

Version : **2024.01**

Last update : 2024/11/27 08:50

# RH12410 - Process Management

## Contents

- **RH12410 - Process Management**
  - Contents
  - Presentation
  - Process Types
  - Lab #1 - Process commands
    - 1.1 - The ps command
    - 1.2 - The pgrep command
    - 1.3 - The pstree command
    - 1.4 - The top command
    - 1.5 - The fg, bg and jobs commands
    - 1.6 - The wait command
    - 1.7 - The nice command
    - 1.8 - The renice command
    - 1.9 - The nohup command
    - 1.10 - The kill command
    - 1.11 - The pkill command

## Présentation

A process is a binary file that is loaded into memory and executed. When the file is loaded it needs the operating system to supply it with information such that it can execute correctly. Collectively, this information is referred to as the **process environment** and includes:

---

- A unique process ID (PID),
- The Parent PID (PPID),
- A User ID (UID),
- A Groupe ID (GID),
- Processing time,
- The process priority,
- The current working directory,
- A list of open files..

This information is stored in the **/proc** directory:

```
[root@redhat9 ~]# cd /proc; ls -d [0-9]*
1      10446 10597 17    22   28   34   406  423  465  504  535  540  5671  5705  5724  5737  5765  5950  6
6200  6384  6444  6462  6571  6668  708  7429  76   77   79   8946  9400
10     10590 12     18    23   29   35   415  43   47   51   536  541  5675  5707  5725  5742  5787  5952  6049
6206  6410  6451  6473  6599  6674  709  743  7618  78   7900  90   9490
10062  10591 13     2     24   3    36   42   434  48   511  537  542  5698  5708  5728  5743  58   5956  61
629    6432  6452  6477  6611  69    710  7430  7619  7896  8    9357  95
10066  10592 14     20    242  30   38   420  44   49   52   538  543  57   5710  5734  5744  5851  5963  614
6318  6433  6453  6479  6617  705   711  7431  7620  7897  80   9365  9543
10223  10595 15     2014  245  32   4    421  45   5    53   539  55   5700  5714  5735  5757  5863  5968  6195
6377  6436  6454  6482  6626  706   712  7432  7621  7898  81   9369  96
10383  10596 16     21    27   33   403  422  46   50   534  54   56   5704  5717  5736  5763  59   5975  62
6381  6439  6455  6486  6640  707   742  7433  7622  7899  853  9392  9896
```

