*信号(signal)是一种软中断，信号机制是进程间通信的一种方式，采用异步通信方式*

## **一、信号类型**

Linux系统共定义了64种信号，分为两大类：可靠信号与不可靠信号，前32种信号为不可靠信号，后32种为可靠信号。

### **1.1 概念**

不可靠信号： 也称为非实时信号，不支持排队，信号可能会丢失, 比如发送多次相同的信号, 进程只能收到一次. 信号值取值区间为1~31；

可靠信号： 也称为实时信号，支持排队, 信号不会丢失, 发多少次, 就可以收到多少次. 信号值取值区间为32~64

### **1.2 信号表**

在终端，可通过kill -l查看所有的signal信号

| **取值** | **名称** | **解释** | **默认动作** |
| --- | --- | --- | --- |
| 1 | SIGHUP | 挂起 |  |
| 2 | SIGINT | 中断 |  |
| 3 | SIGQUIT | 退出 |  |
| 4 | SIGILL | 非法指令 |  |
| 5 | SIGTRAP | 断点或陷阱指令 |  |
| 6 | SIGABRT | abort发出的信号 |  |
| 7 | SIGBUS | 非法内存访问 |  |
| 8 | SIGFPE | 浮点异常 |  |
| 9 | SIGKILL | kill信号 | 不能被忽略、处理和阻塞 |
| 10 | SIGUSR1 | 用户信号1 |  |
| 11 | SIGSEGV | 无效内存访问 |  |
| 12 | SIGUSR2 | 用户信号2 |  |
| 13 | SIGPIPE | 管道破损，没有读端的管道写数据 |  |
| 14 | SIGALRM | alarm发出的信号 |  |
| 15 | SIGTERM | 终止信号 |  |
| 16 | SIGSTKFLT | 栈溢出 |  |
| 17 | SIGCHLD | 子进程退出 | 默认忽略 |
| 18 | SIGCONT | 进程继续 |  |
| 19 | SIGSTOP | 进程停止 | 不能被忽略、处理和阻塞 |
| 20 | SIGTSTP | 进程停止 |  |
| 21 | SIGTTIN | 进程停止，后台进程从终端读数据时 |  |
| 22 | SIGTTOU | 进程停止，后台进程想终端写数据时 |  |
| 23 | SIGURG | I/O有紧急数据到达当前进程 | 默认忽略 |
| 24 | SIGXCPU | 进程的CPU时间片到期 |  |
| 25 | SIGXFSZ | 文件大小的超出上限 |  |
| 26 | SIGVTALRM | 虚拟时钟超时 |  |
| 27 | SIGPROF | profile时钟超时 |  |
| 28 | SIGWINCH | 窗口大小改变 | 默认忽略 |
| 29 | SIGIO | I/O相关 |  |
| 30 | SIGPWR | 关机 | 默认忽略 |
| 31 | SIGSYS | 系统调用异常 |  |

对于signal信号，绝大部分的默认处理都是终止进程或停止进程，或dump内核映像转储。 上述的31的信号为非实时信号，其他的信号32-64 都是实时信号。

## **二、信号产生**

信号来源分为硬件类和软件类：

### **2.1 硬件方式**

* 用户输入：比如在终端上按下组合键ctrl+C，产生SIGINT信号；
* 硬件异常：CPU检测到内存非法访问等异常，通知内核生成相应信号，并发送给发生事件的进程；

### **2.2 软件方式**

通过系统调用，发送signal信号：kill()，raise()，sigqueue()，alarm()，setitimer()，abort()

* kernel,使用 kill\_proc\_info(）等
* native,使用 kill() 或者raise()等
* java,使用 Procees.sendSignal()等

## **三、信号注册和注销**

### **3.1 注册**

在进程task\_struct结构体中有一个未决信号的成员变量 struct sigpending pending。每个信号在进程中注册都会把信号值加入到进程的未决信号集。

* 非实时信号发送给进程时，如果该信息已经在进程中注册过，不会再次注册，故信号会丢失；
* 实时信号发送给进程时，不管该信号是否在进程中注册过，都会再次注册。故信号不会丢失；

