

入门学习 Linux 常用必会 60 个命令实例详解

Linux 必学的 60 个命令

Linux 提供了大量的命令，利用它可以有效地完成大量的工作，如磁盘操作、文件存取、目录操作、进程管理、文件权限设定等。所以，在 Linux 系统上工作离不开使用系统提供的命令。要想真正理解 Linux 系统，就必须从 Linux 命令学起，通过基础的命令学习可以进一步理解 Linux 系统。

不同 Linux 发行版的命令数量不一样，但 Linux 发行版本最少的命令也有 200 多个。这里笔者把比较重要和使用频率最多的命令，按照它们在系统中的作用分成下面六个部分一一介绍。

◆ 安装和登录命令: login、shutdown、halt、reboot、install、mount、umount、chsh、exit、last;

◆ 文件处理命令: file、mkdir、grep、dd、find、mv、ls、diff、cat、ln;

◆ 系统管理相关命令: df、top、free、quota、at、lp、adduser、groupadd、kill、crontab;

◆ 网络操作命令: ifconfig、ip、ping、netstat、telnet、ftp、route、rlogin、rcp、finger、mail、nslookup;

◆ 系统安全相关命令: passwd、su、umask、chgrp、chmod、chown、chattr、sudo ps、who;

◆ 其它命令: tar、unzip、gunzip、unrarj、mtools、man、unendcode、uudecode。

本文以 Mandrake Linux 9.1(Kernel 2.4.21)为例，介绍 Linux 下的安装和登录命令。

immortality 按：请用 ctrl+f 在本页中查找某一部分的内容或某一命令的用法。

Linux 必学的 60 个命令(1)-安装与登陆命令

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:用来向远程登录的之间传输用户名。

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

一般界面是这样的：

```
Mandrake Linux release 9.1(Bamboo) for i586
```

```
renrel 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，我们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。

4.应用技巧

Linux 是一个真正的多用户操作系统，可以同时接受多个用户登录，还允许一个用户进行多次登录。这是因为 Linux 和许多版本的 Unix 一样，提供了虚拟控制台的访问方式，允许用户在同一时间从控制台（系统的控制台是与系统直接相连的监视器和键盘）进行多次登录。每个虚拟控制台可以看作是一个独立的工作站，工作台之间可以切换。虚拟控制台的切换可以通过按下 Alt 键和一个功能键来实现，通常使用 F1-F6 。

例如，用户登录后，按一下“Alt+ F2”键，用户就可以看到上面出现的“login:”提示符，说明用户看到了第二个虚拟控制台。然后只需按“Alt+ F1”键，就可以回到第一个虚拟控制台。一个新安装的 Linux 系统允许用户使用“Alt+F1”到“Alt+F6”键来访问前六个虚拟控制台。虚拟控制台最有用的是，当一个程序出错造成系统死锁时，可以切换到其它虚拟控制台工作，关闭这个程序。

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.重要参数

-t: 在改变到其它运行级别之前，告诉 **init** 程序多久以后关机。

-k: 并不真正关机，只是送警告信号给每位登录者。

-h: 关机后关闭电源。

-c: **cancel current process** 取消目前正在执行的关机程序。所以这个选项当然没有时间参数，但是可以输入一个用来解释的讯息，而这信息将会送到每位使用者。

-F: 在重启计算机时强迫 **fsck**。

-time: 设定关机前的时间。

-m: 将系统改为单用户模式。

-i: 关机时显示系统信息。

4.命令说明

shutdown 命令可以安全地将系统关机。有些用户会使用直接断掉电源的方式来关闭 Linux 系统，这是十分危险的。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（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。

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.主要参数说明

-n: 防止 **sync** 系统调用，它用在用 **fsck** 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) 纪录。

—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。

—f: 强迫关机, 不呼叫 shutdown 这个指令。

—p: 当关机的时候顺便做关闭电源的动作。

—d: 关闭系统, 但不留下纪录。

4.命令说明

halt 就是调用 shutdown -h。halt 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(加上一h 参数)来取代。

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] [-f] [-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 这个参数包含了一d)。

—i: 在重开机之前先把所有与网络相关的装置停止。

install

1.作用

install 命令的作用是安装或升级软件或备份数据, 它的使用权限是所有用户。

2.格式

(1)install [选项]... 来源 目的地

(2)install [选项]... 来源... 目录

(3)install -d [选项]... 目录...

在前两种格式中，会将<来源>复制至<目的地>或将多个<来源>文件复制至已存在的<目录>，同时设定权限模式及所有者/所属组。在第三种格式中，会创建所有指定的目录及它们的主目录。长选项必须用的参数在使用短选项时也是必须的。

3.主要参数

- backup[=CONTROL]: 为每个已存在的目的地文件进行备份。
- b: 类似 --backup, 但不接受任何参数。
- c: (此选项不作处理)。
- d, --directory: 所有参数都作为目录处理，而且会创建指定目录的所有主目录。
- D: 创建<目的地>前的所有主目录，然后将<来源>复制至 <目的地>; 在第一种使用格式中有用。
- g, --group=组: 自行设定所属组，而不是进程目前的所属组。
- m, --mode=模式: 自行设定权限模式 (像 chmod), 而不是 rwxr-xr-x。
- o, --owner=所有者: 自行设定所有者 (只适用于超级用户)。
- p, --preserve-timestamps: 以<来源>文件的访问/修改时间作为相应的目的地文件的时间属性。
- s, --strip: 用 strip 命令删除 symbol table, 只适用于第一及第二种使用格式。
- S, --suffix=后缀: 自行指定备份文件的<后缀>。
- v, --verbose: 处理每个文件/目录时印出名称。
- help: 显示此帮助信息并离开。
- version: 显示版本信息并离开。

mount

1.作用

mount 命令的作用是加载文件系统,它的用权限是超级用户或/etc/fstab 中允许的使用者。

2.格式

mount -a [-fv] [-t vfstype] [-n] [-rw] [-F] device dir

3.主要参数

—h: 显示辅助信息。

—v: 显示信息，通常和—f用来除错。

—a: 将/etc/fstab 中定义的所有文件系统挂上。

—F: 这个命令通常和—a 一起使用，它会为每一个 mount 的动作产生一个行程负责执行。在系统需要挂上大量 NFS 文件系统时可以加快加载的速度。

—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。

—t vfstype: 显示被加载文件系统的类型。

—n: 一般而言，mount 挂上后会在/etc/mtab 中写入一笔资料，在系统中没有可写入文件系统的情况下，可以用这个选项取消这个动作。

4.应用技巧

在 Linux 和 Unix 系统上，所有文件都是作为一个大型树（以/为根）的一部分访问的。要访问 CD-ROM 上的文件，需要将 CD-ROM 设备挂装在文件树中的某个挂装点。如果发行版安装了自动挂装包，那么这个步骤可自动进行。在 Linux 中，如果要使用硬盘、光驱等储存设备，就得先将它加载，当储存设备挂上了之后，就可以把它当成一个目录来访问。挂上一个设备使用 mount 命令。在使用 mount 这个指令时，至少要先知道下列三种信息：要加载对象的文件系统类型、要加载对象的设备名称及要将设备加载到哪个目录下。

（1）Linux 可以识别的文件系统

◆ Windows 95/98 常用的 FAT 32 文件系统: vfat ；

◆ Win NT/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；

◆ CD-ROM 光盘用的文件系统: iso9660。

虽然 vfat 是指 FAT 32 系统，但事实上它也兼容 FAT 16 的文件系统类型。

（2）确定设备的名称

在 Linux 中，设备名称通常都存在/dev 里。这些设备名称的命名都是有规则的，可以用“推理”的方式把设备名称找出来。例如，/dev/hda1 这个 IDE 设备，hd 是 Hard Disk(硬盘)的，sd 是 SCSI Device，fd 是 Floppy Device(或是 Floppy Disk?)。a 代表第一个设

备, 通常 IDE 接口可以接上 4 个 IDE 设备(比如 4 块硬盘)。所以要识别 IDE 硬盘的方法分别就是 hda、hdb、hdc、 hdd。hda1 中的“1”代表 hda 的第一个硬盘分区 (partition), hda2 代表 hda 的第二主分区, 第一个逻辑分区从 hda5 开始, 依此类推。此外, 可以直接检查 /var/log/messages 文件, 在该文件中可以找到计算机开机后系统已辨认出来的设备代号。

(3) 查找挂载点

在决定将设备挂载之前, 先要查看一下计算机是不是有个/mnt 的空目录, 该目录就是专门用来当作挂载点(Mount Point)的目录。建议在/mnt 里建几个/mnt/cdrom、/mnt/floppy、/mnt/mo 等目录, 当作目录的专用挂载点。举例而言, 如要挂载下列 5 个设备, 其执行指令可能如下 (假设都是 Linux 的 ext2 系统, 如果是 Windows XX 请将 ext2 改成 vfat):