Each directory refers to a process PID. It contains data from the **process environment**, for example :

```
[root@redhat9 proc]# cd 1 ; ls -l
total 0
-r--r--r--. 1 root root 0 Sep 25 15:29 arch_status
dr-xr-xr-x. 2 root root 0 Sep 25 12:44 attr
-rw-r--r--. 1 root root 0 Sep 25 15:29 autogroup
-r----- . 1 root root 0 Sep 25 15:29 auxv
-r--r--r--. 1 root root 0 Sep 25 12:44 cgroup
```

```
--w-----. 1 root root 0 Sep 25 15:29 clear_refs
-r--r--r--. 1 root root 0 Sep 25 12:44 cmdline
-rw-r--r--. 1 root root 0 Sep 25 12:44 comm
-rw-r--r--. 1 root root 0 Sep 25 15:29 coredump_filter
-r--r--r--. 1 root root 0 Sep 25 15:29 cpu_resctrl_groups
-r--r--r--. 1 root root 0 Sep 25 15:29 cpuset
lrwxrwxrwx. 1 root root 0 Sep 25 14:56 cwd -> /
-r-----. 1 root root 0 Sep 25 12:44 environ
lrwxrwxrwx. 1 root root 0 Sep 25 12:44 exe -> /usr/lib/systemd/systemd
dr-x-----. 2 root root 0 Sep 25 12:44 fd
dr-xr-xr-x. 2 root root 0 Sep 25 14:56 fdinfo
-rw-r--r--. 1 root root 0 Sep 25 12:44 gid_map
-r-----. 1 root root 0 Sep 25 15:29 io
-r-----. 1 root root 0 Sep 25 15:29 ksm_merging_pages
-r--r--r--. 1 root root 0 Sep 25 15:29 limits
-rw-r--r--. 1 root root 0 Sep 25 12:44 loginuid
dr-x-----. 2 root root 0 Sep 25 15:29 map_files
-r--r--r--. 1 root root 0 Sep 25 12:44 maps
-rw-----. 1 root root 0 Sep 25 15:29 mem
-r--r--r--. 1 root root 0 Sep 25 12:44 mountinfo
-r--r--r--. 1 root root 0 Sep 25 12:44 mounts
-r-----. 1 root root 0 Sep 25 15:29 mountstats
dr-xr-xr-x. 53 root root 0 Sep 25 12:44 net
dr-x--x--x. 2 root root 0 Sep 25 12:44 ns
-r--r--r--. 1 root root 0 Sep 25 15:29 numa_maps
-rw-r--r--. 1 root root 0 Sep 25 15:29 oom_adj
-r--r--r--. 1 root root 0 Sep 25 15:29 oom_score
-rw-r--r--. 1 root root 0 Sep 25 15:29 oom_score_adj
-r-----. 1 root root 0 Sep 25 15:29 pagemap
-r-----. 1 root root 0 Sep 25 15:29 patch_state
-r-----. 1 root root 0 Sep 25 15:29 personality
-rw-r--r--. 1 root root 0 Sep 25 15:29 projid_map
lrwxrwxrwx. 1 root root 0 Sep 25 12:44 root -> /
-rw-r--r--. 1 root root 0 Sep 25 15:29 sched
```

```
-r--r--r--. 1 root root 0 Sep 25 15:29 schedstat
-r--r--r--. 1 root root 0 Sep 25 12:44 sessionid
-rw-r--r--. 1 root root 0 Sep 25 12:44 setgroups
-r--r--r--. 1 root root 0 Sep 25 15:29 smaps
-r--r--r--. 1 root root 0 Sep 25 15:29 smaps_rollup
-r-----. 1 root root 0 Sep 25 15:29 stack
-r--r--r--. 1 root root 0 Sep 25 14:56 stat
-r--r--r--. 1 root root 0 Sep 25 15:29 statm
-r--r--r--. 1 root root 0 Sep 25 12:44 status
-r-----. 1 root root 0 Sep 25 15:29 syscall
dr-xr-xr-x. 3 root root 0 Sep 25 12:44 task
-rw-r--r--. 1 root root 0 Sep 25 15:29 timens_offsets
-r--r--r--. 1 root root 0 Sep 25 15:29 timers
-rw-rw-rw-. 1 root root 0 Sep 25 15:29 timerslack_ns
-rw-r--r--. 1 root root 0 Sep 25 12:44 uid_map
-r--r--r--. 1 root root 0 Sep 25 15:29 wchan
```



**Important** - Note that the content of the files is of little or of no direct use to a System Administrator.

## Process Types

There are three types of process:

- **interactive** - processes generated by typing a command in a terminal,
- **batch** - processes generated by the system itself,
- **daemon** - processes that do not have a parent terminal.

A process can be in one of nine *process states* :

- *user mode* - the process is executing in user mode,
- *kernel mode* - the process is executing in kernel mode,
- *sleeping* - the process is sleeping,
- *swap* - the process is sleeping in swap,
- *new* - the process is new,
- *waiting* - the process is waiting for a resource other than the processor,
- *runnable* - the process has all the resources it requires except the processor itself,
- *elected* - the process is in the processor,
- *zombie* - the process has terminated and is waiting to be killed by the system.

## LAB #1 - Process commands

### 1.1 - The ps command

The output from this command shows the processes attached to the current terminal:

```
[root@redhat9 ~]# cd ~  
  
[root@redhat9 ~]# ps  
  PID TTY          TIME CMD  
 10062 pts/0    00:00:00 su  
 10066 pts/0    00:00:00 bash  
 10602 pts/0    00:00:00 ps
```

You can get more details by using the **-l** switch:

```
[root@redhat9 ~]# ps -l  
F S  UID      PID      PPID  C  PRI  NI ADDR  SZ  WCHAN  TTY          TIME CMD  
4 S   0       10062    9400   0  80   0 - 58719 do_wai pts/0    00:00:00 su  
4 S   0       10066   10062   0  80   0 - 56056 do_wai pts/0    00:00:00 bash  
4 R   0       10604   10066   0  80   0 - 56370 -      pts/0    00:00:00 ps
```

Note in this output :