### **3.2 注销**

* 非实时信号：不可重复注册，最多只有一个sigqueue结构；当该结构被释放后，把该信号从进程未决信号集中删除，则信号注销完毕；
* 实时信号：可重复注册，可能存在多个sigqueue结构；当该信号的所有sigqueue处理完毕后，把该信号从进程未决信号集中删除，则信号注销完毕；

## **四、信号处理**

内核处理进程收到的signal是在当前进程的上下文，故进程必须是Running状态。当进程唤醒或者调度后获取CPU，则会从内核态转到用户态时检测是否有signal等待处理，处理完，进程会把相应的未决信号从链表中去掉。

### **4.1 处理时机**

signal信号处理时机： 内核态 -> signal信号处理 -> 用户态：

* 在内核态，signal信号不起作用；
* 在用户态，signal所有未被屏蔽的信号都处理完毕；
* 当屏蔽信号，取消屏蔽时，会在下一次内核转用户态的过程中执行；

### **4.2 处理方式**

进程对信号的处理方式： 有3种

* 默认 接收到信号后按默认的行为处理该信号。 这是多数应用采取的处理方式。
* 自定义 用自定义的信号处理函数来执行特定的动作
* 忽略 接收到信号后不做任何反应。

### **4.3 信号安装**

进程处理某个信号前，需要先在进程中安装此信号。安装过程主要是建立信号值和进程对相应信息值的动作。

信号安装函数

* signal()：不支持信号传递信息，主要用于非实时信号安装；
* sigaction():支持信号传递信息，可用于所有信号安装；

其中 sigaction结构体

* sa\_handler:信号处理函数
* sa\_mask：指定信号处理程序执行过程中需要阻塞的信号；
* sa\_flags：标示位
  + SA\_RESTART：使被信号打断的syscall重新发起。
  + SA\_NOCLDSTOP：使父进程在它的子进程暂停或继续运行时不会收到 SIGCHLD 信号。
  + SA\_NOCLDWAIT：使父进程在它的子进程退出时不会收到SIGCHLD信号，这时子进程如果退出也不会成为僵 尸进程。
  + SA\_NODEFER：使对信号的屏蔽无效，即在信号处理函数执行期间仍能发出这个信号。
  + SA\_RESETHAND：信号处理之后重新设置为默认的处理方式。
  + SA\_SIGINFO：使用sa\_sigaction成员而不是sa\_handler作为信号处理函数。

函数原型：

int sigaction(int signum, const struct sigaction \*act, struct sigaction \*oldact);

* signum：要操作的signal信号。
* act：设置对signal信号的新处理方式。
* oldact：原来对信号的处理方式。
* 返回值：0 表示成功，-1 表示有错误发生。

### **4.4 信号发送**

* kill()：用于向进程或进程组发送信号；
* sigqueue()：只能向一个进程发送信号，不能像进程组发送信号；主要针对实时信号提出，与sigaction()组合使用，当然也支持非实时信号的发送；
* alarm()：用于调用进程指定时间后发出SIGALARM信号；
* setitimer()：设置定时器，计时达到后给进程发送SIGALRM信号，功能比alarm更强大；
* abort()：向进程发送SIGABORT信号，默认进程会异常退出。
* raise()：用于向进程自身发送信号；

### **4.5 信号相关函数**

信号集操作函数

* sigemptyset(sigset\_t \*set)：信号集全部清0；
* sigfillset(sigset\_t \*set)： 信号集全部置1，则信号集包含linux支持的64种信号；
* sigaddset(sigset\_t \*set, int signum)：向信号集中加入signum信号；
* sigdelset(sigset\_t \*set, int signum)：向信号集中删除signum信号；
* sigismember(const sigset\_t \*set, int signum)：判定信号signum是否存在信号集中。