```
软盘 ==>mount -t ext2 /dev/fd0 /mnt/floppy
```

```
cdrom ==>mount -t iso9660 /dev/hdc /mnt/cdrom
```

```
SCSI cdrom ==>mount -t iso9660 /dev/sdb /mnt/scdrom
```

```
SCSI cdr ==>mount -t iso9660 /dev/sdc /mnt/scdr
```

不过目前大多数较新的 Linux 发行版本 (包括红旗 Linux、中软 Linux、Mandrake Linux 等) 都可以自动挂装文件系统, 但 Red Hat Linux 除外。

umount

1.作用

umount 命令的作用是卸载一个文件系统, 它的使用权限是超级用户或/etc/fstab 中允许的使用者。

2.格式

```
umount -a [-fFnrsvw] [-t vfstype] [-n] [-rw] [-F] device dir
```

3.使用说明

umount 命令是 mount 命令的逆操作, 它的参数和使用方法和 mount 命令是一样的。Linux 挂装 CD-ROM 后, 会锁定 CD-ROM, 这样就不能用 CD-ROM 面板上的 Eject 按钮弹出它。但是, 当不再需要光盘时, 如果已将/cdrom 作为符号链接, 请使用 umount/cdrom 来卸装它。仅当无用户正在使用光盘时, 该命令才会成功。该命令包括了将带有当前工作目录当作该光盘中的目录的终端窗口。

chsh

1.作用

chsh 命令的作用是更改使用者 shell 设定，它的使用权限是所有使用者。

2.格式

```
chsh [ -s ] [ -list ] [ --help ] [ -v ] [ username ]
```

3.主要参数

-l: 显示系统所有 Shell 类型。

-v: 显示 Shell 版本号。

4.应用技巧

前面介绍了 Linux 下有多种 Shell，一般缺省的是 Bash，如果想更换 Shell 类型可以使用 chsh 命令。先输入账户密码，然后输入新 Shell 类型，如果操作正确系统会显示“Shell change”。其界面一般如下：

```
Changing fihanging shell for cao
```

```
Password:
```

```
New shell [/bin/bash]: /bin/tcsh
```

上面代码中，[]内是目前使用的 Shell。普通用户只能修改自己的 Shell，超级用户可以修改全体用户的 Shell。要想查询系统提供哪些 Shell，可以使用 chsh -l 命令，见图 1 所示。

图 1 系统可以使用的 Shell 类型

从图 1 中可以看到，笔者系统中可以使用的 Shell 有 bash（缺省）、csh、sh、tcsh 四种。

```
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。

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][-l -IP][-1][-y][1D]

