

# 1 Linux Signal CheatSheet

## LINUX

- PDF Link: [cheatsheet-signal-A4.pdf](#), Category: linux
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-signal-A4>
- Related posts: CheatSheet: shell, CheatSheet: SRE/DevOps/Sysadmin, #denny-cheatsheets

File me Issues or star this repo.

## 1.1 Basic

Name	Command
Signal for interprocess communication	A Kernel notify a process that some condition has occurred.
<code>kill -l</code>	List all signals
Critical signals	The signals SIGKILL and SIGSTOP cannot be caught, blocked, or ignored.
SIGKILL, SIGTERM	Send kill signal to a given process. <code>kill -term \$pid, kill -9 \$pid</code>
SIGHUP	Reload or restart a process. <code>kill -HUP \$pid</code>
SIGHUP	Terminal warns dependent processes of logout
SIGINT	Users wish to interrupt the process
SIGCHILD	When a child process terminates, instruct OS to cleanup the resource
SIGSTOP	Suspend a process
SIGALRM	
SIGUSR1, SIGUSR2	User defined signals
SIGSTOP vs SIGTSTP	SIGSTOP can't be ignored by the targetted process, but SIGTSTP can
What happens, when pressing Ctrl-C	Kernel translates the ASCII character(^C) to SIGINT signal
Reference	wikipedia: Signal (IPC)

## 1.2 Background & Daemons

Name	Command
Suspend process: bring it to background	<code>kill -STOP \$pid, Ctrl-z</code>
Bring a background process to foreground	<code>fg \$JOB_ID</code>
List all background jobs	<code>jobs, jobs -l</code>
Put a running process to background, and ignore HUP signal	<code>nohup \$command &amp;</code>
nohup with redirection	<code>nohup \$command&gt;foo.out 2&gt;foo.err &lt; /dev/null &amp;</code>
Tools to bring process to background and detach signals	<code>nohup, tmux, screen</code>

## 1.3 More Resources

License: Code is licensignal under MIT License.