<b>F</b>	Process flag. The value of 4 means the process is using root privileges.
<b>S</b>	The process state - S (sleeping), R (In run queue), Z (zombie), N (low priority), D (uninterruptible sleep), T (Traced)
<b>UID</b>	User ID of the user who has started the process
<b>PID</b>	Process ID
<b>PPID</b>	Parent PID
<b>C</b>	Priority factor
<b>PRI</b>	Process priority
<b>NI</b>	Process nice value
<b>ADDR</b>	Memory address
<b>SZ</b>	Virtual memory usage
<b>WCHAN</b>	Name of the kernel function in which the process is asleep
<b>TTY</b>	Name of the terminal in which the process was started
<b>TIME</b>	Processing time
<b>CMD</b>	The command that generated the process

To view the process table, use the ps command with the l and x switches:

```
[root@redhat9 ~]# ps lx | more
F  UID      PID      PPID  PRI  NI       VSZ   RSS WCHAN  STAT TTY          TIME COMMAND
4   0         1         0   20   0 190992 17880 ep_pol Ss   ?           0:13 /usr/lib/systemd/systemd rhgb --
switched-root --system --deserialize 31
1   0         2         0   20   0       0       0 kthrea S    ?           0:00 [kthreadd]
1   0         3         2    0 -20       0       0 rescue I<  ?           0:00 [rcu_gp]
1   0         4         2    0 -20       0       0 rescue I<  ?           0:00 [rcu_par_gp]
1   0         5         2    0 -20       0       0 rescue I<  ?           0:00 [slub_flushwq]
1   0         6         2    0 -20       0       0 rescue I<  ?           0:00 [netns]
1   0         8         2    0 -20       0       0 worker I<  ?           0:00 [kworker/0:0H-events_highpri]
1   0        10         2    0 -20       0       0 rescue I<  ?           0:00 [mm_percpu_wq]
1   0        12         2   20   0       0       0 rcu_ta I    ?           0:00 [rcu_tasks_kthre]
1   0        13         2   20   0       0       0 rcu_ta I    ?           0:00 [rcu_tasks_rude_]
1   0        14         2   20   0       0       0 rcu_ta I    ?           0:00 [rcu_tasks_trace]
```

1	0	15	2	20	0	0	0	smpboo	S	?	0:00	[ksoftirqd/0]
1	0	16	2	20	0	0	0	rcu_gp	I	?	0:01	[rcu_preempt]
1	0	17	2	-100	-	0	0	smpboo	S	?	0:00	[migration/0]
1	0	18	2	-51	-	0	0	smpboo	S	?	0:00	[idle_inject/0]
1	0	20	2	20	0	0	0	smpboo	S	?	0:00	[cpuhp/0]
1	0	21	2	20	0	0	0	smpboo	S	?	0:00	[cpuhp/1]
1	0	22	2	-51	-	0	0	smpboo	S	?	0:00	[idle_inject/1]
1	0	23	2	-100	-	0	0	smpboo	S	?	0:00	[migration/1]
1	0	24	2	20	0	0	0	smpboo	S	?	0:00	[ksoftirqd/1]
1	0	27	2	20	0	0	0	smpboo	S	?	0:00	[cpuhp/2]
1	0	28	2	-51	-	0	0	smpboo	S	?	0:00	[idle_inject/2]
1	0	29	2	-100	-	0	0	smpboo	S	?	0:00	[migration/2]
1	0	30	2	20	0	0	0	smpboo	S	?	0:00	[ksoftirqd/2]
1	0	32	2	0	-20	0	0	worker	I<	?	0:00	[kworker/2:0H-events_highpri]
1	0	33	2	20	0	0	0	smpboo	S	?	0:00	[cpuhp/3]
1	0	34	2	-51	-	0	0	smpboo	S	?	0:00	[idle_inject/3]
1	0	35	2	-100	-	0	0	smpboo	S	?	0:00	[migration/3]
1	0	36	2	20	0	0	0	smpboo	S	?	0:00	[ksoftirqd/3]
1	0	38	2	0	-20	0	0	worker	I<	?	0:00	[kworker/3:0H-events_highpri]
5	0	42	2	20	0	0	0	devtmp	S	?	0:00	[kdevtmpfs]
1	0	43	2	0	-20	0	0	rescue	I<	?	0:00	[inet_frag_wq]
1	0	44	2	20	0	0	0	kaudit	S	?	0:00	[kauditd]
1	0	45	2	20	0	0	0	watchd	S	?	0:00	[khungtaskd]
1	0	46	2	20	0	0	0	oom_re	S	?	0:00	[oom_reaper]
1	0	47	2	0	-20	0	0	rescue	I<	?	0:00	[writeback]
1	0	48	2	20	0	0	0	kcompa	S	?	0:02	[kcompactd0]
1	0	49	2	25	5	0	0	ksm_sc	SN	?	0:00	[ksmd]
1	0	50	2	39	19	0	0	khugep	SN	?	0:00	[khugepaged]
1	0	51	2	0	-20	0	0	rescue	I<	?	0:00	[cryptd]
1	0	52	2	0	-20	0	0	rescue	I<	?	0:00	[kintegrityd]
1	0	53	2	0	-20	0	0	rescue	I<	?	0:00	[kblockd]
1	0	54	2	0	-20	0	0	rescue	I<	?	0:00	[blkcg_punt_bio]
1	0	55	2	0	-20	0	0	rescue	I<	?	0:00	[tpm_dev_wq]
1	0	56	2	0	-20	0	0	rescue	I<	?	0:00	[md]

```

1      0      57      2      0 -20      0      0 rescue I<  ?      0:00 [md_bitmap]
1      0      58      2      0 -20      0      0 rescue I<  ?      0:00 [edac-poller]
1      0      59      2 -51      -      0      0 kthrea S    ?      0:00 [watchdogd]
1      0      61      2      0 -20      0      0 worker I<  ?      0:00 [kworker/0:1H-kblockd]
1      0      62      2     20     0      0      0 kswapd S    ?      0:00 [kswapd0]
1      0      69      2      0 -20      0      0 rescue I<  ?      0:00 [kthrotld]
1      0      76      2      0 -20      0      0 rescue I<  ?      0:00 [acpi_thermal_pm]
1      0      77      2      0 -20      0      0 rescue I<  ?      0:00 [kmpath_rdacd]
1      0      78      2      0 -20      0      0 rescue I<  ?      0:00 [kaluad]
--More--
[q]

