grub-install /dev/sda

Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script 'grub-install'.

(fd0) /dev/fd0
(hd0) /dev/sda
(hd1) /dev/sdb
sh-4.1# ls
System.map-2.6.32-279.el6.i686  initramfs-2.6.32-279.el6.i686.img
config-2.6.32-279.el6.i686      symvers-2.6.32-279.el6.i686.gz
grub                             vmlinuz-2.6.32-279.el6.i686
sh-4.1# cd grub/
sh-4.1# ls
device.map  iso9660_stage1_5  stage1  xfs_stage1_5
e2fs_stage1_5  jfs_stage1_5  stage2
fat_stage1_5  minix_stage1_5  ufs2_stage1_5
ffs_stage1_5  reiserfs_stage1_5  vstafs_stage1_5
sh-4.1# _
```

图 3-17 安装 grub

```
default=0
timeout=5
hiddenmenu
title Red Hat Enterprise Linux 6.3
root (hd0,0)
kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet
initrd /initramfs-2.6.32-279.el6.i686.img_
```

图 3-18 添加 grub 配置文件

```
Red Hat Enterprise Linux Server release 6.3 (Santiago)
Kernel 2.6.32-279.el6.i686 on an i686

justin login: _
```

图 3-19 退出重启

3.4.4 root 密码重置

恢复 root 密码需要在单用户模式下对 root 账户进行初始化,对 root 账户重新设置密码的步骤如下。

(1) 出现如图 3-20 所示的界面时,立即按字母键 E,之后出现如图 3-21 所示的界面。

```
Press any key to enter the menu

Booting Red Hat Enterprise Linux 6.3 in 4 seconds..._
```

图 3-20 系统启动按字母键 E 界面

```
GNU GRUB  version 0.97  (638K lower / 1046464K upper memory)

Red Hat Enterprise Linux 6.3

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.
```

图 3-21 按字母键 E 后出现的界面



(2) 在出现图 3-21 所示界面后，再按字母键 E，进入如图 3-22 所示的界面。

```
GNU GRUB version 0.97 (638K lower / 1046464K upper memory)

root (hd0,0)
kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet
initrd /initramfs-2.6.32-279.el6.i686.img

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

图 3-22 再按字母键 E 后的界面

(3) 在图 3-22 所示界面中选中第二行 (kernel /vm...) 后，再按字母键 E 进入如图 3-23 所示的界面。

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time cancels. ENTER
at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet_
```

图 3-23 选中第二行再按字母键 E 后的界面

(4) 在图 3-23 所示界面中输入空格和数字 1，如图 3-24 所示。

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time cancels. ENTER
at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet 1_
```

图 3-24 输入空格和 1 后的界面

(5) 按 Enter 键后进入如图 3-25 所示界面。

```
GNU GRUB version 0.97 (638K lower / 1046464K upper memory)

root (hd0,0)
kernel /vmlinuz-2.6.32-279.el6.i686 ro root=LABEL=/ rhgb quiet 1
initrd /initramfs-2.6.32-279.el6.i686.img

Use the ↑ and ↓ keys to select which entry is highlighted.
```

图 3-25 按回车键后的界面

(6) 在图 3-25 所示界面中按字母键 B，进入如图 3-26 所示界面，然后修改需要修改的账户密码。

(7) 重启后即可用修改后的密码来登录。

```
telling INIT to go to single user mode.
init: rc main process (1078) killed by TERM signal
[root@justin /]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@justin /]# reboot_
```

图 3-26 按字母键 B 后的界面

3.4.5 系统关机过程

在了解 Linux 系统关机过程之前，有必要先学习 Linux 一些常用的关机命令。在 Linux 下，常用的关机/重启命令有 `shutdown`、`halt`、`reboot` 和 `init` 等，它们都可以达到关机重启系统的目的，但是每一个命令的内部工作过程却是不同的。通过学习这些可以更加灵活地运用各种关机命令，并进一步详细了解 Linux 系统的安全关机过程。

