

Linux基本命令

目录

- 1 标题1
- 2 标题2
- 3 标题3
- 4 标题4
- 5 标题5

2.1 文件和目录操作命令

- **2.1.1 pwd、cd**

- 1. pwd——显示(打印)用户当前所处的目录
- 这是再常用不过的命令了，如果不知道自己当前所处的目录，就必须使用它。这个命令和DOS下的不带任何参数的cd命令的作用是一样的。其用法如下：

- [test @redflag test]\$pwd
- /home/test
- 说明当前目录是/home/test。

- 2. cd目录名——改变当前所处的目录或处理绝对目录和相对目录
- 如果用户当前处于/bin目录，想进入/etc目录，可以键入：
- `[test @redflag /bin]$cd /etc`

- **2.1.2 ls、tree**
- 1. ls [参数] 路径或文件名——列出文件或子目录的信息
- 参数选项:
- -a: 显示所有的文件, 包括以“.”开头的文件(即隐含文件)。
- -l: 以长格式显示文件或子目录的信息。
- -i: 显示每个文件的索引(节点)号。

- 执行命令[test @redflag test]\$ls -a 显示当前目录下的所有文件，输出：
- bak chap1.txt Desktop txt
- Linux系统用颜色来区分文件类别。缺省时，蓝色代表目录，绿色代表可执行文件，红色代表压缩文件，浅蓝色代表链接文件，灰色代表其他文件。
-

- 2. tree 目录名——以树的形式显示指定目录下的内容
- `[test @redflag test]$tree`
- 这是不带任何参数的tree命令，以树的形式显示当前目录下的文件和子目录，会递归到各子目录。例如：
- `[test @redflag test]$tree /etc/rc.d`
- 以树的形式显示目录/etc/rc.d下的文件和子目录。

- **2.1.3 mkdir、rmdir**
- 1. mkdir [参数] 目录名——建立目录
- 目录可以是绝对路径，也可以是相对路径。
- 参数选项：
- -p: 建立目录时，如果父目录不存在，则此时可以与子目录一起建立。

- 例如：
- **[test @redflag test]\$mkdir dir1**
- 在当前目录下建立dir1目录。
- **[test @redflag test]\$mkdir -p dir2/bak**
- 在dir2目录下建立bak目录，如果dir2目录不存在，那么同时建立dir2目录。

- **2. rmdir [参数] 目录名——删除目录**
- 目录同样可以是绝对路径，也可以是相对路径。
- 参数选项：
- **-p:** 一起删除父目录时，父目录下应无其他目录。
- 例如：
- **[root @redflag /root]#rmdir test**

- 删除当前目录下的test目录。删除目录时，被删除的目录下应无文件或目录存在。
- `[root @redflag /root]#rmdir -p longkey/test`
- 删除当前目录下的longkey/test目录。删除目录test时，如果父目录longkey下无其他内容，则一起删除longkey目录。

- **2.1.4 cp、rm、mv、ln**
- 1. cp [参数] 源文件 目标文件——拷贝文件或目录
- 相当于DOS下的copy命令。
- 参数选项：
- -f: 如果目标文件或目录存在，先删除它们再拷贝(即覆盖)，并且不提示用户。
- -i: 如果目标文件或目录存在，提示是否覆盖已有的文件。
- -R: 递归复制目录，即包含目录下的各级子目录。

- 2. rm [参数] 文件名或目录名——删除文件或目录
- 相当于DOS下的del命令。
- 参数选项：
 - -f: 删除文件或目录时不提示用户。
 - -i: 删除文件或目录时提示用户。
 - -R: 递归删除目录，即包含目录下的文件和各级子目录。

- 例如:
- `[test @redflag test]$rm *`
- 删除当前目录下的所有文件，但子目录和以“.”开头的文件(即隐含文件)不删除。
- `[test @redflag test]$rm -iR bak`
- 删除当前目录下的子目录**bak**，包含其下的所有文件和子目录，并且提示用户确认。

- 3. mv [参数] 源文件或目录 目标文件或目录——移动文件或目录
- 相当于DOS下的move命令。
- 参数选项：
 - -i: 如果目标文件或目录存在时，提示是否覆盖目标文件或目录。
 - -f: 不论目标文件或目录是否存在，均不提示是否覆盖目标文件或目录。
- 值得注意的是，mv可以用来更改文件名或目录名。

- 例如:
- `[test @redflag test]$mv 1.txt 2.txt`
- 这里移动文件时并不改变文件的目录，如果2.txt原来不存在，则实际上是1.txt更名为2.txt。
- `[test @redflag test]$mv ~/txtbak /bak`
- 把个人主目录下的目录txtbak移动到/bak目录下。

- 4. ln [参数] 源文件或目录 链接名——建立链接
- 参数选项:
- -s: 建立符号链接(即软链接), 不加该项时建立的是硬链接。
- 例如:
- [test @redflag test]\$ln telno.txt telno2.txt

- 给源文件telno.txt建立一个硬链接telno2.txt，这时telno2.txt可以看作是telno.txt的别名，它和telno.txt不分主次。telno.txt和telno2.txt实际上都指向硬盘上的相同位置，使用telno.txt作为文件名所做的更改，会在telno2.txt得到反映。硬链接有局限性，不能建立目录的硬链接。

- **2.1.5 chmod、chown、chgrp**
- 1. chmod 模式 文件或目录名——改变文件或目录的访问权限
- Linux系统是个多用户系统，应该能做到不同的用户能同时访问不同的文件，因此一定要有文件权限控制机制。Linux系统的权限控制机制和Windows的权限控制机制有着很大的差别。Linux的文件或目录都被一个用户拥有时，这个用户称为文件的拥有者(或所有者)，同时文件还被指定的用户组所拥有，这个用户组称为文件所属组。

- 要说明的是，一个用户可以是不同组的成员，这可以由管理员控制，我们将在用户管理这一章介绍如何控制的问题。文件的权限由权限标志来决定，权限标志决定了文件的拥有者、文件的所属组、其他用户对文件访问的能力。可以使用“ls -l”命令来显示权限标志。例如：
- [test @redflag test]\$ls -l
- -rw-rw-r-- 1 longkey root 16 20A 24 22:23 chap1.txt

- 本例中，文件chap1.txt的拥有者是longkey，所属组是root。这里我们特别关心的是输出行前面的第1~10个字符。第1个字符代表文件类别，第2~4个字符“rw-”是文件拥有者的权限，第5~7个字符“rw-”是文件所属组的权限，第8~10个字符“r--”是其他用户(即除了root用户和longkey用户组里的用户之外的用户)文件拥有者的权限。而权限均用三个字符表示，依次为读(r)、写(w)、执行(x)，如果某一位为“-”，则表示没有相应的权限，例如：“rw-”表示有读、写的权限，没有执行的权限。在本例中，文件拥有者longkey用户对文件有读、写的权限，所属组的所有用户对文件也有读、写的权限。

- 设定文件权限时，在模式中常用以下的字母代表用户或用户组：
- u——文件的拥有者；
- g——文件的所属组；
- o——其他用户；
- a——代表所有用户(即u+g+o)。
- 权限用以下字符表示：
- r——读权限；
- w——写权限；
- x——执行权限；
- 最后要指明是增加(+)还是减少(-)权限，或是绝对

- **【实例2.2】**
- `[root @redflag /root]#chmod o+w chap1.txt`
- chap1.txt的权限由原来的“rw-rw-r--”变为“rw-rw-rw-”，表示增加其他用户对文件的写权限。

- **【实例2.3】**
- `[root @redflag /root]#chmod u=rw,g=rw,o=r chap1.txt`
- chap1.txt 的权限变为 “`rw-rw-r---`”，不论原来的权限是什么，这表示拥有者对文件有读、写的权限，所属组的用户对文件也有读、写的权限，而其他用户只有读的权限。
-

- 我们在以上设置权限时，用字符表示权限和用户，实际上我们也经常使用八进制来表示。读、写、执行依次各自对应一个二进制位“???”，如果某位为“0”，则表示无权限；如果某位为“1”，则表示有权限。例如：文件权限为r---w---x时，用二进制表示为100010001，用八进制可以表示为421。例如：
- `[root @redflag /root]#chmod 664 chap1.txt`
- 等同于：
- `[root @redflag /root]#chmod u=rw,g=rw,o=r chap1.txt`