信号阻塞函数

* sigprocmask(int how, const sigset\_t \*set, sigset\_t \*oldset))； 不同how参数，实现不同功能
  + SIG\_BLOCK：将set指向信号集中的信号，添加到进程阻塞信号集；
  + SIG\_UNBLOCK：将set指向信号集中的信号，从进程阻塞信号集删除；
  + SIG\_SETMASK：将set指向信号集中的信号，设置成进程阻塞信号集；
* sigpending(sigset\_t \*set))：获取已发送到进程，却被阻塞的所有信号；
* sigsuspend(const sigset\_t \*mask))：用mask代替进程的原有掩码，并暂停进程执行，直到收到信号再恢复原有掩码并继续执行进程。

# **Linux使用什么命令查看进程**

1. 查进程  
       ps命令查找与进程相关的PID号：  
       ps a 显示现行终端机下的所有程序，包括其他用户的程序。  
       ps -A 显示所有程序。  
       ps c 列出程序时，显示每个程序真正的指令名称，而不包含路径，参数或常驻服务的标示。  
       ps -e 此参数的效果和指定"A"参数相同。  
       ps e 列出程序时，显示每个程序所使用的环境变量。  
       ps f 用ASCII字符显示树状结构，表达程序间的相互关系。  
       ps -H 显示树状结构，表示程序间的相互关系。  
       ps -N 显示所有的程序，除了执行ps指令终端机下的程序之外。  
       ps s 采用程序信号的格式显示程序状况。  
       ps S 列出程序时，包括已中断的子程序资料。  
       ps -t<终端机编号> 指定终端机编号，并列出属于该终端机的程序的状况。  
       ps u 以用户为主的格式来显示程序状况。  
       ps x 显示所有程序，不以终端机来区分。  
        
       最常用的方法是ps aux,然后再通过管道使用grep命令过滤查找特定的进程,然后再对特定的进程进行操作。  
       ps aux | grep program\_filter\_word,ps -ef |grep tomcat  
     
   ps -ef|grep java|grep -v grep 显示出所有的java进程，去处掉当前的grep进程。  
        
   2.杀进程  
      使用kill命令结束进程：kill xxx  
      常用：kill －9 324  
      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 -9 NAME  
     
   3.进入到进程的执行文件所在的路径下，执行文件 ./文件名  
     
   附：  
     
   这是本人花了两天时间整理得来的，一些最常用的地球人都知道的命令就省去啦！最后提供pdf手册下载   
     
   1. 更改档案拥有者   
   命令 : chown [-cfhvR] [--help] [--version] user[:group] file...   
   功能 : 更改文件或者文件夹的拥有者   
   参数格式 :   
   　　    user : 新的档案拥有者的使用者 IDgroup : 新的档案拥有者的使用者群体(group)   
   　　       -c : 若该档案拥有者确实已经更改，才显示其更改动作   
   　　       -f : 若该档案拥有者无法被更改也不要显示错误讯息   
   　　       -h : 只对于连结(link)进行变更，而非该 link 真正指向的档案   
   　　       -v : 显示拥有者变更的详细资料   
   　       　-R : 对目前目录下的所有档案与子目录进行相同的拥有者变更(即以递回的方式逐个变更)   
     
   例如：chown -R oracle:oinstall /oracle/u01/app/oracle    
         更改目录拥有者为oracle   
     
   2. 修改权限   
       命令：chmod (change mode)   
       功能：改变文件的读写和执行权限。有符号法和八进制数字法。   
       选项：(1)符号法：   
     命令格式：chmod {u|g|o|a}{+|-|=}{r|w|x} filename   
             u (user)   表示用户本人。   
             g (group)  表示同组用户。   
             o (oher)   表示其他用户。   
             a (all)    表示所有用户。   
             +          用于给予指定用户的许可权限。   
             -          用于取消指定用户的许可权限。   
             =          将所许可的权限赋给文件。   
             r (read)   读许可，表示可以拷贝该文件或目录的内容。   
             w (write)  写许可，表示可以修改该文件或目录的内容。   
             x (execute)执行许可，表示可以执行该文件或进入目录。   
       