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。

-1: 用 IP 来显示远端地址。

-y: 显示记录的年、月、日。

-ID: 知道查询的用户名。

-x:显示系统关闭、用户登录和退出的历史。

动手练习

上面介绍了 **Linux** 安装和登录命令，下面介绍几个实例，动手练习一下刚才讲过的命令。

1.一次运行多个命令

在一个命令行中可以执行多个命令，用分号将各个命令隔开即可，例如：

```
#last -x; halt
```

上面代码表示在显示系统关闭、用户登录和退出的历史后关闭计算机。

2.利用 mount 挂装文件系统访问 Windows 系统

许多 **Linux** 发行版本现在都可以自动加载 **Vfat** 分区来访问 **Windows** 系统，而 **Red Hat** 各个版本都没有自动加载 **Vfat** 分区，因此还需要进行手工操作。

mount 可以将 Windows 分区作为 Linux 的一个“文件”挂接到 Linux 的一个空文件夹下，从而将 Windows 的分区和/mnt 这个目录联系起来。因此，只要访问这个文件夹就相当于访问该分区了。首先要在/mnt 下建立 winc 文件夹，在命令提示符下输入下面命令：

```
# mount -t vfat /dev/hda1 /mnt/winc
```

即表示将 Windows 的 C 分区挂到 Linux 的/mnt/winc 目录下。这时，在/mnt/winc 目录下就可以看到 Windows 中 C 盘的内容了。使用类似的方法可以访问 Windows 系统的 D、E 盘。在 Linux 系统显示 Windows 的分区一般顺序这样的：hda1 为 C 盘、hda5 为 D 盘、hda6 为 E 盘……以此类推。上述方法可以查看 Windows 系统有一个很大的问题，就是 Windows 中的所有中文文件名或文件夹名全部显示为问号“？”，而英文却可以正常显示。我们可以通过加入一些参数让它显示中文。还以上面的操作为例，此时输入命令：

```
# mount -t vfat -o iocharset=cp936 /dev/hda1 /mnt/winc
```

现在它就可以正常显示中文了。

3.使用 mount 加挂闪盘上的文件系统

在 Linux 下使用闪盘非常简单。Linux 对 USB 设备有很好的支持，当插入闪盘后，闪盘被识别为一个 SCSI 盘，通常输入以下命令：

```
# mount /dev/sda1 /usb
```

就能够加挂闪盘上的文件系统。

小知识

Linux 命令与 Shell

所谓 Shell，就是命令解释程序，它提供了程序设计接口，可以使用程序来编程。学习 Shell 对于 Linux 初学者理解 Linux 系统是非常重要的。Linux 系统的 Shell 作为操作系统的外壳，为用户提供了使用操作系统的接口。Shell 是命令语言、命令解释程序及程序设计语言的统称，是用户和 Linux 内核之间的接口程序。如果把 Linux 内核想象成一个球体的中心，Shell 就是围绕内核的外层。当从 Shell 或其它程序向 Linux 传递命令时，内核会做出相应的反应。Shell 在 Linux 系统的作用和 MS DOS 下的 COMMAND.COM 和 Windows 95/98 的 explorer.exe 相似。Shell 虽然不是系统核心的一部分，只是系统核心的一个外延，但它能够调用系统内核的大部分功能。因此，可以说 Shell 是 Unix/Linux 最重要的实用程序。

Linux 中的 Shell 有多种类型，其中最常用的是 Bourne Shell(sh)、C Shell(csh)和 Korn Shell(ksh)。大多数 Linux 发行版本缺省的 Shell 是 Bourne Again Shell，它是 Bourne Shell

的扩展，简称 **bash**，与 **Bourne Shell** 完全向后兼容，并且在 **Bourne Shell** 的基础上增加了很多特性。**bash** 放在 `/bin/bash` 中，可以提供如命令补全、命令编辑和命令历史表等功能。它还包含了很多 **C Shell** 和 **Korn Shell** 中的优点，有灵活和强大的编程接口，同时又有很友好的用户界面。**Linux** 系统中 200 多个命令中有 40 个是 **bash** 的内部命令，主要包括 `exit`、`less`、`lp`、`kill`、`cd`、`pwd`、`fc`、`fg` 等。

Linux 必学的 60 个命令(2)-文件处理命令

Linux 系统信息存放在文件里，文件与普通的公务文件类似。每个文件都有自己的名字、内容、存放地址及其它一些管理信息，如文件的用户、文件的大小等。文件可以是一封信、一个通讯录，或者是程序的源语句、程序的数据，甚至可以包括可执行的程序和其它非正文内容。**Linux** 文件系统具有良好的结构，系统提供了很多文件处理程序。这里主要介绍常用的文件处理命令。

file

1.作用 件内容判断文件类型，使用权限是所有用户。

2.格式

file 通过探测文

file [options] 文件名

3.[options]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.简单说明

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 **Shell Script** 文件，或者是其它的什么格式。**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。

5.应用实例

如果我们看到一个没有后缀的文件 **grap**，可以使用下面命令：

```
$ file grap
```

```
grap:  English text
```

此时系统显示这是一个英文文本文件。需要说明的是，**file** 命令不能探测包括图形、音频、视频等多媒体文件类型。

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.[options]主要参数

-m, **--mode=模式**: 设定权限<模式>, 与 **chmod** 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.应用实例

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 **rw**x(即读、写、执行的权限), 那么可以使用以下命令:

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

grep

1.作用

grep 命令可以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

[options]主要参数:

- c: 只输出匹配行的计数。
- l: 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。

pattern 正则表达式主要参数:

- \: 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。
- *: 有字符, 长度可以为 0。

正则表达式是 Linux/Unix 系统中非常重要的概念。正则表达式 (也称为“regex”或“regexp”) 是一个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, 我们就说这个字符和该正则表达式匹配 (Match)。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”操作和其它功能。

4.应用实例

查询 DNS 服务是日常工作之一, 这意味着要维护覆盖不同网络的大量 IP 地址。有时 IP 地址会超过 2000 个。如果要查看 nnn.nnn 网络地址, 但是却忘了第二部分中的其余部分, 只知到有两个句点, 例如 nnn nn..。要抽取其中所有 nnn.nnn IP 地址, 使用[0-9]{3