- 2. chown 用户名 文件或目录名——改变文件(或目录)的拥有者或所属组
- 例如:
- [root @redflag /root]#chown longkey chap1.txt
- 把文件chap1.txt的拥有者改为longkey用户。
- [root @redflag /root]#chown longkey:root chap1.txt
- 把文件的拥有者改为longkey用户，同时文件的所属组改为root组。

- 3. chgrp组 文件或目录——改变文件或目录的所属组
- chown可以同时改变文件拥有者和所属者，chgrp只具有改变所属组的功能。例如：
- [root @redflag /root]#chgrp root chap1.txt
- 文件chap1.txt的所属组设为root组。

- 2.1.6 find、grep
- 1. find 路径 匹配表达式——查找文件所在的目录
- 路径可以是多个路径，路径之间用空格隔开。查找时，会递归到子目录。
- 匹配表达式：
- -name: 指明要查找的文件名，支持通配符 “*” 和 “?”。
- -user username: 查找文件的拥有者为username的文件。
- -group grpname: 查找文件的所属组为grpname的文件。
-

- `-atime n`: 指明查找前n天访问过的文件(仅第n天这一天)。
- `-atime +n`: 指明查找前n天之前访问过的文件。
- `-atime -n`: 指明查找前n天之后访问过的文件。
- `-size n`: 指明查找文件大小为n块(block)的文件。
- `-print`: 搜索结果输出到标准设备。

- 例如:
- `[root @redflag /root]#find / -name passwd -print`
- 从根目录起查找名为passwd的文件，并把结果输出到标准设备。
- `[root @redflag /root]#find /home /etc -user longkey -print`
- 在目录/home和目录/etc中查找longkey用户所拥有的文件。

- 2. `grep [参数] 要查找的字符串 文件名` — 查找文件中包含有指定字符串的行
- 参数选项:
- `-num`: 输出匹配行前后各`num`行的内容。
- `-b`: 显示匹配查找条件的行距离文件开头有多少字节。
- `-c`: 显示文件中包含有指定字符串的行的个数，但不显示内容。
-

- 例如：
- `[root @redflag /root]#grep -2 Hello! chap.txt`
- 在文件 `chap1.txt` 中查找所有含有字符串“Hello!”的行，如果找到，显示该行及该行前后各2行的内容。文件名可以使用通配符*和?，如果要查找的字符串带空格，可以使用单引号或双引号括起来。

- 例如:
- `[root @redflag /root]#grep -2 Hello! chap.txt`
- 在文件chap1.txt中查找所有含有字符串“Hello!”的行，如果找到，显示该行及该行前后各2行的内容。文件名可以使用通配符*和?，如果要查找的字符串带空格，可以使用单引号或双引号括起来。

- 2. diff [参数] 源文件 目标文件——比较两个文件内容的不同
- 参数选项：
- -q: 仅报告是否相同，不报告详细的差异。
- -i: 忽略大小写的差异。
- diff命令的输出表示文件有哪些差别，如果要使文件相同，应该采取怎样的动作。由于其输出常常太复杂，以致于diff命令不太实用。我们不详细介绍输出的含义，有兴趣的读者可以用“diff --help”命令来获得详细的说明。

- Access: Thu Feb 27 23:18:00 2003
- Modify: Mon Feb 24 22:22:28 2003
- Change: Mon Feb 24 22:22:28 2003
- 显示文件passwd的被访问时间、修改时间、变更时间、文件大小、文件所有者、所属组、文件权限等项内容。

- 2. touch [参数] 文件或目录名——修改文件的存取和修改时间
- 参数选项：
- -d yyyymmdd：把文件的存取/修改时间改为 yyyymmdd。
- -a：只把文件的存取时间改为当前时间。
- -m：只把文件的修改时间改为当前时间。

- 例如：
- `[test @redflag test]$touch *`
- 把当前目录下的所有文件的存取和修改时间改为当前系统的时间。
- `[test @redflag test]$touch -d 20030224 chap1.txt`
- 把文件chap1.txt的存取和修改时间改为2003年2月24日。
- `[test @redflag test]$touch test.txt`

- 把test.txt的存取和修改时间改为当前系统的时间，如果test.txt文件不存在，则生成一个空文件(即0字节的文件)。
- touch还有另外一种形式：
- touch MMDDhhmm[YY] 文件名
- 例如：
- [test @redflag test]\$touch 0102120099 chap1.txt
- 把chap1.txt文件的存取和修改时间改为1999年01月02日12:00 。

2.2 显示命令

- **2.2.1 cat、more、less**
- 1. cat 文件名1 文件名2 ——显示文件的内容
- 相当于DOS下的type命令。
- 例如：
- [test @redflag test]\$cat chap1.txt chap2.txt
- 把文件chap1.txt、chap2.txt在标准的输出设备(通常是显示器)上显示出来。

- 2. `more` 文件名——逐页显示文件中的内容
- 如果文件太长，用`cat`命令只能看到文件的最后一页，而用`more`命令时可以一页一页地显示。执行`more`命令后，进入`more`状态，用【Enter】键可以向后移动一行；用【Space】键可以向后移动一页；用“q”键可以退出。在`more`状态下还有许多功能，可用`man more`命令获得。

- 3. less 文件名——逐页显示文件中的内容
- less实际上是more的改进版，其命令的直接含义是more的反义。less的功能比more更灵活。例如：用【Pgup】键可以向前移动一页，用【Pgdn】键可以向后移动一页，用向上光标键可以向前移动一行，用向下光标键可以向后移动一行。“q”键、【Enter】键、【Space】键的功能和more类似。

- **2.2.2 head、tail**
- 1. head [参数] 文件名——显示文件的前几行
- 参数选项：
 - -n num: 显示文件的前num行。
 - -c num: 显示文件的前num个字符。
- 缺省时，head显示文件的前10行。
- 例如：
 - [test @redflag test]\$head -n 20 chap1.txt
 - 显示文件chap1.txt的前20行。

- 2. tail [参数] 文件名——显示文件的末尾几行
- 参数选项：
 - -n num: 显示文件的末尾num行。
 - -c num: 显示文件的末尾num个字符。
- tail命令和head命令相反，它显示文件的末尾。缺省时，tail命令显示文件的末尾10行。
- 例如：
 - [test @redflag test]\$tail -n 20 chap1.txt
 - 显示文件chap1.txt的末尾20行。

- **2.2.3 sort、uniq**
- 1. sort [参数] 文件列表——将文件中的内容排序输出
- 参数选项：
- -r: 反向排序。
- -o filename: 把排序的结果输出到文件filename。
- 如果文件a.txt的内容为

- b

- c

- a

- d

- a

- 则执行**sort a.txt**命令后的显示结果为

- a

- a

- b

- c

- d

- 例如:
- `[test @redflag test]$sort -o c.txt a.txt`
- 把a.txt文件的内容排序, 并输出到文件c.txt。
- `[test @redflag test]$sort a.txt b.txt c.txt`
- 把文件a.txt、 b.txt、 c.txt的内容联合排序输出。

- 2. `uniq` 文件名——比较相邻的行，显示不重复的行
- 如**b.txt**文件的内容为
- b
- c
- c
- a
- d
- a

- 则执行`uniq b.tx`命令后的显示结果为
- b
- c
- a
- d
- a

- **2.2.4 file、locate、which**
- 1. file 文件名或目录——显示文件或目录的类型
- 例如:
- `[root @redflag /root]#file /etc/passwd`
- 则可能输出:
- `/etc/passwd:ASCII text`
- 说明passwd是个ASCII文本文件。

- 2. locate 字符串——查找绝对路径中包含指定字符串的文件
- 例如:
- [test @redflag test]\$locate chap1
- 则可能输出:
- /etc/longkey/chap1.txt
- /usr/share/doc/qt-devel-2.3.0/html/designer/chap10_1.html
- /usr/share/doc/qt-devel-2.3.0/html/designer/chap1_1.html
- /home/longkey/chap1.txt