             (2)八进制数字法：     
     命令格式：chmod abc file   
     其中a,b,c各为一个八进制数字，分别表示User、Group、及Other的权限。   
             4 (100)    表示可读。   
             2 (010)    表示可写。   
             1 (001)    表示可执行。   
     若要rwx属性则4+2+1=7；   
     若要rw-属性则4+2=6；   
     若要r-x属性则4+1=5。   
     
       例如：# chmod a+rx filename   
               让所有用户可以读和执行文件filename。   
             # chmod go-rx filename   
               取消同组和其他用户的读和执行文件filename的权限。   
             # chmod 741 filename   
               让本人可读写执行、同组用户可读、其他用户可执行文件filename。   
     # chmod -R 755 /home/oracle   
       递归更改目录权限，本人可读写执行、同组用户可读可执行、其他用户可读可执行   
     
   3. 修改文件日期   
       命令：touch   
       格式：touch filenae   
       功能：改变文件的日期，不对文件的内容做改动，若文件不存在则建立新文件。   
       例如：% touch file   
     
   4. 链接文件   
       命令：ln (link)   
       格式：ln [option] filename linkname   
             ln [option] directory pathname   
       功能：为文件或目录建立一个链。其中，filename和directory是源文件名和   
             源目录名；linkname和pathname分别表示与源文件或源目录名相链接的   
             文件或目录。   
       选项：-s  为文件或目录建立符号链接。不加-s表示为文件或目录建立硬链接   
       注释：链接的目地在于，对一个文件或目录赋予两个以上的名字，使其可以出   
             现在不同的目录中，既可以使文件或目录共享，又可以节省磁盘空间。   
       例如：% ln -s filename linkname   
     
   5. 显示日期   
       命令：date   
       例如：% date   
     
   6. 显示日历   
       命令：cal (calendar)   
       格式：cal [month] year   
       功能：显示某年内指定的日历   
       例如：% cal 1998    
     
   7. 显示文件头部   
       命令：head   
       格式：head [option] filename   
       功能：显示文件的头部   
       选项：缺省  显示文件的头10行。   
             -i    显示文件的开始 i行。   
       例如：% head filename   
     
   8. 显示文件尾部   
       命令：tail   
       格式：tail [option] filename   
       功能：显示文件的尾部   
       选项：缺省  显示文件的末10行。   
             -i    显示文件最后 i行。   
             +i    从文件的第i行开始显示。   
       例如：% tail filename   
     
   9. 显示用户标识   
       命令：id   
       格式：id [option] [user]   
       功能：显示用户标识及用户所属的所有组。   
       选项：-a 显示用户名、用户标识及用户所属的所有组   
       注释：   
       例如：% id username   
     
   10. 查看当前登录的用户   
       命令：users   
     
   11. 显示都谁登录到机器上   
       命令：who   
       格式：who   
       功能：显示当前正在系统中的所有用户名字，使用终端设备号，注册时间。   
       例如：% who   
     
   12. 显示当前终端上的用户名   
       命令：whoami   
       格式：whoami   
       功能：显示出当前终端上使用的用户。   
       例如：% whoami   
     
   13. 寻找文件   
       命令：find   
       格式：find pathname [option] expression   
       功能：在所给的路经名下寻找符合表达式相匹配的文件。   
       选项：-name     表示文件名   
             -user     用户名，选取该用户所属的文件   
             -size     按大小查找，以block为单位，一个block是512B   
             -mtime n  按最后一次修改时间查找，选取n天内被修改的文件   
     -perm     按权限查找   
             -type     按文件类型查找   
     -atime    按最后一次访问时间查找   
     
       例如：% find ./ -name '\*abc\*' -print   
     
   14. 搜索文件中匹配符   
       命令：grep   
       格式：grep [option] pattern filenames   
       功能：逐行搜索所指定的文件或标准输入，并显示匹配模式的每一行。   
       选项：-i    匹配时忽略大小写   
     -v 找出模式失配的行   
     
       例如：% grep -i 'java\*' ./test/run.sh   
     
   15. 统计文件字数   
       命令：wc [option] filename   
       功能：统计文件中的文件行数、字数和字符数。   
       选项：-l 统计文件的行数   
   -w 统计文件的单词数   
   -c 统计文件的字符数   
       注释：若缺省文件名则指标准输入   
       例如：% wc -c ./test/run.sh   
     