```

This output contains some additional information:

<b>VSZ</b>	The same thing as SZ in the previous example
<b>RSS</b>	Memory in KB used by the process
<b>STAT</b>	The same thing as S in the previous example

Using the a, u and x switches you obtain the following output:

```

[root@redhat9 ~]# ps aux | more
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.2 190992 17880 ?        Ss   Sep25   0:13 /usr/lib/systemd/systemd rhgb --switched-root
--system --deserialize 31
root           2  0.0  0.0     0     0 ?        S    Sep25   0:00 [kthreadd]
root           3  0.0  0.0     0     0 ?        I<   Sep25   0:00 [rcu_gp]
root           4  0.0  0.0     0     0 ?        I<   Sep25   0:00 [rcu_par_gp]
root           5  0.0  0.0     0     0 ?        I<   Sep25   0:00 [slub_flushwq]
root           6  0.0  0.0     0     0 ?        I<   Sep25   0:00 [netns]
root           8  0.0  0.0     0     0 ?        I<   Sep25   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0     0     0 ?        I<   Sep25   0:00 [mm_percpu_wq]
root          12  0.0  0.0     0     0 ?        I    Sep25   0:00 [rcu_tasks_kthre]
root          13  0.0  0.0     0     0 ?        I    Sep25   0:00 [rcu_tasks_rude_]
root          14  0.0  0.0     0     0 ?        I    Sep25   0:00 [rcu_tasks_trace]