- 3. which 命令——确定程序的具体位置
- 例如:
- [test @redflag test]\$which find
- 则输出find命令所处的位置:
- /usr/bin/find



2.3 进程管理和作业控制

- **Linux**是个多用户、多任务的操作系统。多用户系统是指多个用户可以同时使用同一计算机，而多任务是指系统可以同时执行多项任务。**Linux**操作系统将负责管理多个用户的请求和多个任务。用户运行一个程序，就会启动一个或多个进程。用户的感觉是一个人独占系统，实际上并非如此。

- 大多数系统只有一个CPU或有限的内存资源，一个CPU在一个时刻实际上只能运行一个进程，造成用户一个人独占系统的感觉是操作系统的功劳。操作系统控制着每一个运行着的程序(即进程)，给每一进程分配一个合适的时间片，大约有几十毫秒，每个进程轮流被CPU运行一段时间，然后被挂起，系统去处理另外一个进程，经过一段时间后这个进程又被运行。

- 所谓的程序是指程序员编写的计算机指令集，其实就是一个保存在磁盘上的文件。运行一个程序，就会在系统中创建一个或多个进程，进程可以看成是在计算机里正在运行的程序。**Linux**系统启动后，就已经创建了许多进程。
- 本节将介绍**Linux**这一多任务系统提供的关于进程管理的命令。

- **2.3.1 进程的启动**

- 进程的启动有两种方式：手工启动和调度启动。手工启动又分为前台启动和后台启动。前台启动是最常用的方式，用户直接运行一个程序或执行一个命令时就启动了前台进程。例如：用户执行“ls -l”命令就启动了一个新的前台进程，只不过这个进程可能很快就结束了。前台进程的一个特点是进程不结束，终端不出现“#”或“\$”提示符，所以用户不能再执行别的任务。后台进程的启动是用户在输入命令行后加上“&”字符，例如：

- `[root @redflag /root]#find / -name myfile -print > /root/test &`
- 这就启动了一个后台进程。后台进程常用于进程耗时长、用户不着急得到结果的情况。用户启动一个后台进程后，终端会出现“#”或“\$”提示符，而不必等待进程的结束，用户又可以接着执行别的任务。
- 至于调度进程，是指用户事先设定好(如在某个时间)，让系统自行启动进程的方法。有关调度进程的方法将在本节稍后介绍。

- **2.3.2 查看系统的进程**

- 要管理进程，首先要知道系统里有哪些进程存在及进程的状况如何。可以使用下面的命令：

- `ps [参数]`——查看系统的进程。
- 参数选项：
- `a`：显示当前控制终端的进程(包括其他用户的)。
- `u`：显示进程的用户名和启动时间等信息。
- `-w`：宽行输出，不截取输出中的命令行。
- `-l`：按长格式显示输出。
-

- x: 显示没有控制终端的进程。
- -e: 显示所有的进程。
- -t n: 显示第n个终端的进程。
- ps命令的输出，含义如下：
- USER: 启动进程的用户名。
- PID: 进程号。
- PPID: 父进程的进程号。
- TTY: 启动进程的终端号。

- **STAT:** 进程的状态，R表示进程正在运行，S表示进程在睡眠，T表示进程僵死或停止，D表示进程处于不能中断的睡眠(通常是输入输出)。
- **START:** 进程开始的时间。
- **TIME:** 进程已经运行的时间。
- **COMMAND/CMD:** 进程的命令名。
- **%CPU:** 进程占用CPU总时间的百分比。
- **%MEM:** 进程占用系统内存总量的百分比。
- **NI:** nice的优先级。
- **PRI:** 进程的优先级。

- 例如:
- [test @redflag test]\$ps au
- USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
- root 1110 0.0 0.4 2372 1236 pts/0 S 23:17 0:00 login -- test
- test 1111 0.0 0.5 2788 1360 pts/0 S 23:17 0:00 -bash
- root 1144 0.0 0.4 2792 1060 pts/0 S 23:17 0:00 su root
- root 1145 0.0 0.5 2792 1364 pts/0 S 23:17 0:00 bash
- root 1192 0.0 0.2 2624 760 pts/0 R 23:23

- 以上显示当前控制终端的进程。
- `[test @redflag test]$ps -elw`
- 以上以长格式显示所有的进程。
- **【技巧】** `ps`常和重定向、管道命令一起使用，用于查找出所需的进程，例如：
- `[test @redflag test]$ps -e u | grep test`
- 查找test用户启动的进程。
- `[test @redflag test]$ps -e | grep httpd`
- 查找httpd(Web服务守护进程)进程的信息，如进程号等。

- **2.3.3 进程的控制**

- **1. kill命令——给进程发送信号**

- 前台进程在运行时，可以用**【Ctrl-c】**来终止它。但后台进程无法用这种方法来终止，这时候可以使用kill命令向进程发送强制终止信号来达到目的。

- 例如:
- `[root @redflag /root]#kill -l`
- 显示kill命令所能够发送的信号种类，每个信号都有一个数值对应，例如：SIGKILL信号的值是9，而SIGTERM的值是15，SIGTERM信号是kill命令默认的信号。kill命令的格式为:
- `kill [参数] 进程1 进程2...`

- 参数选项:
- -s signal: signal是信号类别, 如: SIGKILL。
- 例如:
- [root @redflag /root]#ps
- PID TTY TIME CMD
- 835 tty1 00:00:00 login
- 843 tty1 00:00:00 bash
- 1212 tty1 00:00:00 ps
- [root @redflag /root]#kill -s SIGKILL 835
- 则系统退到登录界面, 以上命令也可以用以下命令代替:

- 2. killall -s signal 命令名——根据进程名来发送信号
- 参数选项：
- -s signal: signal是信号类别，如：SIGKILL。
- 用kill命令时要用ps命令查出进程号，这样不是很方便。killall可以根据进程名来发送信号。
- 例如：
- [root @redflag /root]#killall -9 vim
- 终止所有vi会话。

- 3. nice 命令——以指定的优先级运行程序
- Linux系统有两个与进程有关的优先级。用“ps -l”命令可以看到两个域：PRI和NI。PRI是进程实际的优先级，它是由操作系统动态计算的，这个优先级的计算机和NI值有关。NI值可以被用户更改，其值范围为-20~20。NI值越高，优先级越低。一般用户只能加大NI值(即降低优先级)，只有超级用户可以减小NI值(即提高优先级)。NI值被改变后，会影响PRI。优先级高的进程被优先运行，缺省时进程的NI为0。nice命令的用法如下：

- `nice -n 程序名` 以指定的优先级运行程序
- `n`: NI值, 正值代表NI值增加, 负值代表NI值减小。
- 例如:
- `[root @redflag /root]#nice --15 ps -l`
- 则输出:

- F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY
TIME CMD
- 100 S 0 1261 1260 0 70 0 - 593 wait4 pts/0
00:00:00 login
- 000 S 0 1295 1262 0 73 0 - 698 wait4 pts/0
00:00:00 su
- 100 S 0 1296 1295 0 76 0 - 698 wait4 pts/0
00:00:00 bash
- 100 R 0 1318 1296 0 78 -15 - 769 - pts/0
00:00:00 ps
- 可以看到ps命令以NI值为-15的优先级运行。

- 4. **renice** 命令——改变进程的优先级
- 运行中的进程的优先级可以被调整，注意只有**root**用户可以提高进程的优先级，一般用户只能降低优先级。**renice**命令就是用来改变进程的优先级的，其用法如下：
- **renice** n 进程号
- n为期望的进程NI值。

- [root @redflag /root]#ps -l
- F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY
TIME CMD
- 100 S 0 1261 1260 0 69 0 - 593 wait4 pts/0
00:00:00 login
- 000 S 0 1295 1262 0 69 0 - 698 wait4 pts/0
00:00:00 su
- 100 S 0 1296 1295 0 73 0 - 698 wait4 pts/0
00:00:00 bash
- 100 R 0 1373 1296 0 75 -15 - 769 - pts/0
00:00:00 ps
- [root @redflag /root]#renice -6 1261

