Linux信号列表及其详解

我们运行如下命令，可看到Linux支持的信号列表：

$ 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 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。

下面我们对编号小于SIGRTMIN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session内的各个作业, 这时它们与控制终端不再关联。

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 调用abort函数生成的信号。

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, 计算的是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。这种情况我们应该避免(父进程或者忽略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. 注意它和terminate以及interrupt的区别:该进程还未结束, 只是暂停执行. 本信号不能被阻塞, 处理或忽略.

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、阻塞或忽略的信号有：SIGKILL,SIGSTOP 不能恢复至默认动作的信号有：SIGILL,SIGTRAP 默认会导致进程流产的信号有：SIGABRT,SIGBUS,SIGFPE,SIGILL,SIGIOT,SIGQUIT,SIGSEGV,SIGTRAP,SIGXCPU,SIGXFSZ 默认会导致进程退出的信号有：SIGALRM,SIGHUP,SIGINT,SIGKILL,SIGPIPE,SIGPOLL,SIGPROF,SIGSYS,SIGTERM,SIGUSR1,SIGUSR2,SIGVTALRM 默认会导致进程停止的信号有：SIGSTOP,SIGTSTP,SIGTTIN,SIGTTOU 默认进程忽略的信号有：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。