我们运行 kill -l 命令，可看到 Linux 支持的信号列表：

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	

列表中，编号为 1 ~ 31 的信号为传统 UNIX 支持的信号，是不可靠信号(非实时的)，编号为 32 ~ 63 的信号是后来扩充的，称做可靠信号(实时信号)。不可靠信号和可靠信号的区别在于前者不支持排队，可能会造成信号丢失，而后者不会。

下面我们对编号 1 -- 31 的信号进行讨论:

1) SIGHUP　本信号在用户终端连接(正常或非正常)结束时发出，通常是在终端的控制进程结束时，通知同一终端(session)内的各个作业，这时它们与控制终端不再关联。

注:登录 Linux 时，系统会分配给登录用户一个终端(Session)。在这个终端运行的所有程序，包括前台进程组和后台进程组，一般都属于这个 Session。当用户退出 Linux 登录时，前台进程组和后台有对终端输出的进程将会收到 SIGHUP 信号。这个信号的默认操作为终止进程，因此前台进程组和后台有终端输出的进程就会中止。不过可以捕获这个信号,比如 wget 能捕获 SIGHUP 信号，并忽略它，这样就算退出了 Linux 登录，wget 也能继续下载。此外，对于与终端脱离关系的守护进程，这个信号用于通知它重新读取配置文件。

2) SIGINT　程序终止(interrupt)信号，在用户键入 INTR 字符(通常是 Ctrl-C)时发出，用于通知前台进程组终止进程。

3) SIGQUIT　和 SIGINT 类似，但由 QUIT 字符(通常是 Ctrl-\)来控制。进程在因收到 SIGQUIT 退出时会产生 core 文件，在这个意义上类似于一个程序托错误信号。

4) SIGILL　　执行了非法指令，通常是因为可执行文件本身出现错误；
或者试图执行数据段，堆栈溢出时也有可能产生这个信号。

5) SIGTRAP　　由断点指令或其它 trap 指令产生，由 debugger 使用。

6) SIGABRT　　程序自己发现错误并调用 abort 时产生。

7) SIGBUS　　非法地址，包括内存地址对齐(alignment)出错。
比如访问一个四个字长的整数，但其地址不是 4 的倍数。
它与 SIGSEGV 的区别在于后者是由于对合法存储地址的非法访问触发的；
(如访问不属于自己存储空间或只读存储空间)。

8) SIGFPE　　在发生致命的算术运算错误时发出. 不仅包括浮点运算错误，还包括溢出及除数为 0 等其它所有的算术的错误。

9) SIGKILL　　用来立即结束程序的运行. 本信号不能被阻塞、处理和忽略。如果管理员发现某个进程终止不了，可尝试发送这个信号。

10) SIGUSR1　　留给用户使用

11) SIGSEGV　　试图访问未分配给自己的内存，或试图往没有写权限的内存地址写数据.

12) SIGUSR2　　留给用户使用

13) SIGPIPE　　管道破裂。
这个信号通常在进程间通信产生，比如采用 FIFO(管道)通信的两个进程，读管道没打开或者意外终止就往管道写，写进程会收到 SIGPIPE 信号。
此外用 Socket 通信的两个进程，写进程在写 Socket 的时候，读进程已经终止。

14) SIGALRM　　时钟定时信号，计算实际的时间或时钟时间. alarm 函数使用该信号.

15) SIGTERM　　程序结束(terminate)信号；与 SIGKILL 不同的是该信号可以被阻塞和处理。通常用来要求程序自己正常退出，shell 命令 kill 缺省产生这个信号。如果进程终止不了，我们才会尝试 SIGKILL。

17) SIGCHLD　　子进程结束时，父进程会收到这个信号。
如果父进程没有处理这个信号，也没有等待(wait)子进程，子进程虽然终止，但是还会在内核进程表中占有表项，这时的子进程称为僵尸进程。应该避免这种情况(父进程可以忽略 SIGCHILD 信号，或者捕捉它，或者 wait 它派生的子进程，或者父进程先终止，这时子进程的终止自动由 init 进程来接管)。

18) SIGCONT　　让一个停止(stopped)的进程继续执行。
本信号不能被阻塞. 可以用一个 handler 来让程序在由 stopped 状态变为继续执行时完成特定的工作. 例如：重新显示提示符。

19) SIGSTOP　　停止(stopped)进程的执行. 本信号不能被阻塞，处理或忽略。
注意它和 terminate 以及 interrupt 的区别:该进程还未结束，只是暂停执行。

20) SIGTSTP　　停止进程的运行，但该信号可以被处理和忽略.
用户键入 SUSP 字符时(通常是 Ctrl-Z)发出这个信号。

21) SIGTTIN　　当后台作业要从用户终端读数据时，该作业中的所有进程会收到 SIGTTIN 信号. 缺省时这些进程会停止执行。

22) SIGTTOU　　类似于 SIGTTIN，但在写终端(或修改终端模式)时收到。

23) SIGURG　　有“紧急”数据或 out-of-band 数据到达 socket 时产生。

24) SIGXCPU　　超过 CPU 时间资源限制.