- [root @redflag /root]#ps -l
- F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY
TIME CMD
- 100 S 0 1261 1260 0 69 -6 - 593 wait4 pts/0
00:00:00 login
- 000 S 0 1295 1262 0 69 0 - 698 wait4 pts/0
00:00:00 su
- 100 S 0 1296 1295 0 75 0 - 698 wait4 pts/0
00:00:00 bash
- 100 R 0 1380 1296 0 77 0 - 769 - pts/0
00:00:00 ps
- 检查进程号为1261的进程优先级是否发生了改变。

- 5. top 命令——实时监控进程程序
- 和ps命令不同，top命令可以实时监控进程状况。top屏幕自动每5秒刷新一次，也可以用“top -d 20”，使得top屏幕每20秒刷新一次。top的屏幕输出如下：
- 11:50pm up 1:06, 4 users, load average: 0.02, 0.01, 0.00
- 50 processes: 49 sleeping, 1 running, 0 zombie, 0 stopped
- CPU states: 0.0% user, 0.0% system, 0.0% nice, 99.9% idle

- Mem: 255212K av, 167244K used, 87968K free, 92K shrd, 42504K buff
- Swap: 771112K av, 0K used, 771112K free
31712K cached
- PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM
TIME COMMAND
- 1 root 8 0 536 536 464 S 0.0 0.2 0:04 init
- 2 root 9 0 0 0 0 SW 0.0 0.0 0:00 keventd
- 3 root 9 0 0 0 0 SW 0.0 0.0 0:00 kapm-
idled
- 4 root 19 19 0 0 0 SWN 0.0 0.0 0:00
ksoftirqd_CPU0

-
 579 snort 9 0 5452 5444 1176 S 0.0 2.1
 0:00 snort
- 606 named 9 0 3836 3836 2176 S 0.0 1.5
 0:00 named

- **top**屏幕前五行的含义如下：
- 第一行：正常运行时间行。显示系统当前时间，系统已经正常运行的时间，系统当前用户数，最近1分钟、5分钟、15分钟准备运行的进程平均数。
- 第二行：进程统计数。显示当前的进程总数，睡眠的进程数，正在运行的进程数，僵死的进程数，暂停的进程数。

- 第三行：CPU统计行。包括用户进程，系统进程，修改过NI值的进程，空闲进程各自使用CPU的百分比。
- 第四行：内存统计行。包括内存总量，已用内存，空闲内存，共享内存，缓冲区的内存总量。
- 第五行：交换区和缓冲区统计行。包括交换区总量，已使用的交换区，空闲交换区，高速缓冲区总量。
- 输出的其他行的含义和ps的输出类似，这里不再介绍。
- 在top屏幕下，用“q”键可以退出，用“h”键可以显示top下的帮助。

- 6. bg、jobs、fg
- 在手工启动前台进程时，如果进程没有执行完毕，则可以使用【Ctrl-z】键暂停进程的执行。例如：
- [root @redflag /root]#du -a / | sort -rn > /root/test.out 按【Ctrl-z】
- [1]+ Stopped du -a / | sort -rn > /root/test.out

- 表示du命令被暂停，这时候我们可以把暂停的进程放到后台继续运行，我们在前台还可以运行别的命令。bg命令用于把进程放到后台。
- [root @redflag /root]#bg du 或bg %1
- [1]+ du -a / | sort -rn > /root/test.out &
- 使用jobs命令可以看到在后台运行的进程。
- [root @redflag /root]#jobs
- [1]+ Running du -a / | sort -rn > /root/test.out

- 这说明了du程序在后台运行。在上面例子中，我们把被暂停的进程放到后台运行，如果我们想直接在后台运行它，在命令行后加“&”字符，不必用【Ctrl-z】键先把它暂停再用bg命令把它放到后台。例如：
- [root @redflag /root]#du -a / | sort -rn > /root/test.out &
- 用bg命令可以把进程放到后台，用fg命令则可以把在后台运行的进程放到前台。
- [root @redflag /root]#du -a / | sort -rn > /root/test.out &
- [root @redflag /root]#jobs

- 7. nohup命令
- 一般来说，当一个进程的父进程被终止时，该进程也会被终止。用“ps -l”命令可以看到进程的父进程。例如：

- F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY
TIME CMD
- 100 S 0 1552 1551 0 69 0 - 593 wait4 pts/0
00:00:00 login
- 000 S 0 1586 1553 0 69 0 - 698 wait4 pts/0
00:00:00 su
- 100 S 0 1587 1586 0 74 0 - 698 wait4 pts/0
00:00:00 bash
- 100 R 0 2642 1587 0 76 0 - 770 - pts/0
00:00:00 ps

- 其中，进程2642的父进程是1587，1587实际上是Shell进程。在Shell下启动的进程的父进程大多是Shell进程，而Shell进程的父进程又是login进程，所以当你logout时，由login进程衍生出的进程就会被终止。如果你希望logout后自己的进程仍然能在后台继续运行，可以使用nohup命令。nohup的用法如下：
- [root @redflag /root]#nohup 命令 &

- **【实例2.4】**
- `[root @redflag /root]#nohup du -a / | sort -rn > /root/test.out &`
- 命令`du-a/ | sort -rn > /root/test.out`不会在用户logout后终止。

- **2.3.4 作业控制**

- 有时我们需要把费时的工作放在深夜进行，这时候我们可以事先进行调度安排，即调度启动进程，系统会自动启动我们安排好的进程。

- 1. at、atq、atrm
- 我们可以使用at命令将要执行的命令安排成队列，例如：
- [root @redflag /root]#at 8:40
- at>du -a >test.out
- at>tree / >>test.out
-

- 按【Ctrl-d】退出。很明显，我们安排系统在8:40执行两个命令，注意这些命令只执行一次。使用at命令时，可以使用不同的时间格式，例如：20:40、8:00am、8:40am feb 23、10am+5days、12:30 pm tomorrow、midnight、noon等。如果执行如下命令：
- [root @redflag /root]#at -f mywork 9:45am+2days
- 则安排系统于两天后的上午9:45执行文件mywork里的作业。用at命令设定好作业后，atd守护进程将负责运行它们。我们可以使用atq命令查看已经安排好的作业。

- `[root @redflag /root]#atq`
- `14 2003-02-28 07:45 a root`
- `13 2003-02-27 08:40 a root`
- 输出行中依次是作业号、作业的启动时间、用户名，遗憾的是这里不能知道作业的内容。如果想知道作业的内容，请到`/var/spool/at`目录里去找(类似`a00011010a2548`这样的文件)。

- 2. crontab命令
- at命令用于安排运行一次的作业比较方便，但如果要重复运行程序，则使用crontab更为简捷。crontab的用法如下：
- crontab [参数] {-e | -l | -r }
- 参数选项：
- -u username: 用户username的作业，不指定时指当前用户。

- -e: 编辑用户cron作业。
- -l: 显示用户cron作业。
- -r: 删除用户cron作业。
- 例如:
- [root @redflag /root]#crontab -e
- 进入编辑cron作业状态，编辑器采用的是vi编辑器。关于vi的使用请参见第3章。一个cron作业的格式如下:
- 分 小时 日 月 星期 命令

- 如果用户不需要指定其中的几项，可以用“*”代替，例如：
- `0,10,20 * * * * updatedb`
- 每逢0、10、20分运行一次命令`updatedb`。
- `* * * * * date>>/root/crontest.out`
- 表示每分钟往`/root/crontest.out`文件添加一行关于时间的行。
- `[root @redflag /root]#crontab -l`

- 列出当前用户的cron作业:
- # DO NOT EDIT THIS FILE - edit the master and reinstall.
- # (/tmp/crontab.1106 installed on Thu Feb 27 23:17:23 2003)
- # (Cron version -- \$Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp \$)
- * * * * * date>>/root/crontest.out



2.4 文件压缩和备份

- **2.4.1 压缩和解压命令**

- 使用压缩文件不仅可以减小文件占用的磁盘空间，也可以减小文件在网络传输时所带来的传输流量。Linux的压缩和解压工具很多，下面我们介绍常用的几个工具。

- 1. `compress [参数] 文件名(压缩文件命令)` / `uncompress [参数] 文件名(文件解压命令)`
- 参数选项:
- `-v`: 显示被压缩的文件的压缩比或解压时的信息。
- 例如:
- `[root @redflag /root]#compress -v test`
- `test: -- replaced with test.Z Compress: 53.56%`