```

root	15	0.0	0.0	0	0 ?	S	Sep25	0:00	[ksoftirqd/0]
root	16	0.0	0.0	0	0 ?	I	Sep25	0:01	[rcu_preempt]
root	17	0.0	0.0	0	0 ?	S	Sep25	0:00	[migration/0]
root	18	0.0	0.0	0	0 ?	S	Sep25	0:00	[idle_inject/0]
root	20	0.0	0.0	0	0 ?	S	Sep25	0:00	[cpuhp/0]
root	21	0.0	0.0	0	0 ?	S	Sep25	0:00	[cpuhp/1]
root	22	0.0	0.0	0	0 ?	S	Sep25	0:00	[idle_inject/1]
root	23	0.0	0.0	0	0 ?	S	Sep25	0:00	[migration/1]
root	24	0.0	0.0	0	0 ?	S	Sep25	0:00	[ksoftirqd/1]
root	27	0.0	0.0	0	0 ?	S	Sep25	0:00	[cpuhp/2]
root	28	0.0	0.0	0	0 ?	S	Sep25	0:00	[idle_inject/2]
root	29	0.0	0.0	0	0 ?	S	Sep25	0:00	[migration/2]
root	30	0.0	0.0	0	0 ?	S	Sep25	0:00	[ksoftirqd/2]
root	32	0.0	0.0	0	0 ?	I<	Sep25	0:00	[kworker/2:0H-events_highpri]
root	33	0.0	0.0	0	0 ?	S	Sep25	0:00	[cpuhp/3]
root	34	0.0	0.0	0	0 ?	S	Sep25	0:00	[idle_inject/3]
root	35	0.0	0.0	0	0 ?	S	Sep25	0:00	[migration/3]
root	36	0.0	0.0	0	0 ?	S	Sep25	0:00	[ksoftirqd/3]
root	38	0.0	0.0	0	0 ?	I<	Sep25	0:00	[kworker/3:0H-events_highpri]
root	42	0.0	0.0	0	0 ?	S	Sep25	0:00	[kdevtmpfs]
root	43	0.0	0.0	0	0 ?	I<	Sep25	0:00	[inet_frag_wq]
root	44	0.0	0.0	0	0 ?	S	Sep25	0:00	[kauditd]
root	45	0.0	0.0	0	0 ?	S	Sep25	0:00	[khungtaskd]
root	46	0.0	0.0	0	0 ?	S	Sep25	0:00	[oom_reaper]
root	47	0.0	0.0	0	0 ?	I<	Sep25	0:00	[writeback]
root	48	0.0	0.0	0	0 ?	S	Sep25	0:02	[kcompactd0]
root	49	0.0	0.0	0	0 ?	SN	Sep25	0:00	[ksmd]
root	50	0.0	0.0	0	0 ?	SN	Sep25	0:00	[khugepaged]
root	51	0.0	0.0	0	0 ?	I<	Sep25	0:00	[cryptd]
root	52	0.0	0.0	0	0 ?	I<	Sep25	0:00	[kintegrityd]
root	53	0.0	0.0	0	0 ?	I<	Sep25	0:00	[kblockd]
root	54	0.0	0.0	0	0 ?	I<	Sep25	0:00	[blkcg_punt_bio]
root	55	0.0	0.0	0	0 ?	I<	Sep25	0:00	[tpm_dev_wq]
root	56	0.0	0.0	0	0 ?	I<	Sep25	0:00	[md]

```

root      57  0.0  0.0    0    0 ?      I<   Sep25  0:00 [md_bitmap]
root      58  0.0  0.0    0    0 ?      I<   Sep25  0:00 [edac-poller]
root      59  0.0  0.0    0    0 ?      S    Sep25  0:00 [watchdogd]
root      61  0.0  0.0    0    0 ?      I<   Sep25  0:00 [kworker/0:1H-kblockd]
root      62  0.0  0.0    0    0 ?      S    Sep25  0:00 [kswapd0]
root      69  0.0  0.0    0    0 ?      I<   Sep25  0:00 [kthrotld]
root      76  0.0  0.0    0    0 ?      I<   Sep25  0:00 [acpi_thermal_pm]
root      77  0.0  0.0    0    0 ?      I<   Sep25  0:00 [kmpath_rdacd]
root      78  0.0  0.0    0    0 ?      I<   Sep25  0:00 [kaluad]
--More--
[q]

```

This output contains some additional information:

<b>USER</b>	The user associated with the process
<b>%CPU</b>	% of the processor resources used by the process
<b>%MEM</b>	% of the memory resources used by the process

The command line switches for this command are :

```
[root@redhat9 ~]# ps --help all
```

Usage:

```
ps [options]
```

Basic options:

```

-A, -e      all processes
-a          all with tty, except session leaders
a           all with tty, including other users
-d          all except session leaders
-N, --deselect  negate selection
r           only running processes
T           all processes on this terminal
x           processes without controlling ttys

```

### Selection by list:

```
-C <command>      command name
-G, --Group <GID>  real group id or name
-g, --group <group> session or effective group name
-p, p, --pid <PID>  process id
    --ppid <PID>   parent process id
-q, q, --quick-pid <PID>
                    process id (quick mode)
-s, --sid <session> session id
-t, t, --tty <tty>  terminal
-u, U, --user <UID> effective user id or name
-U, --User <UID>   real user id or name
```

The selection options take as their argument either:

- a comma-separated list e.g. '-u root,nobody' or
- a blank-separated list e.g. '-p 123 4567'

### Output formats:

```
-F          extra full
-f         full-format, including command lines
 f, --forest  ascii art process tree
-H         show process hierarchy
-j         jobs format
 j         BSD job control format
-l         long format
 l         BSD long format
-M, Z      add security data (for SELinux)
-O <format>  preloaded with default columns
 O <format>  as -O, with BSD personality
-o, o, --format <format>
            user-defined format
 s         signal format
 u         user-oriented format
 v         virtual memory format
```

```
X          register format
-y         do not show flags, show rss vs. addr (used with -l)
--context  display security context (for SELinux)
--headers  repeat header lines, one per page
--no-headers do not print header at all
--cols, --columns, --width <num>
           set screen width
--rows, --lines <num>
           set screen height

Show threads:
H          as if they were processes
-L         possibly with LWP and NLWP columns
-m, m     after processes
-T         possibly with SPID column

Miscellaneous options:
-c         show scheduling class with -l option
c         show true command name
e         show the environment after command
k, --sort specify sort order as: [+|-]key[, [+|-]key[, ...]]
L         show format specifiers
n         display numeric uid and wchan
S, --cumulative include some dead child process data
-y         do not show flags, show rss (only with -l)
-V, V, --version display version information and exit
-w, w     unlimited output width

--help <simple|list|output|threads|misc|all>
        display help and exit
```

For more details see ps(1).

## 1.2 - The pgrep command

The **pgrep** command is used to search for a process based on its name and other properties, then display its PID on the standard output.

For example, the following command displays the PID of the sshd process belonging to root:

```
[root@redhat9 ~]# pgrep -u root sshd
5734
9357
```

The following command displays all the PIDs of processes belonging to root or trainee :

```
[root@redhat9 ~]# pgrep -u root,trainee | more
1
2
3
4
5
6
8
10
12
13
14
15
16
17
18
20
21
22
23
24
```

27  
28  
29  
30  
32  
33  
34  
35  
36  
38  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
61  
62  
69  
76  
77  
78

```
79
--More--
[q]
```

The command line switches for this command are :

```
[root@redhat9 ~]# pgrep --help

Usage:
  pgrep [options] <pattern>

Options:
  -d, --delimiter <string>  specify output delimiter
  -l, --list-name            list PID and process name
  -a, --list-full           list PID and full command line
  -v, --inverse             negates the matching
  -w, --lightweight         list all TID
  -c, --count               count of matching processes
  -f, --full                use full process name to match
  -g, --pgroup <PGID,...>  match listed process group IDs
  -G, --group <GID,...>    match real group IDs
  -i, --ignore-case         match case insensitively
  -n, --newest              select most recently started
  -o, --oldest              select least recently started
  -O, --older <seconds>    select where older than seconds
  -P, --parent <PPID,...>  match only child processes of the given parent
  -s, --session <SID,...>  match session IDs
  -t, --terminal <tty,...> match by controlling terminal
  -u, --euid <ID,...>      match by effective IDs
  -U, --uid <ID,...>       match by real IDs
  -x, --exact               match exactly with the command name
  -F, --pidfile <file>     read PIDs from file
  -L, --logpidfile         fail if PID file is not locked
  -r, --runstates <state>  match runstates [D,S,Z,...]
```



```

daemon├─ibus-dconf—3*[{ibus-dconf}]
├─ibus-engine-sim—2*[{ibus-engine-sim}]
└─2*[{ibus-daemon}]
shell}}
└─17*[{gnome-
settin—3*[{gsd-ally-settin}]
color}}
└─gsd-ally-
└─gsd-color—3*[{gsd-
datetime}}
└─gsd-datetime—3*[{gsd-
housekeepin—3*[{gsd-housekeepin}]
└─gsd-
keyboard}}
└─gsd-keyboard—3*[{gsd-
media-keys}}
└─gsd-media-keys—3*[{gsd-
power}}
└─gsd-power—3*[{gsd-
notif—2*[{gsd-print-notif}]
└─gsd-print-
rfkill}}
└─gsd-rfkill—2*[{gsd-
screensaver—2*[{gsd-screensaver}]
└─gsd-
sharing}}
└─gsd-sharing—3*[{gsd-
smartcard}}
└─gsd-smartcard—5*[{gsd-
sound}}
└─gsd-sound—3*[{gsd-