1. `shutdown` 系统的

`shutdown` 命令可安全地将系统关机。有些用户会使用直接断掉电源的方式来关闭 Linux，这种方法是十分危险的。因为 Linux 与 Windows 不同，其后台运行着许多进程，所以强制关机可能会导致进程的数据丢失，使系统处于不稳定的状态，甚至在有的系统中会损坏硬件设备。而在系统关机前使用 `shutdown` 命令，系统管理员会通知所有登录的用户系统将要关闭。并且 `loginm` 命令会被冻结，即新的用户不能登录。直接关机、延迟一定的时间关机或者重启都是可能，这是由所有进程收到系统所送达的信号决定的。这让像 `vi` 之类的程序有时间来储存目前正在编辑的文档，而像处理邮件和新闻的程序则可以正常地离开等。

`shutdown` 的任务是送信号给 `init` 程序，要求它改变 `runlevel`。`runlevel 0` 用来停机，`runlevel 6` 用来重新激活系统，而 `runlevel 1` 则是用来让系统进入运行管理状态。要想了解在停机或者重新开机过程中做了哪些动作，可以在 `/etc/inittab` 文件里查看 `runlevels` 的相关资料。

`shutdown` 的参数说明如下：

`[-t]` 在改变到其他 `runlevel` 之前，告诉 `init` 多久以后关机。

`[-r]` 重启计算机。

`[-k]` 并不真正关机，只是发送警告信号给每位登录的用户（`login`）。

`[-h]` 关机后关闭电源（`halt`）。

`[-n]` 不用 `init`，而是用户自己来关机。不鼓励使用这个选，该选项所产生的后果往往不是预料得到的。

`[-c]` `cancel current process` 取消目前正在执行的关机程序。这个选项没有时间参数，但是可以输入一个用来解释的信息，该信息将会发送到每位登录的用户。

`[-f]` 在重启计算机（`reboot`）时忽略 `fsck` 命令。

`[-F]` 在重启计算机（`reboot`）时强制执行 `fsck` 命令。

`[-time]` 设定关机（`shutdown`）前的时间。



3-4 立即关机重启。

命令如下：

```
shutdown -r now
```

3-5 立即关机。

命令如下：

```
shutdown -h now
```

3-6 设定 10 分钟后关机，同时发出警告信息给其他登录的用户。

命令如下：

```
shutdown +10 "system will shut down after 10 minutes"
```

2. halt 系统的

halt 是最简单的关机命令，其实 halt 调用的就是 shutdown -h。执行 halt 时，会杀死应用进程，执行 sync 系统调用，文件系统写操作完成后就会停止内核的工作。

halt 的参数说明如下：

[-n] 防止 sync 系统调用。它用在使用 fsck 修补根分区之后，以阻止内核用老版本的超级块（superblock）来覆盖修补过的超级块。

[-w] 并不是真正的重启或关机，只是写 wtmp（/var/log/wtmp）记录。

[-d] 不写 wtmp 记录（已包含在选项[-n]中）。

[-f] 不调用 shutdown 而强制关机或重启。

[-i] 关机（或重启）前，关掉所有的网络接口。

[-p] 该选项为缺省选项，就是关机时调用 poweroff。

3. reboot

reboot 的工作过程和参数都与 halt 相差不多，不过它用于主机重启，而 halt 是关机。

4. telinit init 的

init 是所有进程的祖先，它的进程号始终为 1，所以发送 TERM 信号给 init 会终止所有的用户进程、守护进程等，shutdown 使用的就是这种机制。init 定义了 8 个运行级别（runlevel），其中 init 0 为关机，init 1 为重启。telinit 命令可以改变 init 的运行级别，比如，telinit -iS 可使系统进入单用户模式。在使用 telinit 命令的时候，不会得到像 shutdown 那样友好的警告信息，也得不到使用 shutdown 时的信息和等待时间，所以很少使用。不过对于执行测试 inittab 文件的修改效果来说，使用该命令还是比较方便的。