- 文件test被压缩成 “test.Z”，压缩比为： 53.56%。
- [root @redflag /root]#uncompress -v test 或
uncompress -v test.Z
- test.Z: -- replaced with test
- 解压文件是 “test.Z”。

- 2. `gzip -v 文件名(压缩文件) / gunzip -v 文件名(解压文件)`
- `gzip`、`gunzip`和`compress`、`uncompress`类似，不过压缩后的文件的文件名是以“`.gz`”结尾而已。

- 3. zip 压缩文件名(压缩文件) / unzip 被解压文件名(解压文件)
- zip生成的文件是以“.zip”为文件名的结尾，这种文件是我们在Windows等系统中最常见的压缩文件。zip命令的功能非常强大，可以创建自解压的文件、设置文件的保护口令等。常用man zip命令来获得zip命令的详细帮助。zip命令并不替换原文件。

- 例如:
- `[root @redflag /root]#zip test.zip test`
- `adding: test (deflated 66%)`
- 把文件test压缩到文件“test.zip”。
- `[root @redflag /root]#unzip test.zip`
- `Archive: test.zip`
- `inflating: test`
- 如果test已经存在，unzip命令会提示是否覆盖test文件。

- ## 2.4.2 文件备份

- 任何计算机系统都可能出现问题，从而导致数据的丢失，因此备份是系统维护中不可缺少的一个环节。备份应做到系统崩溃后能快速、简单、完整地恢复系统。针对要备份的内容，备份可以分为两类：系统数据备份和用户数据备份。系统数据是指Linux系统要正常运行所需的文件(如/bin和/boot目录)、系统配置(如/etc目录)等；而用户数据是指计算机用户创建的文件(如/home目录)等，相对于系统数据来说，用户数据的变化要频繁得多。

- 备份系统数据时，可以不备份不必要的数据 (如/`proc/core`)，它是当前物理内存的一个映像，并且系统备份常在系统有变化后进行，例如安装了新的补丁。

- 备份有两种策略：完全备份和增量备份。完全备份是把要备份的数据完完全全备份出来，一旦系统发生故障，可以使用备份的数据把数据恢复到备份前的状态；增量备份则是备份最后一次备份以后发生了变化的数据，因此被备份的数据量要少得多。实际工作中，完全备份和增量备份常常是结合起来使用的。

- 例如：一周(例如在星期一)进行一次完全备份，而每天进行一次增量备份，如果系统于星期四出现故障，恢复系统时先用完全备份恢复系统，然后再顺序用周二、周三的增量备份恢复系统。
Linux中的tar工具是最常用的备份和恢复工具，同时tar也是软件商发布补丁、新软件的常用工具，所以掌握tar的使用是非常重要的。我们把备份产生的文件称为文档文件(或文档)。格式如下：
- tar [参数] 文件或目录名

- 参数选项：
- **-c**: 创建一个新的文档。
- **-r**: 用于将文件附加到已存在的文档后面。
- **-u**: 仅仅添加比文档文件更新的文件，如原文档中不存在旧的文件，则追加它到文档中，如存在则更新它。
- **-x**: 从文档文件中恢复被备份的文件。

- **-t:** 用于列出一个文档文件中的被备份出的文件名。
- **-z:** 用zip命令压缩或用unzip解压。
- **-f:** 使用档案文件或设备，这个选项通常是必选的。
- **-v:** 列出处理过程中的详细信息。
- **-C directory:** 把当前目录切换到directory。

- 例如:
- [root @redflag /root]#tar -cvf longkey.tar /home/longkey
- 把/home/longkey目录下的文件和子目录(包括隐含文件和目录)备份到longkey.tar文档中。备份产生的文档文件的文件名最好用“.tar”结尾,以示区别。以上是一个完全备份。
- [root @redflag /root]#tar -uvf longkey.tar /home/longkey
- 把/home/longkey目录中比文档文件longkey.tar还新的文件添加到longkey.tar中。

- 以gzip压缩文件的形式把/home/longkey的内容备份到longkey.tar.gz中。备份产生的文档是压缩过的，这样可以减小文档文件的大小。注意，文件是以“.tar.gz”结尾的。
- [root @redflag /root]#tar -xzf longkey.tar.gz

- 从longkey.tar.gz文档中恢复数据。注意，恢复出来的数据是放在当前目录下的，而不是恢复到原来的目录。
- [root @redflag /root]#tar -xzf longkey.tar.gz -C /home
- 可以把文件恢复到指定的目录/home下。
- tar命令的参数还有很多，而且它常常和其他的命令如find等一起使用，以实现完全备份和增量备份，用户甚至还常写脚本来实现自己备份的策略。

2.5 网 络 命 令

- Linux系统也是一个网络操作系统，其网络功能也相当强大。目前Linux系统大多是被用来提供网络服务的，Linux系统可以提供各种各样的网络服务，例如：Web服务、FTP服务、DNS服务。这些内容将放在以后的章节介绍，我们先介绍基本的网络命令。

- **2.5.1 hostname、ping、host**

- 1. hostname [主机名]——显示或设置系统的主机名

- 例如：

- [root @redflag /root]#hostname

- redflag

- 表示本人的系统主机名是“redflag”。

- [root @redflag /root]#hostname
server.redflag.com

- 把主机名设置为“server.redflag.com”。

- 2. ping [参数] 主机名(或IP地址)——测试本主机和目标主机连通性
- 参数选项：
- -c count: 共发出count次信息，不加此项，则发无限次信息。
- -i interval: 两次信息之间的时间间隔为interval，不加此项，间隔为1秒。

- 例如:
- `[root @redflag /root]#ping -c 10 -i 0.5 192.168.0.1`
- `PING 192.168.0.1 (192.168.0.1) from 192.168.0.10 : 56(84) bytes of data.`
- `Warning: time of day goes back, taking countermeasures.`
- `64 bytes from 192.168.0.1: icmp_seq=0 ttl=128 time=648 usec`
- `64 bytes from 192.168.0.1: icmp_seq=1 ttl=128 time=578 usec`

- 64 bytes from 192.168.0.1: icmp_seq=8 ttl=128 time=569 usec
- 64 bytes from 192.168.0.1: icmp_seq=9 ttl=128 time=576 usec
-
- --- 192.168.0.1 ping statistics ---
- 10 packets transmitted, 10 packets received, 0% packet loss
- round-trip min/avg/max/mdev = 0.544/0.575/0.648/0.028 ms
- 以上命令共发出10次信息，信息之间的间隔为0.5秒。

- 3. host 主机名或IP地址——IP地址查找工具
- 例如:
- [root @redflag /root]#host www.sina.com
- www.sina.com. has address 66.77.9.79
- DNS 查 找 出 “ www.sina.com” 的 IP 地 址 为 “66.77.9.79”。
- [root @redflag /root]#host 202.96.134.133
- 133.134.96.202.in-addr.arpa. domain name
pointer ns.szptt.net.cn.

