C

Table of Signals

TABLE C.1 LISTS SOME OF THE LINUX SIGNALS YOU'RE MOST LIKELY to encounter or use. Note that some signals have multiple interpretations, depending on where they occur.

The names of the signals listed here are defined as preprocessor macros. To use them in your program, include `<signal.h>`. The actual definitions are in `/usr/include/sys/signum.h`, which is included as part of `<signal.h>`.

For a full list of Linux signals, including a short description of each and the default behavior when the signal is delivered, consult the `signal` man page in Section 7 by invoking the following:

```
% man 7 signal
```

Table C.1 **Linux Signals**

Name	Description
`SIGHUP`	Linux sends a process this signal when it becomes disconnected from a terminal. Many Linux programs use `SIGHUP` for an unrelated purpose: to indicate to a running program that it should reread its configuration files.

continues

Table C.1 **Continued**

Name	Description
`SIGINT`	Linux sends a process this signal when the user tries to end it by pressing Ctrl+C.
`SIGILL`	A process gets this signal when it attempts to execute an illegal instruction. This could indicate that the program's stack is corrupted.
`SIGABRT`	The `abort` function causes the process to receive this signal.
`SIGFPE`	The process has executed an invalid floating-point math instruction. Depending on how the CPU is configured, an invalid floating-point operation may return a special non-number value such as `inf` (infinity) or `NaN` (not a number) instead of raising `SIGFPE`.
`SIGKILL`	This signal ends a process immediately and cannot be handled.
`SIGUSR1`	This signal is reserved for application use.
`SIGUSR2`	This signal is reserved for application use.
`SIGSEGV`	The program attempted an invalid memory access. The access may be to an address that is invalid in the process's virtual memory space, or the access may be forbidden by the target memory's permissions. Dereferencing a "wild pointer" can cause a `SIGSEGV`.
`SIGPIPE`	The program has attempted to access a broken data stream, such as a socket connection that has been closed by the other party.
`SIGALRM`	The `alarm` system call schedules the delivery of this signal at a later time. See Section 8.13, "`setitimer`: Setting Interval Timers," in Chapter 8, "Linux System Calls," for information about `setitimer`, a generalized version of `alarm`.
`SIGTERM`	This signal requests that a process terminate. This is the default signal sent by the `kill` command.
`SIGCHLD`	Linux sends a process this signal when a child process exits. See Section 3.4.4, "Cleaning Up Children Asynchronously," in Chapter 3, "Processes."
`SIGXCPU`	Linux sends a process this signal when it exceeds the limit of CPU time that it can consume. See Section 8.5, "`getrlimit` and `setrlimit`: Resource Limits," in Chapter 8 for information on CPU time limits.
`SIGVTALRM`	The `setitimer` schedules the delivery of this signal at a future time. See Section 8.13, "`setitimer`: Setting Interval Timers."