5. poweroff

poweroff 命令等价于 halt，不同之处在于，Linux 关闭后 poweroff 可以向电源管理系统（在支持这项功能的系统上）发送一则请求来关闭系统的主电源，这项功能易于实现远程关机。Linux 的文件系统缓冲一般在内存中变化，只是偶尔才把它们写回磁盘。这种方案使得磁盘 I/O 速度更快，但是当系统被粗暴地中止时，这种方式也更容易使文件系统丢失数据。传统的装有 Unix 和 Linux 系统的计算机要非常小心地处理关机方式才行，虽然现在的系统已经变得强壮了（尤其是使用 ext3 这样强健的文件系统时），但是妥善关机总是

一个好习惯。

关闭或者重新引导系统有许多种方法，比如关闭电源，但即使是在桌面系统上，关闭电源也不是关闭系统的一个好办法，这样做有可能丢失数据和破坏文件系统。许多计算机有一个软电源开关，当用户按下电源按钮时，计算机实际上是在运行一串命令来执行关闭系统的正确操作。不过，这种情况也是相对的，当发生紧急情况时，如果没有足够的时间来妥善关机，那么关闭电源也是可以的。当然，Linux 系统下的关机操作更为常用是使用上述 shutdown 命令、halt 或者 reboot 命令、以 telinit 命令改变 init 运行级别及 poweroff 命令关闭系统电源等方法。

3.5 Linux 与 SecureCRT

初学者会觉得在 Linux 操作系统上进行学习或者维护不够方便，比如要上传文件到 Linux 中，可能需要挂载 U 盘；公司有很多服务器放在较远的机房里，一旦应用出问题就需要去机房维护等等。其实，这些问题都是不用担心的，使用 VanDyke SecureCRT 软件可以很好地解决上述问题。

3.5.1 SecureCRT 概述

SecureCRT 是一款支持 SSH（SSH1 和 SSH2）的终端仿真程序，简单地说是 Windows 环境下登录到 Unix 或 Linux 服务器主机的软件。SecureCRT 同时支持 Telnet 和 rlogin 协议。SecureCRT 是一款用于连接运行包括 Windows、Unix 和 VMS 的理想工具。通过使用内含的 VCP 命令行程序可以进行加密文件的传输。具有流行 CRTTelnet 客户机的所有特点，包括：自动注册、对不同主机保持不同的特性、打印功能、颜色设置、可变屏幕尺寸等。其他特点包括易于使用的工具条、用户的键位图编辑器、可定制的 ANSI 颜色等。SecureCRT 的 SSH 协议支持 DES、3DES、RC4 和 RSA 密码。

SSH 的英文全称是 Secure Shell。传统的网络服务程序，如 ftp 和 telnet 在本质上都是不安全的，因为它们在网上用明文传送口令和数据，别有用心的人非常容易就可以截获这些口令和数据。而 SSH 客户端与服务器端通讯时，对用户名及口令均进行了加密，有效防止了对口令的窃取。同时，通过 SSH 的数据传输是经过压缩的，所以可以提高数据的传输速度。SSH 是由客户端和服务端的软件组成的，有两个不兼容的版本，分别是 1.x 和 2.x。至于具体如何安装到服务器端，普通用户就不需要关心了。