- 这是反向IP地址解析。
- `[root @redflag /root]# host mylinux 192.168.0.10`
- Using domain server:
- Name: 192.168.0.10
- Address: 192.168.0.10#53
- Aliases:
- mylinux.test.com. has address 192.168.0.1

- 以上指明了从DNS服务器192.168.0.10上查找主机mylinux的IP地址，如果没有指明DNS服务，则系统使用本机网络设置中设定的DNS服务器。有关IP地址的查找请进一步参见本书有关DNS的章节。

- 2.5.2 ifconfig
- ifconfig是用于配置网卡和显示网卡信息的工具。例如:
- [root @redflag /root]#ifconfig -a
- eth0 Link encap:Ethernet HWaddr 00:E0:4C:30:03:F7
- inet addr:192.168.0.10 Bcast:192.168.0.255
Mask:255.255.255.0
- UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
- RX packets:1257 errors:0 dropped:0 overruns:0
frame:0

- TX packets:558 errors:0 dropped:0 overruns:0 carrier:0
- collisions:0 txqueuelen:100
- RX bytes:138072 (134.8 Kb) TX bytes:48879 (47.7 Kb)
- Interrupt:5 Base address:0xc000
- lo Link encap:Local Loopback
- inet addr:127.0.0.1 Mask:255.0.0.0
- UP LOOPBACK RUNNING MTU:16436
Metric:1

- RX packets:14 errors:0 dropped:0 overruns:0 frame:0
- TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
- collisions:0 txqueuelen:0
- RX bytes:1046 (1.0 Kb) TX bytes:1046 (1.0 Kb)
- 显示本主机有一个网卡eth0。lo是本地环路(虚拟的)网卡，不是物理上实实在在的网卡。在以上的输出中有几个重要的信息：IP地址、网卡MAC地址、网卡的配置以及网卡的一些统计数(如接收和发送包的总量)。

- **2.5.3 traceroute 目标主机名或IP地址**
- 该命令显示本机到达目标主机的路由路径，例如：
- `[root @redflag /root]#traceroute www.sina.com`
- `traceroute to www.sina.com (66.77.9.79), 30 hops max, 38 byte packets`
- `1 * * *`
- `2 211.162.65.62 (211.162.65.62) 1.116 ms 1.010 ms 0.945 ms`
- `3 211.162.78.201 (211.162.78.201) 1.061 ms 1.053 ms 1.030 ms`

-
.....
- 18 svl-core-01.inet.qwest.net (205.171.8.242)
 1250.125 ms 1265.016 ms 1255.987 ms
- 19 svc-cntr-01.inet.qwest.net (205.171.14.2)
 1255.304 ms 1254.937 ms 1249.087 ms
- 20 msfc-07.jsv.qwest.net (66.77.106.98) 1247.149
 ms 1271.762 ms 1267.991 ms
- 21 66.77.9.79 (66.77.9.79) 1250.633 ms 1253.705
 ms 1260.267 ms

- **2.5.4 Telnet、FTP**

- 1. Telnet 主机名或IP地址——远程登录客户程序

- 例如：

- `[root @redflag /root]#telnet 192.168.0.200`

- 远程登录到服务器192.168.0.200。服务器192.168.0.200应开启Telnet服务，否则会连接失败。如果成功连接Telnet，程序会提示输入用户名和口令，登录成功后就可以远程管理或使用服务器。

- 2. FTP 主机名或IP地址——FTP客户程序
- 例如:
- `[root @redflag /root]#ftp 192.168.10`
- FTP到远程FTP服务器192.168.0.10，同样服务器192.168.0.10要开启FTP服务。连接成功后，FTP程序会提示输入用户名和口令。如果连接成功，将得到“ftp>”提示符。现在可以自由使用FTP提供的命令，可以用help命令或“?”取得可供使用的命令清单，也可以在help命令后面指定具体的命令名称，获得这条命令的说明。

- 最常用的命令有：
- `ls`: 列出远程机的当前目录。
- `cd`: 在远程机上改变工作目录。
- `lcd`: 在本地机上改变工作目录。
- `ascii`: 设置文件传输方式为ASCII模式。
- `binary`: 设置文件传输方式为二进制模式。
- `close`: 终止当前的FTP会话。
- `hash`: 每次传输完数据缓冲区中的数据后就显示一个“#”号。
- `get(mget)`: 从远程机传送指定文件到本地机。

- `put(mput)`: 从本地机传送指定文件到远程机。
- `open`: 连接远程FTP站点。
- `quit`: 断开与远程机的连接并退出FTP。
- `?`: 显示本地帮助信息。
- `!`: 转到Shell中。

- 下面简单将FTP常用命令作以介绍。
- (1) 启动FTP会话。open命令用于打开一个与远程主机的会话。该命令的一般格式是：
 - open 主机名/IP
- (2) 终止FTP会话。close、disconnect、quit和bye命令用于终止与远程机的会话。close和disconnect命令用于关闭与远程机的连接，但是仍没有退出FTP程序。quit和bye命令都用于关闭用户与远程机的连接，然后退出用户机上的FTP程序。

- (3) 改变目录。“cd [目录]”命令用于在FTP会话期间改变远程机上的目录，lcd命令可改变本地目录，使用户能指定查找或放置本地文件的位置。
- (4) 远程目录列表。ls命令列出远程目录的内容，就像使用一个交互Shell中的ls命令一样。ls命令的一般格式是：
 - ls [目录]
 - 如果指定了目录作为参数，那么ls就列出该目录的内容。

- (5) 从远程系统下载文件。`get`和`mget`命令用于从远程机上获取文件，`get`命令的一般格式为：
 - `get` 源文件名 目标文件名
 - 源文件名是要下载的文件名，目标文件名是文件下载后在本地机上保存时的文件名。如果不给出目标文件名，那么就使用源文件的名字。`mget`命令一次可获取多个远程文件。`mget`命令的一般格式为：
 - `mget` 文件名列表
 - 使用用空格分隔的或带通配符的文件名列表来指定要获取的文件。

- (6) 向远程系统上传文件。put和mput命令用于向远程机发送文件，put命令的一般格式为：
 - put 文件名
 - mput命令一次发送多个本地文件，mput命令的一般格式为：
 - mput 文件名列表
 - 使用用空格分隔的或带通配符的文件名列表来指定要发送的文件。

- (7) 改变文件传输模式。默认情况下，FTP按ASCII模式传输文件，用户也可以指定其他模式。ascii和binary命令的功能是设置传输的模式。用ASCII模式传输纯文本文件是非常好的，而二进制文件以二进制模式传输更为可靠。
- (8) 切换“#”提示。hash命令使FTP在每次传输完数据缓冲区中的数据后，就在屏幕上打印一个“#”字符。本命令在发送和接收文件时都可以使用。hash命令是一个开关。

- **【实例2.5】**
- 以下是下载一个文件的过程：
- [root @redflag /root]#ftp
- ftp> open 192.168.0.10
- Connected to 192.168.0.10.
- 220 mylinux.wlj.com FTP server (Version wu-2.6.1-16) ready.
- 530 Please login with USER and PASS.
- 530 Please login with USER and PASS.
- KERBEROS_V4 rejected as an authentication type

- Name (192.168.0.10:test): test
- 331 Password required for test.
- Password:
- 230-----
- 230-This is wanglongjie's FTP Server
- 230-----
- 230-
- 230 User test logged in.
- Remote system type is UNIX.

- Using binary mode to transfer files.
- ftp> pwd
- 257 "/home/test" is current directory.
- ftp> ls
- 227 Entering Passive Mode (192,168,0,10,249,113)
- 150 Opening ASCII mode data connection for directory listing.
- total 9963

- lrwxrwxrwx 1 root root 12 Feb 18 20:29
Desktop -> .Desktop_gb/
- -rw-rw-r-- 1 503 503 0 Mar 1 10:46 a.zip
- -rw-r--r-- 1 503 503 5044297 Feb 28 23:59
aaa
- drwxr-xr-x 2 root root 80 Feb 19 23:08
public_html
- drwxrwxr-x 2 503 503 288 Feb 26 13:59
ttt
- 226 Transfer complete.
- ftp> get a.zip /root/1.zip
- local: /root/1.zip remote: a.zip
- 227 Entering Passive Mode (192,168,0,10,20,213)

- 226 Transfer complete.
- ftp> close
- 221-You have transferred 0 bytes in 1 files.
- 221-Total traffic for this session was 783 bytes in 1 transfers.
- 221 Thank you for using the FTP service on mylinux.wlj.com.
- ftp> bye
- [root @redflag /root]#

- **2.5.5 wall、write、mesg**
- 1. wall——向任何用户终端发送字符消息
- 例如：
- `[root @redflag /root]#wall`
- 表示进入消息输入状态，可以输入一行或多行消息，按【Ctrl-d】键结束。在进行系统管理时，如果有紧急消息要通知所有在线用户，wall命令十分有用。

- 2. write 用户名 [终端]——向用户发送字符消息
- 例如:
- [root @redflag /root]#write user1
- 表示进入消息输入状态，可以输入一行或多行消息，按【Ctrl-d】键结束。write命令和下面介绍的mesg命令也有关。

- 3. `mesg [参数]`——控制他人向自己的终端发送消息的能力
- 参数选项：
- `y`：允许他人往自己的终端发送消息。
- `n`：不允许他人往自己的终端发送消息，但无法阻止`root`用户向自己发送信息。