   16. 显示磁盘空间   
       命令：df (disk free)   
       格式：df [option]   
       功能：显示磁盘空间的使用情况，包括文件系统安装的目录名、块设备名、总   
             字节数、已用字节数、剩余字节数占用百分比。   
       选项：   
   -a：显示全部的档案系统和各分割区的磁盘使用情形   
   -i：显示i -nodes的使用量   
   -k：大小用k来表示 (默认值)   
   -t：显示某一个档案系统的所有分割区磁盘使用量   
   -x：显示不是某一个档案系统的所有分割区磁盘使用量   
   -T：显示每个分割区所属的档案系统名称   
   -h: 表示使用「Human-readable」的输出，也就是在档案系统大小使用 GB、MB 等易读的格式。   
       注释：   
       例如：% df -hi   
     
   17. 查询档案或目录的磁盘使用空间   
       命令：du (disk usage)   
       格式：du [option] [filename]   
       功能：以指定的目录下的子目录为单位，显示每个目录内所有档案所占用的磁盘空间大小   
       选项：   
   -a：显示全部目录和其次目录下的每个档案所占的磁盘空间   
   -b：大小用bytes来表示 (默认值为k bytes)   
   -c：最后再加上总计 (默认值)   
   -s：只显示各档案大小的总合   
   -x：只计算同属同一个档案系统的档案   
   -L：计算所有的档案大小   
   -h: 表示档案系统大小使用 GB、MB 等易读的格式。   
       例如：% du -a   
   % du -sh /etc 只显示该目录的总合   
   % du /etc | sort -nr | more 统计结果用sort 指令进行排序，   
   sort 的参数 -nr 表示要以数字排序法进行反向排序。   
     
   18. 显示进程   
       命令：ps   
       格式：ps [option]   
       功能：显示系统中进程的信息。包括进程ID、控制进程终端、执行时间和命令。   
       选项：   
     -a 显示所有进程信息   
     -U uidlist 列出这个用户的所有进程   
             -e 显示当前运行的每一个进程信息   
             -f 显示一个完整的列表   
     -x 显示包括没有终端控制的进程状况 。   
       注释：   
       例如：% ps -ef   
     % ps -aux 然后再利用一个管道符号导向到grep去查找特定的进程,然后再对特定的进程进行操作。   
     
   19. 终止进程   
       命令：kill   
       格式：kill [option] pid   
       功能：向指定的进程送信号或终止进程。kill指令的用途是送一个signal给某一个process，   
       因为大部份送的都是用来杀掉 process 的 SIGKILL 或 SIGHUP ，因此称为 kill    
       选项：-9  强行终止进程   
       注释：pid标示进程号，可由ps命令得到。   
       例如：% kill -9 pid   
       你也可以用 kill -l 来察看可代替 signal 号码的数目字。kill 的详细情形请参阅 man kill。   
     
   20. 查看自己的IP地址   
       命令：ifconfig   
       格式：ifconfig -a   
        
   21. 查看路由表   
       命令：netstat   
       格式：netstat -rn   
     
   22. 远程登录   
       命令：telnet   
       格式：telnet hostname   
     
   23. 文件传输   
       命令：ftp (file transfer program)   
       格式：ftp hostname   
       功能：网络文件传输及远程操作。   
       选项：ftp命令：   
              cd [dirname]  进入远程机的目录   
              lcd [dirname] 设置本地机的目录   
              dir/ls        显示远程的目录文件   
              bin           以二进制方式进行传输   
      asc           以文本文件方式进行传输   
              get/mget      从远程机取一个或多个文件   
              put/mput      向远程机送一个或多个文件   
              prompt        打开或关闭多个文件传送时的交互提示   
              close         关闭与远程机的连接   
              quit          退出ftp   
      !/exit ftp登陆状态下，!表示暂时退出ftp状态回到本地目录，exit表示返回ftp状态   
       注释：   
       例如：% ftp hostname   
     