`\.[0-0{3}\]`。含义是任意数字出现 3 次，后跟句点，接着是任意数字出现 3 次，后跟句点。

```
$grep '[0-9 ]{3 } \.[0-0{3}\]' ipfile
```

补充说明，`grep` 家族还包括 `fgrep` 和 `egrep`。`fgrep` 是 `fix grep`，允许查找字符串而不是一个模式；`egrep` 是扩展 `grep`，支持基本及扩展的正则表达式，但不支持 `\q` 模式范围的应用及与之相对应的一些更加规范的模式。

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.[options]主要参数

`bs=字节`：强迫 `ibs=<字节>` 及 `obs=<字节>`。

`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>`。

`of=文件`：将数据写入 `<文件>`，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.应用实例

`dd` 命令常常用来制作 Linux 启动盘。先找一个可引导内核，令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
```

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

`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，系统从这里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` 是 0 的话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`: 不在其它文件系统（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 grpoupname`: 搜索所有组为 `grpoupname` 的文件。

—`user` 用户名: 搜索所有文件属主为用户名（ID 或名称）的文件。

—`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.应用技巧

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

有时我们知道只某个文件包含有 **abvd** 这 4 个字, 那么要查找系统中所有包含有这 4 个字符的文件可以输入下面命令:

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

输入这个命令以后, Linux 系统会将在/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 **abvdmysql** 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, 我们想在 **/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。

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

```
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.[options]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。

4.应用实例

(1) 将 `/usr/cbu` 中的所有文件移到当前目录（用“.”表示）中：

```
$ mv /usr/cbu/ * .
```

(2) 将文件 `cjh.txt` 重命名为 `wjz.txt`：

```
$ mv cjh.txt wjz.txt
```

```
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.options 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: 块以指定<大小>的字节为单位。

—B, —ignore—backups: 不列出任何以“~”字符结束的项目。

—f: 不进行排序，—aU 参数生效，—lst 参数失效。

—F, —classify: 加上文件类型的指示符号 (*!=@| 其中一个)。

- g: like —l, but do not list owner。
- G, —no—group: inhibit display of group information。
- i, —inode: 列出每个文件的 inode 号。
- l, —ignore=样式: 不印出任何符合 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, 而并非符号链接本身的信息。
- m: 所有项目以逗号分隔, 并填满整行行宽。
- 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 (/=@| 其中一个)。
- Q, —quote—name: 将项目名称括上双引号。
- 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.应用举例

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, 灰色表示是其它格式文件。ls 最常使用的是 ls-l, 见图 1 所示。

图 1 使用 ls-l 命令

文件类型开头是由 10 个字符构成的字符串。其中第一个字符表示文件类型, 它可以是下述类型之一: — (普通文件)、d (目录)、l (符号链接)、b (块设备文件)、c (字符设备文件)。后面的 9 个字符表示文件的访问权限, 分为 3 组, 每组 3 位。第一组表示文件属主的权限, 第二组表示同组用户的权限, 第三组表示其他用户的权限。每一组的三个字符分别表示对文件的读 (r)、写 (w) 和执行权限 (x)。对于目录, 表示进入权限。s 表示当文件被执行时, 把该文件的 UID 或 GID 赋予执行进程的 UID (用户 ID)

或 GID（组 ID）。t 表示设置标志位（留在内存，不被换出）。如果该文件是目录，那么在该目录中的文件只能被超级用户、目录拥有者或文件属主删除。如果它是可执行文件，那么在该文件执行后，指向其正文段的指针仍留在内存。这样再次执行它时，系统就能更快地装入该文件。接着显示的是文件大小、生成时间、文件或命令名称。

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.[options]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。

-l: 忽略大小写的变化。

-n --rcs: 输出 RCS 格式。

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.[options]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。

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.[options]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.应用举例

(1) 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
```

(2) 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
```

(3) cat 还有一个重要的功能就是可以对行进行编号，见图 2 所示。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。

图 2 使用 cat 命令/etc/named.conf 文件进行编号

对行进行编号功能有-b（只能对非空白行进行编号）和-n（可以对所有行进行编号）两个参数：

```
$ cat -b /etc/named.conf
```

```
1n
```

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]`

3.参数

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

链接有两种,一种被称为硬链接(Hard Link),另一种被称为符号链接(Symbolic Link)。默认情况下, `ln` 命令产生硬链接。

硬链接指通过索引节点来进行的连接。在 Linux 的文件系统中, 保存在磁盘分区中的文件不管是什么类型都给它分配一个编号, 称为索引节点号(Inode Index)。在 Linux 中, 多个文件名指向同一索引节点是存在的。一般这种连接就是硬连接。硬连接的作用是允许一个文件拥有多个有效路径名, 这样用户就可以建立硬连接到重要文件, 以防止“误删”的功能。其原因如上所述, 因为对应该目录的索引节点有一个以上的连接。只删除一个连接并不影响索引节点本身和其它的连接, 只有当最后一个连接被删除后, 文件的数据块及目录的连接才会被释放。也就是说, 文件才会被真正删除。

与硬连接相对应, Linux 系统中还存在另一种连接, 称为符号连接 (Symbolic Link), 也叫软连接。软链接文件有点类似于 Windows 的快捷方式。它实际上是特殊文件的一种。在符号连接中, 文件实际上是一个文本文件, 其中包含的有另一文件的位置信息。

动手练习

上面我们介绍了 Linux 文件处理命令, 下面介绍几个实例, 大家可以动手练习一下刚才讲过的命令。

1.利用符号链接快速访问关键目录

符号链接是一个非常实用的功能。假设有一些目录或文件需要频繁使用, 但由于 Linux 的文件和目录结构等原因, 这个文件或目录在很深的子目录中。比如, Apache Web 服务器文档位于系统的 `/usr/local/httpd/htdocs` 中, 并且不想每次都要从主目录进入这样一个长的路径之中(实际上, 这个路径也非常不容易记忆)。

为了解决这个问题, 可以在主目录中创建一个符号链接, 这样在需要进入该目录时, 只需进入这个链接即可。

为了能方便地进入 Web 服务器(`/usr/local/httpd/htdocs`)文档所在的目录, 在主目录下可以使用以下命令:

```
$ ln -s /usr/local/httpd/htdocs gg
```

这样每次进入 **gg** 目录就可访问 **Web** 服务器的文档，以后如果不再访问 **Web** 服务器的文档时，删除 **gg** 即可，而真正的 **Web** 服务器的文档并没有删除。

2.使用 **dd** 命令将 **init.rd** 格式的 **root.ram** 内容导入内存

```
dd if=/dev/fd0 of=floppy.fd
```

```
dd if=root.ram of=/dev/ram0 #
```

3.**grep** 命令系统调用

grep 是 Linux/Unix 中使用最广泛的命令之一，许多 Linux 系统内部都可以调用它。

(1)如果要查询目录列表中的目录，方法如下：

```
$ ls -l | grep '^d'
```

(2)如果在一个目录中查询不包含目录的所有文件，方法如下：

```
$ ls -l | grep '^[^d]'
```

(3)用 **find** 命令调用 **grep**，如所有 C 源代码中的“Chinput”，方法如下：

```
$find /ZhXwin -name *.c -exec grep -q -s Chinput {} \;-print
```

Linux 必学的 60 个命令(3)-系统管理命令

Linux 必学的系统管理命令

对于 Linux 系统来说，无论是中央处理器、内存、磁盘驱动器、键盘、鼠标，还是用户等都是文件，Linux 系统管理的命令是它正常运行的核心。熟悉了 Linux 常用的文件处理命令以后，这一讲介绍对系统和用户进行管理的命令。

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.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。

5.应用实例

Linux 支持的文件系统非常多, 包括 **JFS**、**ReiserFS**、**ext**、**ext2**、**ext3**、**ISO9660**、**XFS**、**Minx**、**vfat**、**MSDOS** 等。使用 **df -T** 命令查看磁盘空间时还可以得到文件系统的信息:

```
# df -T
```

```
文件系统 类型 容量 已用 可用 已用% 挂载点
/dev/hda7 reiserfs 5.2G 1.6G 3.7G 30% /
/dev/hda1 vfat 2.4G 1.6G 827M 66% /windows/C
/dev/hda5 vfat 3.0G 1.7G 1.3G 57% /windows/D
/dev/hda9 vfat 3.0G 2.4G 566M 82% /windows/E
/dev/hda10 NTFS 3.2G 573M 2.6G 18% /windows/F
```

```
/dev/hda11 vfat 1.6G 1.5G 23M 99% /windows/G
```

从上面除了可以看到磁盘空间的容量、使用情况外，分区的文件系统类型、挂载点等信息也一览无遗。

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。

4.说明

top 命令是 Linux 系统管理的一个主要命令，通过它可以获得许多信息。这里我们结合图 1 来说明它给出的信息。

图 1 top 命令的显示

在图 1 中，第一行表示的项目依次为当前时间、系统启动时间、当前系统登录用户数目、平均负载。第二行显示的是所有启动的进程、目前运行的、挂起 (Sleeping)的和无用 (Zombie)的进程。第三行显示的是目前 CPU 的使用情况，包括系统占用的比例、用户使用比例、闲置(Idle)比例。第四行显示物理内存的使用情况，包括总的可以使用的内存、已用内存、空闲内存、缓冲区占用的内存。第五行显示交换分区使用情况，包括总的交换分区、使用的、空闲的和用于高速缓存的大小。第六行显示的项目最多，下面列出了详细解释。

PID (Process ID) : 进程标示号。

USER: 进程所有者的用户名。

PR: 进程的优先级别。

NI: 进程的优先级别数值。

VIRT: 进程占用的虚拟内存值。

RES: 进程占用的物理内存值。

SHR: 进程使用的共享内存值。

S: 进程的状态，其中 **S** 表示休眠，**R** 表示正在运行，**Z** 表示僵死状态，**N** 表示该进程优先值是负数。

%CPU: 该进程占用的 CPU 使用率。

%MEM: 该进程占用的物理内存和总内存的百分比。

TIME+: 该进程启动后占用的总的 CPU 时间。

Command: 进程启动的启动命令名称，如果这一行显示不下，进程会有一个完整的命令行。

top 命令使用过程中，还可以使用一些交互的命令来完成其它参数的功能。这些命令是通过快捷键启动的。

<空格>: 立刻刷新。

P: 根据 CPU 使用大小进行排序。

T: 根据时间、累计时间排序。

q: 退出 **top** 命令。

m: 切换显示内存信息。

t: 切换显示进程和 CPU 状态信息。

c: 切换显示命令名称和完整命令行。

M: 根据使用内存大小进行排序。

W: 将当前设置写入 `~/toprc` 文件中。这是写 **top** 配置文件的推荐方法。

可以看到，**top** 命令是一个功能十分强大的监控系统的工具，对于系统管理员而言尤其重要。但是，它的缺点是会消耗很多系统资源。

5.应用实例

使用 **top** 命令可以监视指定用户，缺省情况是监视所有用户的进程。如果想查看指定用户的情况，在终端中按“U”键，然后输入用户名，系统就会切换为指定用户的进程运行界面，见图 2 所示。

图 2 使用 **top** 命令监视指定用户

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.应用实例

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。

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.应用实例

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。要查询自己的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。

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 x] [-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.应用实例

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

上面命令表示让系统在两天后的 17: 30 执行文件 data 中指明的作业。

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。

4.应用实例

(1) 使用 lp 命令打印多个文件

```
# lp 2 3 4
```

```
request id is 11 (3 file(s))
```

其中 2、3、4 分别是文件名；“request id is 11 (3 file(s))”表示这是第 11 个打印命令，依次打印这三个文件。

(2) 设定打印优先级别