- 例如:
- `[root @redflag /root]#mesg n`
- 表示其他用户用`wall`命令发送消息时，不会对自己的终端产生影响。
- `[root @redflag /root]#mesg`
- `is n`
- 显示当前终端是否允许他人往自己的终端发送消息，以上表示不允许。
-

- 2.5.6 mail
- mail 用户名或E-mail地址——SMTP客户端程序。
- 可以使用这个程序在系统内发送和接收邮件，也可以往Internet上的主机发送邮件或从Internet的主机接收邮件。例如：
- [root @redflag /root]#mail longkey
- Subject:This is a test mail
- Hello,longkey!
- Cc:
-

- 输入时按 **【Ctrl-d】** 键可以结束输入，把邮件发出。当longkey用户登录时，系统会提示“**You have mail**”。这时longkey用户可以直接使用**mail**命令来接收邮件和回复邮件。键入该命令时，出现“**&**”提示符，用“**?**”命令可以得到**mail**的帮助。**mail**的使用较为复杂，这里就不详细讨论了。

- 2.5.7 finger
- `finger [用户名@主机]`——显示主机系统中用户的信息。
- 例如:
- `[root @redflag /root]#finger`
- | Login | Name | Tty | Idle | Login | Time |
|-------|--------|--------|-------|-------|-------|
| | Office | Office | Phone | | |
| root | root | tty1 | 2 | Mar | 1 |
| | | | | | 09:44 |
| test | tty2 | 4 | Mar 1 | | 11:08 |
| test | pts/0 | | Mar 1 | | 11:13 |
- (wwwb.szpt.net)

- 显示用户当前登录的主机上的所有登录用户的信息。`finger`命令要求主机要提供`finger`服务，否则会连接失败。
- `[root @redflag /root]#finger longkey@192.168.0.10`
- Login: longkey Name: (null)
- Directory: /home/longkey Shell: /bin/bash
- Last login Sat Mar 1 08:32 (EST) on tty3
- No mail.
- No Plan.

- **2.5.8 netstat [参数选项]**
- netstat命令显示网络连接、路由表、网卡统计数等信息。
- 参数选项：
 - -i: 显示网卡的统计数。
 - -r: 显示路由表。
 - -a: 显示所有信息。
- 例如：
 - [root @redflag /root]#netstat -i
 - 表示网卡的统计数。

- 以下是网卡的各种统计数据，如：接收错误的包的数量。
- Kernel Interface table
- Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
- eth0 1500 0 460 0 0 0 320 0 0
0 BMRU
- lo 16436 0 54 0 0 0 54 0 0 0
LRU

2.6 其他命令

- **2.6.1 clear、dmesg、uname**
- 1. clear——清除屏幕
- 该命令相当于DOS下cls命令。
- 2. dmesg——显示内核引导时的状态信息
- 该命令对于内核引导出现故障时查找问题十分有用。

- 3. `uname -a`——显示系统的信息
- 例如:
- `[root @redflag /root]#uname -a`
- `Linux mylinux.test.com 2.4.7-2 #1 一 8月 27 14:04:34 CST 2001 i686 unknown`
- 说明主机名是mylinux.test.com，Linux的内核是2.4.7-2，CPU是i686结构。

- **2.6.2 date、cal**
- 1. date [时间]——显示或设置系统的时间
- 格式如下：
- [时间]: MMDDhhmmCCYY.SS
- [root @redflag /root]#date
- 三 2月 26 17:04:44 EST 2003
- 当前系统的时间为2003年2月26日 17:04:44，星期三。

- 2. cal [月][年]——显示指定年月的月历

- 如未指明年月，则显示当月的月历。

- [root @redflag /root]#cal 3 2003

- 三月 2003

- 日 一 二 三 四 五 六

- 1

- 2 3 4 5 6 7 8

- 9 10 11 12 13 14 15

- 16 17 18 19 20 21 22

- 23 24 25 26 27 28 29

- 30 31

- **2.6.3 help、man**

- 1. help [内置命令]——用于查看Linux内置命令的帮助

- 例如：

- `[root @redflag /root]#help`

- 输出全部内置命令。我们曾介绍Linux命令有内部命令和外部命令之分，如果读者不能区别命令是哪类，通过这一命令就可以区别它们了。

- `[root @redflag /root]#help alias`

- 输出内置命令alias的帮助。

- 2. man [命令名]——命令的帮助手册
- Linux的命令不仅多，而且每个命令的功能都十分强大，其参数也多如牛毛，幸运的是它有一个帮助系统能够帮助我们。例如：
- [root @redflag /root]#man ls
- 获得ls的帮助。
-

- 典型的man手册包括以下几个部分：
- **NAME:** 命令的名字。
- **SYNOPSIS:** 名字的概要，简单说明命令的使用方法。
- **DESCRIPTION:** 详细描述命令的使用，如各种参数选项的作用。
- **SEE ALSO:** 列出可能要查看的其他有关的手册页条目。
- **AUTHOR, COPYRIGHT:** 作者和版权等信息。

- **2.6.4 init、shutdown、halt、reboot、poweroff**
- 关闭Linux系统要采取正确的步骤，否则会引起文件系统损坏。由于Linux系统使用磁盘缓冲技术，Linux并不把数据立即写到磁盘上，因此不能直接用关闭电源来关机。正确的步骤应是执行如下指令：
 - `[root @redflag /root]#sync;sync;sync`
 - `[root @redflag /root]#shutdown -h now` (或下面我们介绍的关闭系统指令)

- 这三个sync可确保磁盘缓冲的内容全部写到磁盘中。此外，缺省时按【Ctrl】+【Alt】+【Del】键可以重新启动系统，用户可以禁止这一功能，方法是先找到/etc/inittab文件，把以下行屏蔽即可：
- `ca::ctrlaltdel:/sbin/shutdown -t3 -r now`

- 1. init n——改变系统的运行等级
- n: 指定的系统运行等级。
- 0——停止系统。
- 1——单用户。
- 2——多用户，但不支持NFS。
- 3——全多用户模式，即系统正常的模式。
- 5——进入X11(即窗口模式)。
- 6——重启系统。

- 例如：
- `[root @redflag /root]#init 6`
- 重新启动系统。
- `[root @redflag /root]#init 0`
- 关闭系统。

- 2. shutdown [参数] 时间 [警告消息]——在指定时间关闭系统
- 参数选项：
 - -r: 系统关闭后重启。
 - -h: 关闭后停机。
- 时间可以有以下几种形式：
 - now: 表示立即。
 - hh:mm: 指定绝对时间，hh表示小时，mm表示分钟。
 - +m: 表示m分钟以后。

- 例如:
- `[root @redflag /root]#shutdown -r +5 "System will reboot in 5 minutes."`
- 该命令警告用户5分钟后系统重启。
- 3. `halt`——立即停止系统
- 该命令不自动关闭电源，需要人工关闭电源。
- 4. `reboot`——立即重启系统
- 相当于命令:
- `shutdown -r now`

- 5. `poweroff`——立即停止系统，并且关闭电源
- 该命令要求计算机支持关机功能。相当于命令：
- `shutdown -h now`
-

- **2.6.5 alias、unalias、history**
- 1. alias 命令别名 =“命令行” ——创建命令的别名
- 例如:
- [root @redflag /root]#alias

- 显示已有的命令别名，读者在其中可以发现`ls='ls --color'`，这就是用户执行`ls`命令后为什么会用颜色表示不同的文件类别的原因了。用户可以把自已常用的长命令通过定义别名来用短命令替代。
- `[root @redflag /root]#alias mydir='ls --color'`
- 创建自己的命令`mydir`代替“`ls -color`”。

- 2. `unalias` 命令别名——删除已创建的别名
- 例如:
- `[root @redflag /root]#unalias mydir`
- 删除之前已经定义的别名`mydir`。

- 3. history——显示用户最近执行的命令
- 可以保留的历史命令数和环境变量 HISTSIZE 有关。只要在编号加 “!”，很容易地就可以重新运行 history 中的显示出的命令行。例如：
 - [root @redflag /root]#!25
 - 表示重新运行第25个历史命令。