这个限制可以由 getrlimit/setrlimit 来读取/改变。

25) SIGXFSZ　　当进程企图扩大文件以至于超过文件大小资源限制。

26) SIGVTALRM　虚拟时钟信号. 类似于 SIGALRM,但是计算的是该进程占用的 CPU 时间。

27) SIGPROF　　类似于 SIGALRM/SIGVTALRM，

但包括该进程使用的 CPU 时间以及系统调用的时间。

28) SIGWINCH　窗口大小改变时发出.

29) SIGIO　　文件描述符准备就绪，可以开始进行输入/输出操作.

30) SIGPWR　　Power failure

31) SIGSYS　　非法的系统调用。

在以上列出的信号中，

程序不可捕获、阻塞或忽略的信号：SIGKILL, SIGSTOP

不能恢复至默认动作的信号：　SIGILL, SIGTRAP

默认会导致进程流产的信号：　SIGABRT, SIGBUS, SIGFPE, SIGILL, SIGIOT, SIGQUIT, SIGSEGV, SIGTRAP, SIGXCPU, SIGXFSZ

默认会导致进程退出的信号：　SIGALRM, SIGHUP, SIGINT, SIGKILL, SIGPIPE, SIGPOLL, SIGPROF, SIGSYS, SIGTERM, SIGUSR1, SIGUSR2, SIGVTALRM

默认会导致进程停止的信号：　SIGSTOP, SIGTSTP, SIGTTIN, SIGTTOU

默认进程忽略的信号：　　SIGCHLD, SIGPWR, SIGURG, SIGWINCH

此外，　SIGIO 在 SVR4 是退出，在 4.3BSD 中是忽略；

SIGCONT 在进程挂起时是继续，否则是忽略，不能被阻塞。

SIGTERM 信号和 SIGKILL 信号都可以停止进程。

SIGTERM 信号，进程能捕捉这个信号，根据需要来关闭程序。在关闭程序之前，您可以结束打开的记录文件和完成正在做的任务。在某些情况下，假如进程正在进行作业而且不能中断，那么进程可以忽略这个 SIGTERM 信号。

SIGKILL 信号，进程是不能忽略的。这是一个“不管在做什么,立刻停止”的信号。假如发送 SIGKILL 信号给进程，Linux 就将进程停止在那里。

以下是其它网友的补充：

SIGHUP	终止进程	终端线路挂断
SIGINT	终止进程	中断进程
SIGQUIT	建立 CORE 文件终止进程，并且生成 core 文件	
SIGILL	建立 CORE 文件	非法指令
SIGTRAP	建立 CORE 文件	跟踪自陷
SIGBUS	建立 CORE 文件	总线错误
SIGSEGV	建立 CORE 文件	段非法错误
SIGFPE	建立 CORE 文件	浮点异常
SIGIOT	建立 CORE 文件	执行 I/O 自陷
SIGKILL	终止进程	杀死进程
SIGPIPE	终止进程	向一个没有读进程的管道写数据
SIGALARM	终止进程	计时器到时
SIGTERM	终止进程	软件终止信号
SIGSTOP	停止进程	非终端来的停止信号
SIGTSTP	停止进程	终端来的停止信号
SIGCONT	忽略信号	继续执行一个停止的进程
SIGURG	忽略信号	I/O 紧急信号
SIGIO	忽略信号	描述符上可以进行 I/O
SIGCHLD	忽略信号	当子进程停止或退出时通知父进程
SIGTTOU	停止进程	后台进程写终端
SIGTTIN	停止进程	后台进程读终端
SIGXGPU	终止进程	CPU 时限超时
SIGXFSZ	终止进程	文件长度过长
SIGWINCH	忽略信号	窗口大小发生变化
SIGPROF	终止进程	统计分布图用计时器到时
SIGUSR1	终止进程	用户定义信号 1
SIGUSR2	终止进程	用户定义信号 2
SIGVTALRM	终止进程	虚拟计时器到时