

SIGHUP - The SIGHUP signal disconnects a process from the parent process. This can also be used to restart processes. For example, "killall -SIGUP compiz" will restart Compiz. This is useful for daemons with memory leaks.

SIGINT - This signal is the same as pressing ctrl-c. On some systems, "delete" + "break" sends the same signal to the process. The process is interrupted and stopped. However, the process can ignore this signal.

SIGQUIT - This is like SIGINT with the ability to make the process produce a core dump.

SIGILL - When a process performs a faulty, forbidden, or unknown function, the system sends the SIGILL signal to the process. This is the Illegal Signal.

SIGTRAP - This signal is used for debugging purposes. When a process has performed an action or a condition is met that a debugger is waiting for, this signal will be sent to the process.

SIGABRT - This kill signal is the abort signal. Typically, a process will initiate this kill signal on itself.

SIGBUS - When a process is sent the SIGBUS signal, it is because the process caused a bus error. Commonly, these bus errors are due to a process trying to use fake physical addresses or the process has its memory alignment set incorrectly.

SIGFPE - Processes that divide by zero are killed using SIGFPE. Imagine if humans got the death penalty for such math. NOTE: The author of this article was recently drug out to the street and shot for dividing by zero.

SIGKILL - The SIGKILL signal forces the process to stop executing immediately. The program cannot ignore this signal. This process does not get to clean-up either.

SIGUSR1 - This indicates a user-defined condition. This signal can be set by the user by programming the commands in sigusr1.c. This requires the programmer to know C/C++.

SIGSEGV - When an application has a segmentation violation, this signal is sent to the process.

SIGUSR2 - This indicates a user-defined condition.

SIGPIPE - When a process tries to write to a pipe that lacks an end connected to a reader, this signal is sent to the process. A reader is a process that reads data at the end of a pipe.

SIGALRM - SIGALRM is sent when the real time or clock time timer expires.

SIGTERM - This signal requests a process to stop running. This signal can be ignored. The process is given time to gracefully shutdown. When a program gracefully shuts down, that means it is given time to save its progress and release resources. In other words, it is not forced to stop. SIGINT is very similar to SIGTERM.

SIGCHLD - When a parent process loses its child process, the parent process is sent the SIGCHLD signal. This cleans up resources used by the child process. In computers, a child process is a process started by another process known as a parent.

SIGCONT - To make processes continue executing after being paused by the SIGTSTP or SIGSTOP signal, send the SIGCONT signal to the paused process. This is the CONTInue SIGnal. This signal is beneficial to Unix job control (executing background tasks).

SIGSTOP - This signal makes the operating system pause a process's execution. The process cannot ignore the signal.

SIGTSTP - This signal is like pressing ctrl-z. This makes a request to the terminal containing the process to ask the process to stop temporarily. The process can ignore the request.

SIGTTIN - When a process attempts to read from a tty (computer terminal), the process receives this signal.

SIGTTOU - When a process attempts to write from a tty (computer terminal), the process receives this signal.

SIGURG - When a process has urgent data to be read or the data is very large, the SIGURG signal is sent to the process.

SIGXCPU - When a process uses the CPU past the allotted time, the system sends the process this signal. SIGXCPU acts like a warning; the process has time to save the progress (if possible) and close before the system kills the process with SIGKILL.

SIGXFSZ - Filesystems have a limit to how large a file can be made. When a program tries to violate this limit, the system will send that process the SIGXFSZ signal.

SIGVTALRM - SIGVTALRM is sent when CPU time used by the process elapses.

SIGPROF - SIGPROF is sent when CPU time used by the process and by the system on behalf of the process elapses.

SIGWINCH - When a process is in a terminal that changes its size, the process receives this signal.

SIGIO - Alias to SIGPOLL or at least behaves much like SIGPOLL.

SIGPWR - Power failures will cause the system to send this signal to processes (if the system is still on).

SIGSYS - Processes that give a system call an invalid parameter will receive this signal.

SIGRTMIN* - This is a set of signals that varies between systems. They are labeled SIGRTMIN+1, SIGRTMIN+2, SIGRTMIN+3,, and so on (usually up to 15). These are user-defined signals; they must be programmed in the Linux kernel's source code. That would require the user to know C/C++.

SIGRTMAX* - This is a set of signals that varies between systems. They are labeled SIGRTMAX-1, SIGRTMAX-2, SIGRTMAX-3,, and so on (usually up to 14). These are user-defined signals; they must be programmed in the Linux kernel's source code. That would require the user to know C/C++.

SIGEMT - Processes receive this signal when an emulator trap occurs.

SIGINFO - Terminals may sometimes send status requests to processes. When this happens, processes will also receive this signal.

SIGLOST - Processes trying to access locked files will get this signal.

SIGPOLL - When a process causes an asynchronous I/O event, that process is sent the SIGPOLL signal.