```
# lp -d LaserJet -p 90 /etc/aliases
```

通过添加“-p 90”，规定了打印作业的优先级为 90。它将在优先级低于 90 的打印作业之前打印，包括没有设置优先级的作业，缺省优先级是 50

useradd

1.作用

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

2.格式

`useradd [-d home] [-s shell] [-c comment] [-m [-k template]] [-f inactive] [-e expire] [-p passwd] [-r] 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`: 指定用户登入时的起始目录。

`-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`: 自动建立用户的登入目录。

`-M`: 不要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。

`-r`: 建立系统账号。

`-s`: 指定用户登入后所使用的 `shell`。

`-u`: 指定用户 ID 号。

4.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。

5.应用实例

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 到 499 之间的值留给 bin、mail 这样的系统账号。

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，不必惟一。

-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.应用实例

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。

kill

1.作用

kill 命令用来中止一个进程。

2.格式

kill [-s signal | -p] [-a] pid ...

kill -l [signal]

3.参数

-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.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，系统上经常同时运行着多个进程。我们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，所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。

Linux 操作系统包括三种不同类型的进程，每种进程都有自己的特点和属性。交互进程是由一个 Shell 启动的进程。交互进程既可以在前台运行，也可以在后台运行。批处理进程和终端没有联系，是一个进程序列。监控进程（也称系统守护进程）是 Linux 系统启动时启动的进程，并在后台运行。例如，httpd 是著名的 Apache 服务器的监控进程。

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 中止某些进程来提高系统资源。在讲解安装和登陆命令时，曾提到系统多个虚拟控制台的作用是当一个程序出错造成系统死锁时，可以切换到其它虚拟控制台工作关闭这个程序。此时使用的命令就是 kill，因为 kill 是大多数 Shell 内部命令可以直接调用的。

5.应用实例

（1）强行中止（经常使用杀掉）一个进程标识号为 324 的进程：

```
#kill -9 324
```

（2）解除 Linux 系统的死锁

在 Linux 中有时会发生这样一种情况：一个程序崩溃，并且处于死锁的状态。此时一般不用重新启动计算机，只需要中止(或者说是关闭)这个有问题的程序即可。当 kill 处于 X-Window 界面时，主要的程序(除了崩溃的程序之外)一般都已经正常启动了。此时打开一个终端，在那里中止有问题的程序。比如，如果 Mozilla 浏览器程序出现了锁死的情况，可以使用 kill 命令来中止所有包含有 Mozilla 浏览器的程序。首先用 top 命令查处该程序的 PID，然后使用 kill 命令停止这个程序：

```
#kill -SIGKILL XXX
```

其中，XXX 是包含有 Mozilla 浏览器的程序的进程标识号。

（3）使用命令回收内存

我们知道内存对于系统是非常重要的，回收内存可以提高系统资源。kill 命令可以及时地中止一些“越轨”的程序或很长时间没有相应的程序。例如，使用 top 命令发现一个无用 (Zombie) 的进程，此时可以使用下面命令：

```
# kill -9 XXX
```

其中，XXX 是无用的进程标识号。

然后使用下面命令：

```
# free
```

此时会发现可用内存容量增加了。

(4) killall 命令

Linux 下还提供了一个 killall 命令，可以直接使用进程的名字而不是进程标识号，例如：

```
# killall -HUP inetd
```

crontab

1.作用

使用 crontab 命令可以修改 crontab 配置文件，然后该配置由 cron 公用程序在适当的时间执行，该命令使用权限是所有用户。

2.格式

```
crontab [ -u user ] 文件
```

```
crontab [ -u user ] { -l | -r | -e }
```

3.主要参数

—e: 执行文字编辑器来设定日程表，内定的文字编辑器是 vi。

—r: 删除目前的日程表。

—l: 列出目前的日程表。

crontab 文件的格式为“M H D m d cmd”。其中，M 代表分钟 (0~59)，H 代表小时 (0~23)，D 代表天 (1~31)，m 代表月 (1~12)，d 代表一星期内的天 (0~6, 0 为星期天)。cmd 表示要运行的程序，它被送入 sh 执行，这个 Shell 只有 USER、HOME、SHELL 三个环境变量。

4.说明

和 `at` 命令相比，`crontab` 命令适合完成固定周期的任务。

5.应用实例

设置一个定时、定期的系统提示：

```
[cao @www cao]#crontab -e
```

此时系统会打开一个 `vi` 编辑器。

如果输入以下内容：`35 17 * * 5 wall "Tomorrow is Saturday I will go CS"`，然后存盘退出。这时在 `/var/spool/cron/` 目录下会生产一个 `cao` 的文件，内容如下：

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
```

```
# (/tmp/crontab.2707 installed on Thu Jan 1 22:01:51 2004)
```

```
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
```

```
35 17 * * 5 wall "Tomorrow is Saturday I will play CS "
```

这样每个星期五 17: 35 系统就会弹出一个终端，提醒星期六可以打打 CS 了！显示结果见图 3 所示。

图 3 一个定时、定期的系统提示

动手练习

1.联合使用 `kill` 和 `top` 命令观察系统性能的变化

首先启动一个终端运行 `top` 命令，然后再启动一个终端使用 `kill` 命令，见图 4 所示。

图 4 观察 `kill` 命令对 `top` 终端的影响

这时利用上面介绍的 `kill` 命令来中止一些程序：

```
# kill SIGKILL XXX
```

然后再看 `top` 命令终端的变化，包括内存容量、CPU 使用率、系统负载等。注意，有些进程是不能中止的，不过学习 Linux 命令时可以试试，看看系统有什么反应。

2.使用 `at` 和 `halt` 命令定时关机

首先设定关机时间是 17:35，输入下面代码：

```
# at 17:35
```

```
warning: commands will be executed using (in order) a) $SHELL b) login shell c)
```

```
/bin/sh
```

```
at>halt -i -p
```

```
at> <EOT>
```

```
job 6 at 2004-01-01 17:35
```

