Contents

[**Kill Command** 2](#_Toc486093701)

[**Signals** 2](#_Toc486093702)

[ **SIGTERM** 2](#_Toc486093703)

[ **SIGKILL** 2](#_Toc486093704)

[**Commands with options** 2](#_Toc486093705)

[ Format 2](#_Toc486093706)

[ To list out all signals 2](#_Toc486093707)

[ To kill a process 3](#_Toc486093708)

[ To kill process forcefully and immediately 3](#_Toc486093709)

[ To kill a process by process name 3](#_Toc486093710)

[ To kill all instances of process 3](#_Toc486093711)

[ To kill more than one processes at a time 3](#_Toc486093712)

[**Notes** 3](#_Toc486093713)

**Kill Command**:

* Kill command is used to terminate the process.
* Kill command sends a specified signal to the system to instruct it to terminate the specified process. If no signal is specified, the TERM signal is sent.

**Different Signals**:

* Signal is an inter process communication used in Linux.
* It is an asynchronous notification sent to a process in order to instruct to perform specified event.
* Signal number is assigned to each and every signal. Signal number can be used with Kill command instead of using signal name.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal** | **Signal code** | **Signal number** | **Short description** | **Detailed description** |
| HUP | SIGHUP | 1 | Hangup | disconnect process from its parent process also used to restart process |
| INIT | SIGINIT | 2 | Keyboard interrupt | Ctrl+C process is interrupted and stopped signal can be ignored by the process may data loss occurs |
| QUIT |  | 3 | Keyboard quit | Ctrl+l OR q same as SIGINIT with ability to make the process to produce a core dump |
| KILL | SIGKILL | 9 | Kill, unblockable | Hardest and most unsafe kill forcefully kill force process to stop executing immediately cannot be ignored |
| TERM | SIGTERM | 15 | Terminate | Default safe way to kill instruct process to stop whatever doing and end itself can be ignored process is given time to gracefully shutdown |
| CONT |  | 18 | Continue | Make process continue executing after being paused by SIGSTOP or SIGTSTP |
| STOP | SIGSTOP | 19 | Stop, unblockable | pauses process’s execution cannot be ignored |
| TSTP | SIGTSTP | 20 | Keyboard stop | Ctrl+Z Make request to the terminal containing process to ask the process to stop temporarily  can be ignored |

* SIGTERM:
* This signal requests a process to stop running.
* The process is given time to gracefully shutdown. It means, the process is given a time to save its progress and release resources.
* So this signal is not forced to stop process.
* This signal can be ignored by process.
* Assigned signal number: 15
* SIGKILL**:**
* This signal forces the process to stop executing immediately.
* The process is not given time to gracefully shutdown.
* So this signal is forced to stop process.
* This signal cannot be ignored by process.
* Even if it terminates a process forcefully, it is not able to terminate zombie processes.
* Assigned signal number: 9

**Commands with options**:

* Format: kill <option> <PID>
* To list out all signals,

Command: kill -l

* To kill a process,

Command: kill <PID> **OR**

Command: kill -SIGTERM <PID>

* To kill process forcefully and immediately,

Command: kill -9 <PID> **OR**

Command: kill -SIGKILL <PID> **OR**

Command: kill -KILL <PID> **OR**

* To kill a process by process name,

Command: pkill <Process name>

* To kill all instances of process,

Command: killall <process name> **OR**

Command: killall <PID>

* To kill more than one processes at a time,

Command: kill <PID1> <PID2> <PID3>

* To kill processes matching selected pattern,

Command: killall -<signal name> <command pattern>

Example**:** killall -SIGTERM \*yes\*

* To kill all processes owned by particular user,

Command: killall -u <user name>

|  |  |
| --- | --- |
| Default section | Description |
| Term | causes a program to terminate (exit) once |
| Core | causes a program to save memory image (core dump) and then terminate |
| Stop | causes a program to stop executing (suspend) and wait to continue (resume) |

**Notes**:

* Kill command by default sends SIGTERM signal to the process if no signal is specified.
* Sometimes it happens that process is not terminated even if sending SIGTERM signal. In such a case, SIGKILL (kill -9) signal is sent to terminate process forcefully.
* Processes init/systemd cannot be killed even with SIGKILL signal because kernel ignores all signals to avoid any disaster.

So kill -9 1 won’t work.

PPID of init/systemd is 0. Even parent process of init/systemd cannot be killed.

gcore= take core dump of process