

Title:

Unix Signals

Word Count:

232

Summary:

A Unix signal can be defined as a message that is sent to a process that is currently running.

Keywords:

unix signals

Article Body:

Unix signals are initiated either by running programs, computer administrators, or operators. One example of a signal is the SIGHUP signal, which is sent to the internet to instruct the Internet Daemon, also known as inetd, to repeat reading the Daemon's configuration file.

To interrupt the Daemon's currently running process (ID, or PID), key in the command `kill -SIGHUP 4140`.

It is also possible to use signals to terminate a currently running process. To end the inetd process entirely, key in the following command: `kill 4140`. This command will, by default, send a SIGTERM signal. If this command fails, then the SIGKILL command is entered to end this process: `kill -9 4140`. Since this command cannot be controlled, it is usually not a good idea to utilize it to stop a process. This command does not permit a process to clean up or to end gracefully.

How are Unix Signals Managed?

Every Unix signal results in a default range of effects when used with a Unix program. Computer programmers can opt to code their own applications for a customized response upon the receipt of most signals. The term used to describe these individually designed strings of code is "signal handlers".

Signal handlers are unable to redefine two unique signals. In every situation, SIGKILL is used to end a process. Similarly, SIGSTOP always sends a running application to the background when it is in the foreground.