此时实际上就已经进入 Linux 系统的 Shell，并且编写一个最简单程序：halt -i -p。

上面 Shell 中的文本结束符号表示按“Ctrl+D”组合键关闭命令，提交任务退出 Shell。“Job 6 at 2004-01-01 17:35”表示系统接受第 6 个 at 命令，在“2004-01-01 17:35”时执行命令：先把所有网络相关的装置停止，关闭系统后关闭电源。

3.用 crontab 命令实现每天定时的病毒扫描

前面已经介绍了一个简单的 crontab 命令操作，这里看一些更重要的操作。

(1) 建立一个文件，文件名称自己设定，假设为 caoproject:

```
# crontab -e
```

(2) 文件内容如下:

```
05 09 * * * antivir
```

用 vi 编辑后存盘退出。antivir 是一个查杀 Linux 病毒的软件，当然需要时先安装在系统中。

(3) 使用 crontab 命令添加到任务列表中:

```
# crontab caoproject
```

这样系统内所有用户在每天的 9 点 05 分会自动进行病毒扫描。

4.用 kill 使修改的配置文件马上生效

Windows 用户一般都知道，重要配置文件修改后往往都要重新启动计算机才能使修改生效。而 Linux 由于采用了模块化设计，可以自己根据需要实时设定服务。这里以网络服务 inetd 为例介绍一些操作技巧。

inetd 是一个监听守护进程，监听与提供互联网服务进程（如 rlogin、telnet、ftp、rsh）进行连接的要求，并扩展所需的服务进程。默认情况下，inetd 监听的这些 daemon 均列于/etc/inetd.conf 文件中。编辑/etc/inetd.conf 文件，可以改变 inetd 启动服务器守护进程的选项，然后驱使 inetd 以 SIGHUP（signal 1）向当前的 inetd 进程发送信号，使 inetd 重读该文件。这一过程由 kill 命令来实现。

用 vi 或其它编辑器修改 inetd.conf 后，首先使用下面命令:

```
# ps -ef |grep inetd
```

上面代码表明查询 `inetd.conf` 的进程号(PID)，这里假设是 1426，然后使用下面命令：

```
# kill -1426 inetd
```

这样配置文件就生效了。

这一讲介绍的系统管理命令都是比较重要的，特别是 `crontab` 命令和 `quota` 命令使用起来会有一定难度，需要多做一些练习。另外，使用 `kill` 命令要注意“-9”这个参数，练习时最好不要运行一些重要的程序。

Linux 必学的 60 个命令(4)-网络操作命令

Linux 必学的 60 个命令：网络操作命令

因为 Linux 系统是在 Internet 上起源和发展的，它与生俱来拥有强大的网络功能和丰富的网络应用软件，尤其是 TCP/IP 网络协议的实现尤为成熟。Linux 的网络命令比较多，其中一些命令像 `ping`、`ftp`、`telnet`、`route`、`netstat` 等在其它操作系统上也能看到，但也有一些 Unix/Linux 系统独有的命令，如 `ifconfig`、`finger`、`mail` 等。Linux 网络操作命令的一个特点是，命令参数选项和功能很多，一个命令往往还可以实现其它命令的功能。

`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。

`pointopoint`：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.应用说明

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。要赋给 **eth0** 接口 IP 地址 **207.164.186.2**，并且马上激活它，使用下面命令：

```
#ifconfig eth0 210.34.6.89 netmask 255.255.255.128 broadcast 210.34.6.127
```

该命令的作用是设置网卡 **eth0** 的 IP 地址、网络掩码和网络的本地广播地址。若运行不带任何参数的 **ifconfig** 命令，这个命令将显示机器所有激活接口的信息。带有“-a”参数的命令则显示所有接口的信息，包括没有激活的接口。注意，用 **ifconfig** 命令配置的网络设备参数，机器重新启动以后将会丢失。

如果要暂停某个网络接口的工作，可以使用 **down** 参数：

```
#ifconfig eth0 down
```

ip

1.作用

ip 是 **iproute2** 软件包里面的一个强大的网络配置工具，它能够替代一些传统的网络管理工具，例如 **ifconfig**、**route** 等，使用权限为超级用户。几乎所有的 Linux 发行版本都支持该命令。

2.格式

```
ip [OPTIONS] OBJECT [COMMAND [ARGUMENTS]]
```

3.主要参数

OPTIONS 是修改 **ip** 行为或改变其输出的选项。所有的选项都是以-字符开头，分为长、短两种形式。目前，**ip** 支持如表 1 所示选项。

OBJECT 是要管理者获取信息的对象。目前 **ip** 认识的对象见表 2 所示。

表 1 **ip** 支持的选项

-V,-Version 打印 **ip** 的版本并退出。

-s,-stats,-statistics 输出更为详尽的信息。如果这个选项出现两次或多次，则输出的信息将更为详尽。

-f,-family 这个选项后面接协议种类，包括 **inet**、**inet6** 或 **link**，强调使用的协议种类。如果没有足够的信息告诉 **ip** 使用的协议种类，**ip** 就会使用默认值 **inet** 或 **any**。**link** 比较特殊，它表示不涉及任何网络协议。

-4 是 **-family inet** 的简写。

-6 是 **-family inet6** 的简写。

-0 是 **-family link** 的简写。

-o,-oneline 对每行记录都使用单行输出，回行用字符代替。如果需要使用 **wc**、**grep** 等工具处理 **ip** 的输出，则会用到这个选项。

-r,-resolve 查询域名解析系统，用获得的主机名代替主机 IP 地址

COMMAND 设置针对指定对象执行的操作，它和对象的类型有关。一般情况下，**ip** 支持对象的增加(**add**)、删除(**delete**)和展示(**show** 或 **list**)。有些对象不支持这些操作，或者有其它的一些命令。对于所有的对象，用户可以使用 **help** 命令获得帮助。这个命令会列出这个对象支持的命令和参数的语法。如果没有指定对象的操作命令，**ip** 会使用默认的命令。一般情况下，默认命令是 **list**，如果对象不能列出，就会执行 **help** 命令。