3.5.2 SecureCRT 的安装与使用

SecureCRT 的最新版本是 7.x，在因特网中有各种版本可以下载，其安装也很简单，这里不再赘述。SecureCRT 的工作界面如图 3-27 所示。

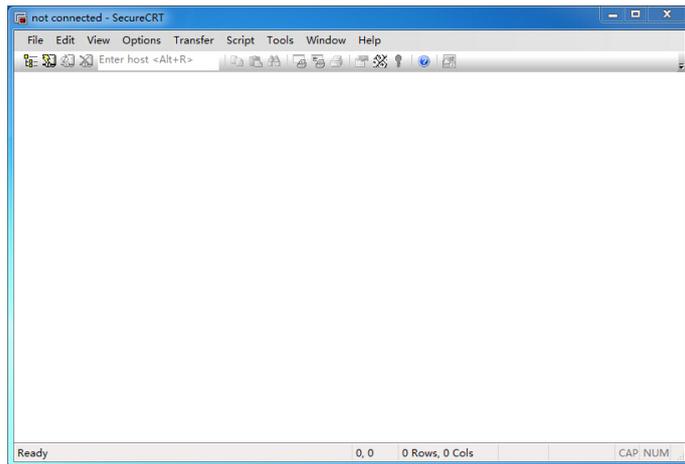


图 3-27 SecureCRT 界面

安装完毕后，用户就可以使用 SecureCRT 连接各种程序。

1. SecureCRT

步骤如下。

(1) 单击菜单栏【file】/【Quick Connect】选项或者按快捷键 Alt+Q，如图 3-28 所示。

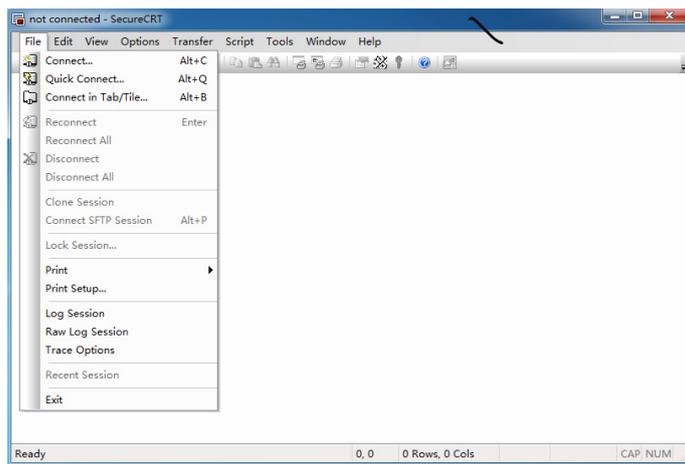


图 3-28 选择菜单选项

(2) 单击快捷连接按钮，如图 3-29 所示圆圈标注处。

2. SSH

步骤如下。

(1) 打开快速连接窗口。

(2) 在协议【Protocol】下拉列表中选择“SSH2”。

(3) 在【Hostname】输入框中输入服务端的 IP 地址。

(4) 在【Username】输入框中输入需要登录的用户名，如图 3-30 所示。

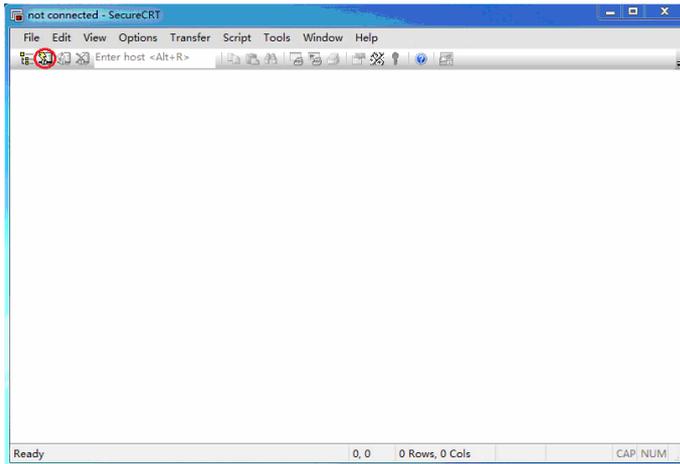


图 3-29 单击快捷连接按钮

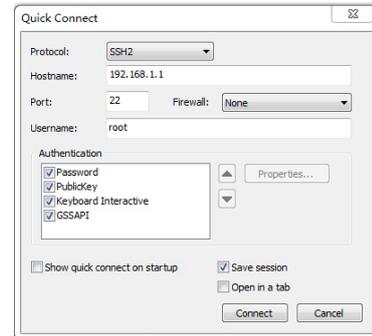


图 3-30 连接 SSH 界面

3. Telnet

步骤如下。

- (1) 打开快速连接窗口。
- (2) 在协议【Protocol】下拉列表中选择“Telnet”。