   24. 查看自己的电子邮件   
       命令：mailx   
       格式：mailx   
       选项：   
   delete  删除   
   next    下一个   
   quit    退出   
            reply   回复      
     
   25. 回忆命令   
       命令：history   
       格式：history   
       功能：帮助用户回忆执行过的命令。   
       选项：   
       注释：   
       例如：% history   
     
   26. 网上对话   
       命令：talk   
       格式：talk username   
       功能：在网上与另一用户进行对话。   
       选项：   
       注释：对话时系统把终端分为上下两部分，上半部显示自己键入信息，下半部   
             显示对方用户键入的信息。键入delete或Ctrl+C则结束对话。   
       例如：% talk username   
     
   27. 允许或拒绝接受信息   
       命令：mesg (message)   
       格式：mesg [n/y]   
       功能：允许或拒绝其它用户向自己所用的终端发送信息。   
       选项：n 拒绝其它用户向自己所用的终端写信息   
             y 允许其它用户向自己所用的终端写信息（缺省值）   
       注释：   
       例如：% mesg n   
     
   28. 给其他用户写信息   
       命令：write   
       格式：write username [ttyname]   
       功能：给其他用户的终端写信息。   
       选项：   
       注释：若对方没有拒绝，两用户可进行交谈，键入EOF或Ctrl+C则结束对话。   
       例如：write username   
     
   29. 创建、修改、删除用户和群组   
       a. 创建群组：   
   例如： groupadd oinstall    创建群组名为oinstall的组   
   groupadd -g 344 dba    
   创建组号是344的组，此时在/etc/passwd文件中产生一个组ID（GID）是344的项目。   
       b. 修改群组：   
   groupmod:该命令用于改变用户组帐号的属性   
   groupmod –g 新的GID 用户组帐号名   
   groupmod –n 新组名 原组名：此命令由于改变用户组的名称   
     
       c. 删除群组：   
   groupdel 组名：该命令用于删除指定的组帐号   
     
       d. 新建用户：   
   命令： useradd [－d home] [－s shell] [－c comment] [－m [－k template]]   
   [－f inactive] [－e expire ] [－p passwd] [－r] name   
   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：指定用户登入时的启始目录。   
   -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。   
     
   举例： # useradd -g oinstall -G dba oracle  创建Oracle用户   
         
       e. 删除用户   
   命令： userdel 用户名   
   删除指定的用户帐号   
   userdel –r 用户名(userdel 用户名;rm 用户名)：删除指定的用户帐号及宿主目录   
   例：#useradd -g root kkk //把kkk用户加入root组里   
     
       f. 修改用户   
   命令： usermod  
   修改已有用户的信息   
   usermod –l 旧用户名 新用户名： 修改用户名   
   usermod –L 用户名： 用于锁定指定用户账号，使其不能登陆系统   
   usermod –U 用户名： 对锁定的用户帐号进行解锁   
   passwd –d 用户名： 使帐号无口令，即用户不需要口令就能登录系统   
   例：#usermod -l user2 user1 //把用户user2改名为user1   
     
   30. 启动、关闭防火墙   
   永久打开或则关闭   
   chkconfig iptables on   
   chkconfig iptables off   
   即时生效：重启后还原   
   service iptables start   
   service iptables stop   
        或者：   
   /etc/init.d/iptables start   
   /etc/init.d/iptables stop   
     
   31. 启动VSFTP服务   
   即时启动： /etc/init.d/vsftpd start   
   即时停止： /etc/init.d/vsftpd stop   
     
   开机默认VSFTP服务自动启动:   
   方法一:(常用\方便)   
   [root@localhost etc]# chkconfig --list|grep vsftpd ( 查看情况)   
   vsftpd          0:off   1:off   2:off   3:off   4:off   5:off   6:off   
   [root@localhost etc]# chkconfig vsftpd on  (执行ON设置)   
   或者:方法二:   
   修改文件 /etc/rc.local , 把行/usr/local/sbin/vsftpd & 插入文件中，以实现开机自动启动。   
     