ARGUMENTS 是命令的一些参数，它们依赖于对象和命令。**ip** 支持两种类型的参数：**flag** 和 **parameter**。**flag** 由一个关键词组成；**parameter** 由一个关键词加一个数值组成。为了方便，每个命令都有一个可以忽略的默认参数。例如，参数 **dev** 是 **ip link** 命令的默认参数，因此 **ip link ls eth0** 等于 **ip link ls dev eth0**。我们将在后面的详细介绍每个命令的使用，命令的默认参数将使用 **default** 标出。

4.应用实例

添加 IP 地址 192.168.2.2/24 到 eth0 网卡上：

```
#ip addr add 192.168.1.1/24 dev eth0
```

丢弃源地址属于 192.168.2.0/24 网络的所有数据报：

```
#ip rule add from 192.168.2.0/24 prio 32777 reject
```

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 [-dfnqrRv][-c][-i][-l][-p][-s][-t] 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。
- l: 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。

ping 命令是使用最多的网络指令，通常我们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，我们可以浏览器查看一个网页，但是却无法 ping 通，这是因为一些网站处于安全考虑安装了防火墙。另外，也可以在自己计算机上试一试，通过下面的方法使系统对 ping 没有反应：

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

netstat

1.作用

检查整个 Linux 网络状态。

2.格式

```
netstat [-acCeFghilMnNoprstuvVwx][[-A][--ip]
```

3.主要参数

- a--all: 显示所有连线中的 Socket。
- A: 列出该网络类型连线中的 IP 相关地址和网络类型。
- c--continuous: 持续列出网络状态。
- C--cache: 显示路由器配置的快取信息。
- e--extend: 显示网络其它相关信息。
- F--fib: 显示 FIB。
- g--groups: 显示多重广播功能群组组员名单。
- h--help: 在线帮助。
- i--interfaces: 显示网络界面信息表单。
- l--listening: 显示监控中的服务器的 Socket。
- M--masquerade: 显示伪装的网络连线。
- n--numeric: 直接使用 IP 地址，而不通过域名服务器。
- N--netlink--symbolic: 显示网络硬件外围设备的符号连接名称。
- o--timers: 显示计时器。
- p--programs: 显示正在使用 Socket 的程序识别码和程序名称。
- r--route: 显示 Routing Table。
- s--statistic: 显示网络工作信息统计表。
- t--tcp: 显示 TCP 传输协议的连线状况。
- u--udp: 显示 UDP 传输协议的连线状况。
- v--verbose: 显示指令执行过程。
- V--version: 显示版本信息。
- w--raw: 显示 RAW 传输协议的连线状况。
- x--unix: 和指定“-A unix”参数相同。
- ip--inet: 和指定“-A inet”参数相同。

4.应用实例

netstat 主要用于 Linux 察看自身的网络状况，如开启的端口、在为哪些用户服务，以及服务的状态等。此外，它还显示系统路由表、网络接口状态等。可以说，它是一个综合性的网络状态的察看工具。在默认情况下，**netstat** 只显示已建立连接的端口。如果要显示处于监听状态的所有端口，使用 **-a** 参数即可：

```
#netstat -a
```

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

```
tcp 0 0 *:32768 *: LISTEN
```

```
tcp 0 0 *:32769 *: LISTEN
```

```
tcp 0 0 *:nfs *: LISTEN
```

```
tcp 0 0 *:32770 *: LISTEN
```

```
tcp 0 0 *:868 *: LISTEN
```

```
tcp 0 0 *:617 *: LISTEN
```

```
tcp 0 0 *:mysql *: LISTEN
```

```
tcp 0 0 *:netbios-ssn *: LISTEN
```

```
tcp 0 0 *:sunrpc *: LISTEN
```

```
tcp 0 0 *:10000 *: LISTEN
```

```
tcp 0 0 *:http *: LISTEN
```

```
.....
```

上面显示出，这台主机同时提供 HTTP、FTP、NFS、MySQL 等服务。

telnet

1.作用

telnet 表示开启终端机阶段作业，并登入远端主机。**telnet** 是一个 Linux 命令，同时也是 一个协议（远程登陆协议）。

2.格式

```
telnet [-8acdEfFKLrx][-b][-e][-k][-l][-n][-S][-X][主机名称 IP 地址<通信端口>]
```

3.主要参数

- 8: 允许使用 8 位字符资料，包括输入与输出。
- a: 尝试自动登入远端系统。
- b: 使用别名指定远端主机名称。
- c: 不读取用户专属目录里的.telnetrc 文件。
- d: 启动排错模式。
- e: 设置脱离字符。
- E: 滤除脱离字符。
- f: 此参数的效果和指定“-F”参数相同。
- F: 使用 Kerberos V5 认证时，加上此参数可把本地主机的认证数据上传到远端主机。
- k: 使用 Kerberos 认证时，加上此参数让远端主机采用指定的领域名，而非该主机的域名。
- K: 不自动登入远端主机。
- l: 指定要登入远端主机的用户名称。
- L: 允许输出 8 位字符资料。
- n: 指定文件记录相关信息。
- r: 使用类似 rlogin 指令的用户界面。
- S: 服务类型，设置 telnet 连线所需的 IP TOS 信息。
- x: 假设主机有支持数据加密的功能，就使用它。
- X: 关闭指定的认证形态。

4.应用说明

用户使用 telnet 命令可以进行远程登录，并在远程计算机之间进行通信。用户通过网络在远程计算机上登录，就像登录到本地机上执行命令一样。为了通过 telnet 登录到远程计算机上，必须知道远程机上的合法用户名和口令。虽然有些系统确实为远程用户提供登录功能，但出于对安全的考虑，要限制来宾的操作权限，因此，这种情况下能使用的功能是很少的。