- (3) 在【Hostname】输入框中输入服务端的 IP 地址，如图 3-31 所示。

4. Serial

步骤如下。

- (1) 打开快速连接窗口
- (2) 在协议【Protocol】下拉列表中选择“Serial”。
- (3) 在【Port】下拉列表中选择需要连接的 com 端口。
- (4) 在【Baud rate】波特率中设置正确的波特率。
- (5) 在【Flow Control】选项组中的【RTS/CTS】项一定要取消选取，否则无法输入，如图 3-32 所示。

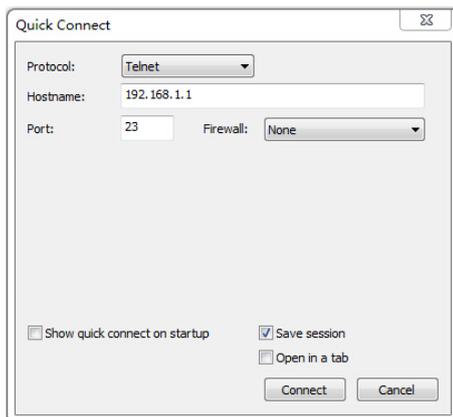


图 3-31 连接 Telnet 界面

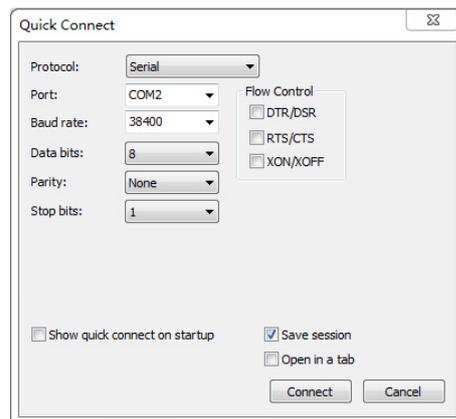


图 3-32 连接 Serial 界面

成功连接后，可以进行风格和字体的设置。

(1) 选择菜单栏【Options】/【Session Options】选项，如图 3-33 所示。之后，界面将如图 3-34 所示。

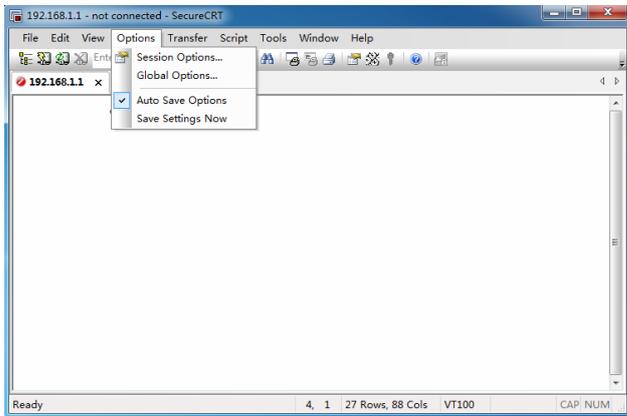


图 3-33 SecureCRT 界面设置 1

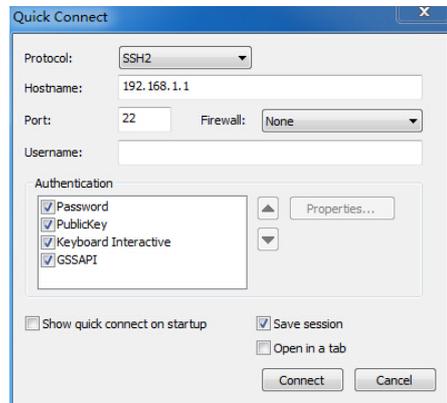


图 3-34 SecureCRT 界面设置 2

(2) 单击【Connect】按钮进行连接。如果连接到远端 Linux 系统，会提示输入密码。输入密码验证成功后，就登录到 Linux 系统了。此时就可以在本地管理和维护远程的 Linux 系统了。