- 2.6.6 su
- su [用户名]——改变用户的ID或成为超级用户。
- su可以让用户在一个登录的Shell中不退出就改变成为另一用户。如果su命令不跟用户名，则su命令缺省地成为超级用户。执行su命令后系统会要求输入密码。su之后，当前所有的用户变量都会传递过去。su命令在远程管理时相当有用，一般情况下超级用户(即root用户)不被允许远程登录。这时候，可以用普通用户Telnet到主机，再用su成为超级用户后进行远程管理，例如：
 - [test@mylinux test]\$ su
 - [root@redflag test]#

- **2.6.7 who、whoami、w、last**

- 1. who——显示谁登录系统

- 例如:

- [root @redflag /root]#who

- root tty1 Mar 1 12:17

- test tty2 Mar 1 12:17

- longkey tty3 Mar 1 12:17

- test pts/0 Mar 1 11:35

- 输出表示有4个用户登录系统，输出行中分别是用户名、登录的终端号和登录时间。

- 2. whoami——显示当前登录的用户名
- 例如:
- [root @redflag /root]#whoami
- root
- 表示当前用户是root。

- 3. w——显示谁登录系统并且在做什么
- 例如:
- [root @redflag /root]#w
- 12:18pm up 44 min, 4 users, load average: 0.06 ,
0.03, 0.00
- USER TTY FROM LOGIN@ IDLE JCPU PCPU
WHAT
- root tty1 - 12:17pm 1:07 0.04s 0.03s -bash
- test tty2 - 12:17pm 1:04 0.10s 0.03s -bash
- test pts/0 wwwb.szpt.net 11:35am 0.00s 0.13s
0.01s w

- 4. last——显示过去有多少用户在本机登录
- last 命令显示的是直到 /var/log/wtmp 文件创建以来有多少用户在本机登录。

- **2.6.8 rpm——安装软件包**

- 以前的Linux软件几乎都是源程序代码的形式，要安装时必须先取得压缩文件，解压得到源程序，再编译成可执行的文件，然后将相关的文件放到正确的目录中。RPM系统(Red Hat Package Manager)就是为了解决软件的安装问题而开发的，有了RPM就可以用一条命令完成软件的安装，RPM自动地帮我们完成了复杂的安装步骤。

- 软件开发人员将软件源程序代码、补丁(Patch)及安装指示包装成一个RPM套件(也就是一个rpm文件)。安装软件包只要一个rpm文件，执行rpm命令就可以轻松安装了。系统的RPM数据库记载了所有的以RPM方法安装的数据，因此可以非常方便地删除、查询和升级软件。
- rpm命令非常复杂，其格式如下：
- rpm [参数选项]
- 我们这里以实例来说明常用参数选项的使用。

- **【实例2.6】**
- [root @redflag
/root]#rpm -qa
- glibc-common-2.2.3-12
- htmlview-1.1.0-2
- mailcap-2.1.4-2
- rhl-gsg-6.1en-3
- sash-3.4-8
- basesystem-7.0-3

- chkconfig-1.2.22-1
- db2-2.4.14-4
- e2fsprogs-1.27-8
- finger-0.17-9
- gdbm-1.8.0-3
- hdparm-5.2-1
- isapnptools-1.22-2
- libattr-2.0.8-2

- **【实例2.7】**
- `[root @redflag /root]#rpm -q nfs-utils`
- `nfs-utils-0.3.1-8`
- 查询某一特定软件包，用“-q”参数，如果要显示软件包的完整信息，加“-i”参数。
- `[root @redflag /root]#rpm -qi nfs-utils`
- Name : nfs-utils Relocations: (not relocateable)
- Version : 0.3.1 Vendor: RedFlag SoftWare

- Release : 8 Build Date: 2001年08月03日 星期五 14时05分15秒
- Install date: 2003年03月19日 星期三 08时56分43秒
Build Host: xiejue.redflag-linux.com
- Group : System Environment/Daemons Source
RPM: nfs-utils-0.3.1-8.src.rpm
- Size : 520527 License: GPL
- Packager : RedFlag SoftWare<<http://www.redflag-linux.com>>

- Summary : NFS utilities and supporting daemons for the kernel NFS server. Description :
- The nfs-utils package provides a daemon for the kernel NFS server and related tools, which provides a
- much higher level of performance than the traditional Linux NFS server used by most users.

- This package also contains the showmount program. Showmount queries the mount daemon on a
- remote host for information about the NFS (Network File System) server on the remote host. For example,
- showmount can display the clients which are mounted on that host.

- **【实例2.8】**
- `#rpm -ql nfs-utils`
- `/etc/rc.d/init.d/nfs`
- `/etc/rc.d/init.d/nfslock`
- `/sbin/rpc.lockd`
- `/sbin/rpc.statd`
- `/sbin/rpcdebug`
- `/usr/sbin/exportfs`

- /usr/sbin/nfsstat
- /usr/sbin/nhfsstone
- /usr/sbin/rpc.mountd
-
- 查询nfs-utils软件包包含的文件。

- **【实例2.9】**
- 用rpm命令安装软件包时，软件包要求是以“.rpm”结尾的文件。把红旗3.0的光盘安装在 /mnt/cdrom 目录后，这些软件包就在 /mnt/cdrom/RedFlag/RPMS目录下。
- [root @redflag RPMS]#rpm -ivh zsh-4.0.1-1.i386.rpm
- Preparing...

[100%]
- 1:zsh

- 以上命令安装了zsh软件包。可以看到安装是多么轻松的一件事。如果要安装的软件已经安装了，系统会出现提示信息：
- Preparing...

[100%]
- package zsh-4.0.1-1 is already installed

- **【实例2.10】**
- 删除软件包使用“-e”参数。例如：
- `[root @redflag /root]#rpm -e zsh`
- 表示如果成功删除就没有输出。
- `[root @redflag /root]#rpm -e apache`
- `error: removing these packages would break dependencies:`

- apache is needed by apacheconf-0.7-2
- webserver is needed by mod_perl-1.24_01-2
- webserver is needed by mod_ssl-2.8.10-1
- webserver is needed by auth_ldap-1.6.0-5
- 表示如果一个软件包和其他软件包依存时，就无法删除它，这样就避免了许多错误。

- **【实例2.11】**
- 要升级软件包时用“-U”参数。例如：
- `#rpm -Uvh zsh-4.0.1-1.i386.rpm`
- RPM会自动删除旧版，安装新版，而旧版的配置保留。如果旧版不存在，就会自动安装新版软件包。“-F”参数则会要求RPM把系统中的旧版和要升级的版本进行比较，只有升级版比旧版新时才进行安装。如果系统中不存在旧的版本，RPM也不会安装软件包。

- **【实例2.12】**
- 要确认软件包的完整性，可以使用“-V”参数。
- `[root @redflag /root]#rpm -V nfs-utils`
- `.....T c /var/lib/nfs/etab`
- `S.5....T c /var/lib/nfs/rmtab`
- `.....T c /var/lib/nfs/xtab`
-

- RPM比较系统现有软件包和原有软件包的所有文件数据，如果完全一样，就不会有任何输出。有输出说明文件被改动，文件名前的“c”表示是设置文件。其余的表示如下：
- S为文件大小，L为符号连接，5为MD5 checksum，T为文件修改时间，D为设备，U为拥有者，G为所属组，M为模式(包括文件权限等) 被改变。

习 题

- 1. 使用 “ls -l”命令有以下输出：
- -rw-rw-r-- 1 longkey longkey 16 20A 24 22:23
chap1.txt
- 请解析各输出部分的含义。
- 2. 命令chmod 664 test.txt的结果是什么？
- 3. 增加所有用户对test.txt文件的读权限应使用什么命令？
- Chmod a+r teat.text

- 4. find命令和grep命令各用于什么目的?
- 5. 建立软链接和硬链接后, 用 “ls -l”命令显示链接, 有何差别?
- 6. 用crontab命令, 如果要每星期一12时执行命令echo hello|mail root, 作业应如何书写?
- 7. 网上有一软件包soft.tar.gz, 下载后应首先如何处理?
- 8. 测试自己的主机和某一主机是否通信正常, 通常使用__命令。
- A. telnet B. host C. ping D. ftp