

Process Management

**Create
Monitor
And kill Processes**

types of Linux process

Interactive >>>>> find

automatic >>>>> at , cron

daemon >>>>> ssh

fork () creates a new process by duplicating the calling process. The new process, referred to as the **child**, is an exact duplicate of the calling process, referred to as the **parent**, except for the following points:

Create processes

A running application is a process.

Process identifier (**PID**)

Parent Process ID (**PPID**)

init pid 1 ppid 0

```
$ pstree
init--NetworkManager--dhclient
                        --dnsmasq
                        --2*[{NetworkManager}]
--3*[VBoxClient--{VBoxClient}]
--VBoxService--7*[{VBoxService}]
--accounts-daemon--{accounts-daemon}
--acpid
--at-spi-bus-laun--dbus-daemon
                  --3*[{at-spi-bus-laun}]
--at-spi2-registr--{at-spi2-registr}
--atd
--avahi-daemon--avahi-daemon
--bluetoothd
--colord--2*[{colord}]
--console-kit-dae--64*[{console-kit-dae}]
--cron
--cupsd
--2*[dbus-daemon]
--dbus-launch
--dconf-service--2*[{dconf-service}]
--evolution-sourc--2*[{evolution-sourc}]
--gconfd-2
--geoclue-master--2*[{geoclue-master}]
--6*[getty]
--gnome-keyring-d--4*[{gnome-keyring-d}]
```

Monitor processes

The top program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel.

-b : Batch-mode operation . useful for sending output to a file.

d or **s** Change delay/sleep. Space Immediately updates the display.

-n : Number-of-iterations limit as: -n number

-i Start top ignoring any idle or zombie processes.

k Kill a process. default signal, as with kill(1), is 15, SIGTERM.

r renice

Terminal

File Edit View Search Terminal Help

```
 6 root      rt    0      0      0      0 S    0.0  0.0   0:00.00 migration/0
 7 root      rt    0      0      0      0 S    0.0  0.0   0:00.02 watchdog/0
```

```
top - 23:45:45 up 3 min,  2 users,  load average: 2.13, 1.70, 0.72
```

```
Tasks: 131 total,  1 running, 130 sleeping,  0 stopped,  0 zombie
```

```
%Cpu(s):  8.3 us, 42.4 sy,  0.0 ni, 49.0 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
```

```
KiB Mem:  799972 total,  401120 used,  398852 free,   41484 buffers
```

```
KiB Swap:  817148 total,      0 used,  817148 free,  226648 cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1086	root	20	0	175m	19m	6576	S	23.7	2.5	0:12.20	Xorg
1542	moha	20	0	137m	13m	10m	S	12.6	1.7	0:02.38	gnome-panel
1522	moha	20	0	45528	10m	8816	S	4.0	1.4	0:01.58	metacity
1798	moha	20	0	5204	1332	992	R	3.1	0.2	0:01.64	top
1410	moha	20	0	216m	16m	12m	S	2.5	2.2	0:04.46	gnome-settings-
1572	moha	20	0	163m	17m	12m	S	0.9	2.3	0:03.89	nemo
1749	moha	20	0	144m	14m	10m	S	0.9	1.9	0:02.11	gnome-terminal
1587	moha	20	0	132m	12m	9928	S	0.6	1.6	0:01.55	indicator-apple
1273	moha	20	0	50260	8800	7148	S	0.3	1.1	0:01.44	gnome-session
1401	moha	20	0	17088	2968	2540	S	0.3	0.4	0:00.36	at-spi2-registr
1676	moha	20	0	41516	8340	6884	S	0.3	1.0	0:00.47	gnome-screensav
1	root	20	0	3504	1912	1248	S	0.0	0.2	0:04.45	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.31	ksoftirqd/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kworker/0:0
5	root	20	0	0	0	0	S	0.0	0.0	0:01.06	kworker/u:0
6	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	watchdog/0
8	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	cpuset
9	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper
10	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kdevtmpfs
11	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	sync_supers
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	bdi-default

\$

control CPU usage for PID

\$ top -p 1900

ps process status

