<https://blog.csdn.net/wesleyluo/article/details/5279482>

**Linux 信号表**

Linux支持POSIX标准信号和实时信号。下面给出Linux Signal的简表，详细细节可以查看man 7 signal。

默认动作的含义如下：

Term    终止进程

|  |  |  |  |
| --- | --- | --- | --- |
| **信号** | **取值** | **默认动作** | **含义（发出信号的原因）** |
| SIGHUP | 1 | Term | 终端的挂断或进程死亡 |
| SIGINT | 2 | Term | 来自键盘的中断信号 |
| SIGQUIT | 3 | Core | 来自键盘的离开信号 |
| SIGILL | 4 | Core | 非法指令 |
| SIGABRT | 6 | Core | 来自abort的异常信号 |
| SIGFPE | 8 | Core | 浮点例外 |
| SIGKILL | 9 | Term | 杀死 |
| SIGSEGV | 11 | Core | 段非法错误(内存引用无效) |
| SIGPIPE | 13 | Term | 管道损坏：向一个没有读进程的管道写数据 |
| SIGALRM | 14 | Term | 来自alarm的计时器到时信号 |
| SIGTERM | 15 | Term | 终止 |
| SIGUSR1 | 30,10,16 | Term | 用户自定义信号1 |
| SIGUSR2 | 31,12,17 | Term | 用户自定义信号2 |
| SIGCHLD | 20,17,18 | Ign | 子进程停止或终止 |
| SIGCONT | 19,18,25 | Cont | 如果停止，继续执行 |
| SIGSTOP | 17,19,23 | Stop | 非来自终端的停止信号 |
| SIGTSTP | 18,20,24 | Stop | 来自终端的停止信号 |
| SIGTTIN | 21,21,26 | Stop | 后台进程读终端 |
| SIGTTOU | 22,22,27 | Stop | 后台进程写终端 |
|  |  |  |  |
| SIGBUS | 10,7,10 | Core | 总线错误（内存访问错误） |
| SIGPOLL |  | Term | Pollable事件发生(Sys V)，与SIGIO同义 |
| SIGPROF | 27,27,29 | Term | 统计分布图用计时器到时 |
| SIGSYS | 12,-,12 | Core | 非法系统调用(SVr4) |
| SIGTRAP | 5 | Core | 跟踪/断点自陷 |
| SIGURG | 16,23,21 | Ign | socket紧急信号(4.2BSD) |
| SIGVTALRM | 26,26,28 | Term | 虚拟计时器到时(4.2BSD) |
| SIGXCPU | 24,24,30 | Core | 超过CPU时限(4.2BSD) |
| SIGXFSZ | 25,25,31 | Core | 超过文件长度限制(4.2BSD) |
|  |  |  |  |
| SIGIOT | 6 | Core | IOT自陷，与SIGABRT同义 |
| SIGEMT | 7,-,7 |  | Term |
| SIGSTKFLT | -,16,- | Term | 协处理器堆栈错误(不使用) |
| SIGIO | 23,29,22 | Term | 描述符上可以进行I/O操作 |
| SIGCLD | -,-,18 | Ign | 与SIGCHLD同义 |
| SIGPWR | 29,30,19 | Term | 电力故障(System V) |
| SIGINFO | 29,-,- |  | 与SIGPWR同义 |
| SIGLOST | -,-,- | Term | 文件锁丢失 |
| SIGWINCH | 28,28,20 | Ign | 窗口大小改变(4.3BSD, Sun) |
| SIGUNUSED | -,31,- | Term | 未使用信号(will be SIGSYS) |

说明：

一些信号的取值是硬件结构相关的（一般alpha和sparc架构用第一个值，i386、ppc和sh架构用中间值，mips架构用第三个值， - 表示相应架构的取值未知）。

蓝色的是POSIX.1-1990标准信号。

SIGKILL和SIGSTOP信号不能被挂钩、阻塞或忽略。

青色的是SUSv2和POSIX.1-2001定义的信号。

在Linux 2.2（包括）内核之前，SIGSYS、SIGXCPU、SIGXFSZ和SIGBUS （SPARC和MIPS架构除外）的默认动作是终止进程，但没有core dump。Linux 2.4遵循POSIX.1-2001要求，这些信号的默认动作改为：终止进程同时做core dump。

橙色的是其他常见的信号。

信号29在Alpha上为SIGINFO / SIGPWR ，在Sparc上为SIGLOST。

SIGEMT没有在POSIX.1-2001中说明，但是在大多数的Unices中仍然能见到，典型的默认动作是终止进程并做core dump。

SIGPWR没有在POSIX.1-2001中说明，在使用它的一些Unices中典型的默认动作是忽略。

SIGIO没有在POSIX.1-2001中说明，在使用它的一些Unices中典型的默认动作是忽略。

    进程可以通过使用sigaction和signal系统调用来改变信号的默认处理方式（使用signal的可移植性差）。进程可以选择下列3种信号处理方式中的一种：

1、执行默认操作；

2、忽略该信号；

3、捕获该信号，但是通过信号句柄来调用自定义的处理函数。

    信号可能被阻塞。进程中的每个线程拥有独立的信号掩码，用来表示本线程的信号被阻塞。线程通过pthread\_sigmask来设置它的信号掩码。单线程程序可以用sigprocmask来操作信号掩码。在多线程程序中，所有线程处理一个指定信号的默认行为都是一样的。