   32. vi技巧   
   a. 进入输入模式   
   新增 (append)   
   a ：从光标所在位置後面开始新增资料，光标後的资料随新增资料向後移动。   
   A：从光标所在列最後面的地方开始新增资料。   
     
   插入 (insert)   
   i：从光标所在位置前面开始插入资料，光标後的资料随新增资料向後移动。   
   I ：从光标所在列的第一个非空白字元前面开始插入资料。   
     
   开始 (open)   
   o ：在光标所在列下新增一列并进入输入模式。   
   O: 在光标所在列上方新增一列并进入输入模式。   
   b. 退出vi   
   在指令模式下键入:q,:q!,:wq或:x(注意:号），就会退出vi。其中:wq和:x是存盘退出，而:q是直接退出，如果文件已有新的变化，vi会提示你保存文件而:q命令也会失效，这时你可以用:w命令保存文件后再用:q 退出，或用:wq或:x命令退出，如果你不想保存改变后的文件，你就需要用:q!命令，这个命令将不保存文件而直接退出vi。   
     
   c. 删除与修改文件的命令：   
   x：删除光标所在字符。   
   dd ：删除光标所在的列。   
   r ：修改光标所在字元，r 後接著要修正的字符。   
   R：进入取替换状态，新增文字会覆盖原先文字，直到按 [ESC] 回到指令模式下为止。   
   s：删除光标所在字元，并进入输入模式。   
   S：删除光标所在的列，并进入输入模式。   
     
   d. 屏幕翻滚类命令   
   Ctrl+u: 向文件首翻半屏   
   Ctrl+d: 向文件尾翻半屏   
   Ctrl+f: 向文件尾翻一屏   
   Ctrl＋b: 向文件首翻一屏   
   nz: 将第n行滚至屏幕顶部，不指定n时将当前行滚至屏幕顶部。   
     
   e. 删除命令   
   ndw或ndW: 删除光标处开始及其后的n-1个字   
   do: 删至行首   
   d$: 删至行尾   
   ndd: 删除当前行及其后n-1行   
   x或X: 删除一个字符，x删除光标后的，而X删除光标前的   
   Ctrl+u: 删除输入方式下所输入的文本   
     
   f. 搜索及替换命令   
   /pattern: 从光标开始处向文件尾搜索pattern   
   ?pattern: 从光标开始处向文件首搜索pattern   
   n: 在同一方向重复上一次搜索命令   
   N: 在反方向上重复上一次搜索命令   
   :s/p1/p2/g: 将当前行中所有p1均用p2替代   
   :n1,n2s/p1/p2/g: 将第n1至n2行中所有p1均用p2替代   
   :g/p1/s//p2/g: 将文件中所有p1均用p2替换   
     
   g. 复制，黏贴   
   (1) 选定文本块，使用v进入可视模式；移动光标键选定内容   
   (2) 复制选定块到缓冲区，用y；复制整行，用yy   
   (3) 剪切选定块到缓冲区，用d；剪切整行用dd   
   (4) 粘贴缓冲区中的内容，用p   
     
   h. 其他   
   在同一编辑窗打开第二个文件，用:sp [filename]   
   在多个编辑文件之间切换，用Ctrl+w

# **Shell向进程发送信号（kill、pkill和killall命令**

1.用kill来杀死某一个进程

#kill，加选项-9，加PID，表示杀死进程编号为PID的这个进程

# -1 重启

#kill -9 PID

#下面表示杀死进程PID为2899的进程

[root@localhost ~]# kill -9 2899

2.用killall杀死一类进程

#killall，加选项-9，加服务名， 表示杀死该服务的所有进程

# -1 重启

#killall -9 service

#下面表示杀死httpd服务的所有进程

[root@localhost ~]# killall -9 httpd

3.用pkill踢出某个终端

#pkill，加选项-9，加终端号，表示踢出该终端

# -1 重启

#pkill -9 pts/0

#下面表示提出我的远程登录终端pts/0

[root@localhost ~]# pkill -9 pts/0