```
▼ Terminal
File Edit View Search Terminal Help
$ ps -l
 F S   UID    PID   PPID   C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD
 0 S   1000    1754   1749   0   80   0  -   1537 wait   pts/1        00:00:00 bash
 0 R   1000    1865   1754   0   80   0  -   1242 -       pts/1        00:00:00 ps
$
```

\$ ps -p 1 -f

Uid **STIME**
Root **10:10**

PROCESS STATE CODES

- D un interruptible sleep (usually IO)
- R running or runnable (on run queue)
- S interruptible sleep (waiting for cpu)
- T stopped, either by a job control signal or because it is being traced
- W paging (not valid since the 2.6.xx kernel)
- X dead (should never be seen)
- Z defunct ("zombie") process, terminated but not reaped by its parent

Signals

- 1 Hangup (POSIX) - logout * modem status disconnected
- 2 Interrupt (ANSI) **ctrl+c**
- 9 Kill, unblockable (POSIX)
- 15 Termination – default kill command (ANSI)
- 18 Continue (POSIX). fg . bg
- 20 Keyboard stop (POSIX) - **ctrl+z**

Terminal

File Edit View Search Terminal Help

```
$ kill -l
```

1) SIGHUP	2) SIGINT	3) SIGQUIT	4) SIGILL	5) SIGTRAP
6) SIGABRT	7) SIGBUS	8) SIGFPE	9) SIGKILL	10) SIGUSR1
11) SIGSEGV	12) SIGUSR2	13) SIGPIPE	14) SIGALRM	15) SIGTERM
16) SIGSTKFLT	17) SIGCHLD	18) SIGCONT	19) SIGSTOP	20) SIGTSTP
21) SIGTTIN	22) SIGTTOU	23) SIGURG	24) SIGXCPU	25) SIGXFSZ
26) SIGVTALRM	27) SIGPROF	28) SIGWINCH	29) SIGIO	30) SIGPWR
31) SIGSYS	34) SIGRTMIN	35) SIGRTMIN+1	36) SIGRTMIN+2	37) SIGRTMIN+3
38) SIGRTMIN+4	39) SIGRTMIN+5	40) SIGRTMIN+6	41) SIGRTMIN+7	42) SIGRTMIN+8
43) SIGRTMIN+9	44) SIGRTMIN+10	45) SIGRTMIN+11	46) SIGRTMIN+12	47) SIGRTMIN+13
48) SIGRTMIN+14	49) SIGRTMIN+15	50) SIGRTMAX-14	51) SIGRTMAX-13	52) SIGRTMAX-12
53) SIGRTMAX-11	54) SIGRTMAX-10	55) SIGRTMAX-9	56) SIGRTMAX-8	57) SIGRTMAX-7
58) SIGRTMAX-6	59) SIGRTMAX-5	60) SIGRTMAX-4	61) SIGRTMAX-3	62) SIGRTMAX-2
63) SIGRTMAX-1	64) SIGRTMAX			

Example

kill -SIGKILL [pid]

kill -9 [pid] [pid]

killall sleep

pkill sleep

basic process management

By default a process is started in the foreground and it is the only one to receive keyboard input.

Use **CTRL+Z** to suspend it.

To start a process in the background use the **&**

Example

start a process directly in background

```
$ sleep 100 &
```

```
[1] 1969
```

```
[1]+ Done sleep 100
```

```
[1] + Done(2)      ls no_such_file &
```

```
set -o monitor      # monitor on
```

```
set +o monitor      # monitor off
```

```
set -o              # show all option
```

Example

start a process and put it in background

This is useful when you start something long,
and realize you need your prompt back to give some
other command.

Example

\$ sleep 100

^Z

[1]+ Stopped sleep 100

\$ bg

[1]+ sleep 100 &

\$

[1]+ Done sleep 100

Move a process from background to foreground

Usefull if we want to bring back on foreground a command we have sent in the bg.

```
sleep 100 & ; sleep 100; < CTRL > Z ;  
bg ; jobs ; fg 2
```

jobs

Lists the active jobs

\$ jobs

[1]- Running sleep 100 &

[2]+ Running sleep 100 &

\$ kill %jobnumber

\$ kill %2

Start a process, log off and have the process running

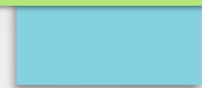
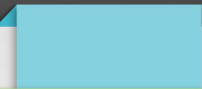
```
$ nohup myscript.sh &
```

nohup: ignoring input and appending output to
'no hup.out'

Log off from a terminal keeping active a process we started without nohup

You have to log off from the terminal, but you are in the middle of that long task what to do ?

```
mylongtask.sh ; < ctrl > Z; bg; disown -h %1
```



nice

run a program with modified scheduling priority

\$ nice -n -10 top -i

\$ nice -n --10 top -i

-20 (most favorable scheduling)

19 (least favorable)

?

?