(3) 还可以进一步设置界面的仿真风格，在【Terminal】/【Emulation (仿真)】中，将【Terminal】设置为【Linux】，并将【ANSI Color】和【User color scheme】复选框选取上，如图 3-35 所示。

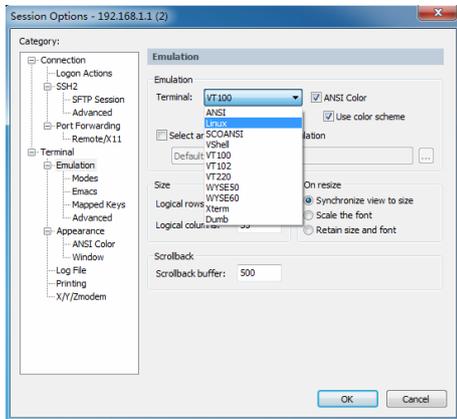


图 3-35 设置 SecureCRT 仿真风格

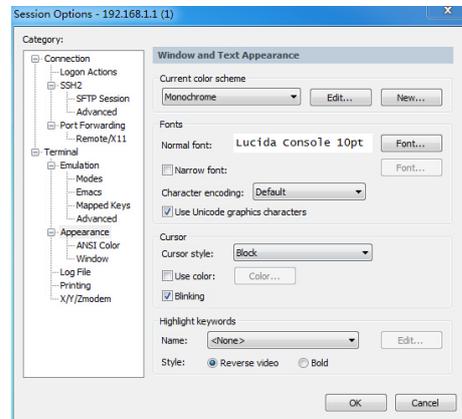


图 3-36 设置 SecureCRT 外观

3.5.3 与 SecureCRT 相关的 Linux 命令

SecureCRT 可以使用 Linux 下的 zmodem 协议来快速地传送文件，使用非常方便。只

要在【options】/【session options】/【file transfer】下设置上传和下载默认目录即可。如果从 Windows 上传一个文件到 Linux 系统，只需执行命令 rz，然后根据弹出对话框的提示输入相应的文件路径即可。如果要从 Linux 系统上下载一个文件到本地，只需执行如下命令即可。

```
sz filename (下载文件路径)
```

这样，该文件就能下载到本地了。默认情况下从 Linux 系统下载到本地的文件都存放在 SecureCRT 安装目录下的 download 目录下。SecureCRT 延续了 Windows 下的复制和粘贴等功能。复制命令的快捷键是 Ctrl+Insert，粘贴命令的快捷键是 Shift+Insert。当然用户也可以用单击鼠标右键选择菜单选项的方式进行复制粘贴操作。

3.6 本章小结

本章从整体上讲述了 Linux 系统的基本结构，首先介绍了 Linux 与硬件的关系，接着概要介绍了 Linux 的文件系统结构，然后详细讲述了 Linux 系统运行级的概念和含义，Linux 系统的启动、引导等过程以及各种关机操作的异同，最后讲解了与 Linux 相关的 SecureCRT 软件的安装和使用。

3.7 思考与练习

- (1) 尝试在 Linux 系统上挂载软盘、光盘和 U 盘。
- (2) 尝试在 Linux 系统上卸载软盘、光盘和 U 盘。
- (3) Linux 系统有几个运行级别，每个运行级别的含义是什么？
- (4) Linux 有几种关机方法，每种关机方法有何异同？