kill 命令

linux kill命令详解

1．命令格式：

kill[参数][进程号]

2．命令功能：

发送指定的信号到相应进程。不指定型号将发送SIGTERM（15）终止指定进程。如果任无法终止该程序可用"-KILL" 参数，其发送的信号为SIGKILL(9) ，将强制结束进程，使用ps命令或者jobs 命令可以查看进程号。root用户将影响用户的进程，非root用户只能影响自己的进程。

3．命令参数：

-l 信号，若果不加信号的编号参数，则使用"-l"参数会列出全部的信号名称
-a 当处理当前进程时，不限制命令名和进程号的对应关系
-p 指定kill 命令只打印相关进程的进程号，而不发送任何信号
-s 指定发送信号
-u 指定用户

注意：

1、kill命令可以带信号号码选项，也可以不带。如果没有信号号码，kill命令就会发出终止信号(15)，这个信号可以被进程捕获，使得进程在退出之前可以清理并释放资源。也可以用kill向进程发送特定的信号。例如：

kill -2 123

它的效果等同于在前台运行PID为123的进程时按下Ctrl+C键。但是，普通用户只能使用不带signal参数的kill命令或最多使用-9信号。

2、kill可以带有进程ID号作为参数。当用kill向这些进程发送信号时，必须是这些进程的主人。如果试图撤销一个没有权限撤销的进程或撤销一个不存在的进程，就会得到一个错误信息。

3、可以向多个进程发信号或终止它们。

4、当kill成功地发送了信号后，shell会在屏幕上显示出进程的终止信息。有时这个信息不会马上显示，只有当按下Enter键使shell的命令提示符再次出现时，才会显示出来。

5、应注意，信号使进程强行终止，这常会带来一些副作用，如数据丢失或者终端无法恢复到正常状态。发送信号时必须小心，只有在万不得已时，才用kill信号(9)，因为进程不能首先捕获它。要撤销所有的后台作业，可以输入kill 0。因为有些在后台运行的命令会启动多个进程，跟踪并找到所有要杀掉的进程的PID是件很麻烦的事。这时，使用kill 0来终止所有由当前shell启动的进程，是个有效的方法。

4．使用实例：

实例1：列出所有信号名称

命令：

kill -l

输出：

```
[root@localhost test6]# kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     16) SIGSTKFLT
17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU
25) SIGXFSZ     26) SIGVTALRM   27) SIGPROF     28) SIGWINCH
29) SIGIO       30) SIGPWR      31) SIGSYS      34) SIGRTMIN
35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3  38) SIGRTMIN+4
39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6
59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

说明：

只有第9种信号(SIGKILL)才可以无条件终止进程，其他信号进程都有权利忽略。 下面是常用的信号：

HUP　1　终端断线

INT　2　中断（同 Ctrl + C）
QUIT　3　退出（同 Ctrl + \）
TERM　15　终止
KILL　9　强制终止
CONT　18　继续（与STOP相反， fg/bg命令）
STOP　19　暂停（同 Ctrl + Z）

实例2：得到指定信号的数值
命令：
输出：
[root@localhost test6]# kill -l KILL
9[root@localhost test6]# kill -l SIGKILL
9[root@localhost test6]# kill -l TERM
15[root@localhost test6]# kill -l SIGTERM
15[root@localhost test6]#

说明：

实例3：先用ps查找进程，然后用kill杀掉
命令：
kill 3268
输出：
[root@localhost test6]# ps -ef|grep vim
root　3268 2884 0 16:21 pts/1　00:00:00 vim install.log
root　3370 2822 0 16:21 pts/0　00:00:00 grep vim
[root@localhost test6]# kill 3268
[root@localhost test6]# kill 3268
-bash: kill: (3268) - 没有那个进程
[root@localhost test6]#
说明：

实例4：彻底杀死进程
命令：
kill –9 3268
输出：
[root@localhost test6]# ps -ef|grep vim
root　3268 2884 0 16:21 pts/1　00:00:00 vim install.log
root　3370 2822 0 16:21 pts/0　00:00:00 grep vim
[root@localhost test6]# kill –9 3268
[root@localhost test6]# kill 3268
-bash: kill: (3268) - 没有那个进程
[root@localhost test6]#

说明：

实例5：杀死指定用户所有进程
命令：
kill -9 $(ps -ef | grep peidalinux)
kill -u peidalinux
输出：
[root@localhost ~]# kill -9 $(ps -ef | grep peidalinux)
[root@localhost ~]# kill -u peidalinux
说明：
方法一，过滤出hnlinux用户进程并杀死

实例6：init进程是不可杀的
命令：
kill -9 1
输出：
[root@localhost ~]# ps -ef|grep init
root　1　0 0 Nov02 ?　00:00:00 init [3]
root　17563 17534 0 17:37 pts/1　00:00:00 grep init

```
[root@localhost ~]# kill -9 1
[root@localhost ~]# kill -HUP 1
[root@localhost ~]# ps -ef|grep init
root       1     0  0 Nov02 ?       00:00:00 init [3]
root    17565 17534  0 17:38 pts/1    00:00:00 grep init
[root@localhost ~]# kill -KILL 1
[root@localhost ~]# ps -ef|grep init
root       1     0  0 Nov02 ?       00:00:00 init [3]
root    17567 17534  0 17:38 pts/1    00:00:00 grep init
[root@localhost ~]#
```

说明：

init是Linux系统操作中不可缺少的程序之一。所谓的init进程，它是一个由内核启动的用户级进程。内核自行启动（已经被载入内存，开始运行，并已初始化所有的设备驱动程序和数据结构等）之后，就通过启动一个用户级程序init的方式，完成引导进程。

所以,init始终是第一个进程（其进程编号始终为1）。 其它所有进程都是init进程的子孙。init进程是不可杀的！