

Overview of signals

Signals are asynchronous messages that are delivered to processes. They are identified by their signal number, but they also have somewhat human-friendly names that all start with `SIG`. Some common ones include:

`SIGINT`

Delivered when you type `CTRL-C`. By default, this stops the program.

`SIGQUIT`

Delivered when you type `CTRL-`. By default, this also stops the program, but programs treat this signal more seriously than `SIGINT`. This signal also attempts to produce a core dump of the program before exiting.

`SIGKILL`

There is no keyboard shortcut for this. This signal stops the program forcibly and cannot be overridden by the program.

`SIGTERM`

There is no keyboard shortcut for this either. It behaves the same way as `SIGQUIT`.

`SIGTSTP`

Delivered when you type `CTRL-Z`. By default, this pauses the program. In `bash`, if you type `CTRL-Z`, the current program will be paused, and `bash` (which can detect that you paused the current program) will start accepting more commands.

`SIGCONT`

Delivered when you run `fg` or `fg %NUMBER` in `bash`. This signal resumes a paused program.

`SIGTTIN`

Delivered to a background process that is trying to read input from the keyboard. By default, this pauses the program, since background processes cannot read input from the keyboard. When

you resume the background process with `SIGCONT` and put it in the foreground, it can try to read input from the keyboard again.

`SIGTTOU`

Delivered to a background process that is trying to write output to the terminal console, but there is another foreground process that is using the terminal. Behaves the same as `SIGTTIN` by default.