```



```
└─wpa_supplicant
└─xdg-permission-——2*[{xdg-permission-}]
```

The command line switches for this command are:

```
[root@redhat9 ~]# pstree --help
pstree: unrecognized option '--help'
Usage: pstree [-acglpsStTuZ] [ -h | -H PID ] [ -n | -N type ]
           [ -A | -G | -U ] [ PID | USER ]
  or: pstree -V
```

Display a tree of processes.

```
-a, --arguments      show command line arguments
-A, --ascii          use ASCII line drawing characters
-c, --compact-not    don't compact identical subtrees
-C, --color=TYPE     color process by attribute
                    (age)
-g, --show-pgids     show process group ids; implies -c
-G, --vt100          use VT100 line drawing characters
-h, --highlight-all highlight current process and its ancestors
-H PID, --highlight-pid=PID
                    highlight this process and its ancestors
-l, --long            don't truncate long lines
-n, --numeric-sort   sort output by PID
-N TYPE, --ns-sort=TYPE
                    sort output by this namespace type
                    (cgroup, ipc, mnt, net, pid, time, user, uts)
-p, --show-pids      show PIDs; implies -c
-s, --show-parents   show parents of the selected process
-S, --ns-changes     show namespace transitions
-t, --thread-names   show full thread names
-T, --hide-threads   hide threads, show only processes
-u, --uid-changes    show uid transitions
```

```
-U, --unicode      use UTF-8 (Unicode) line drawing characters
-V, --version      display version information
-Z, --security-context
                   show security attributes
```

```
PID   start at this PID; default is 1 (init)
USER  show only trees rooted at processes of this user
```

## 1.4 - The top command

**top** shows a continuous real time list of running processes:

```
[root@redhat9 ~]# top
```

```
top - 14:41:00 up 2 days, 1:56, 1 user, load average: 0.01, 0.01, 0.00
Tasks: 199 total, 1 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 0.1 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7685.1 total, 4518.6 free, 1105.4 used, 2375.3 buff/cache
MiB Swap: 5120.0 total, 5120.0 free, 0.0 used. 6579.8 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10744	root	20	0	226032	4096	3328	R	0.3	0.1	0:00.04	top
1	root	20	0	190992	17880	10644	S	0.0	0.2	0:13.87	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.05	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthre
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace

15	root	20	0	0	0	0	S	0.0	0.0	0:00.05	ksoftirqd/0
16	root	20	0	0	0	0	I	0.0	0.0	0:01.27	rcu_preempt
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.31	migration/0
18	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
22	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
23	root	rt	0	0	0	0	S	0.0	0.0	0:00.44	migration/1
24	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/1
27	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/2
28	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/2
29	root	rt	0	0	0	0	S	0.0	0.0	0:00.46	migration/2
30	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/2
32	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/2:0H-events_highpri
33	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/3
34	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/3
35	root	rt	0	0	0	0	S	0.0	0.0	0:00.47	migration/3
36	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/3
38	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/3:0H-events_highpri
42	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
43	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	inet_frag_wq
44	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kauditd
45	root	20	0	0	0	0	S	0.0	0.0	0:00.13	khungtaskd
46	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
47	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
48	root	20	0	0	0	0	S	0.0	0.0	0:02.29	kcompactd0
49	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
50	root	39	19	0	0	0	S	0.0	0.0	0:00.93	khugepaged
51	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	cryptd
52	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
53	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
54	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
55	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
56	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md

```

 57 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 md_bitmap
 58 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 edac-poller
 59 root     -51   0      0      0      0 S   0.0   0.0   0:00.00 wat

```

```
...
```

To display the **top** command help, press **h** :

Help for Interactive Commands - procps-ng 3.3.17

Window 1:Def: Cumulative mode Off. System: Delay 3.0 secs; Secure mode Off.

```

Z,B,E,e  Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
l,t,m,I  Toggle: 'l' load avg; 't' task/cpu; 'm' memory; 'I' Irix mode
0,1,2,3,4 Toggle: '0' zeros; '1/2/3' cpu/numa views; '4' cpus two abreast
f,F,X    Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width

```

```

L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
R,H,J,C . Toggle: 'R' Sort; 'H' Threads; 'J' Num justify; 'C' Coordinates
c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
x,y     . Toggle highlights: 'x' sort field; 'y' running tasks
z,b     . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u,U,o,0 . Filter by: 'u'/'U' effective/any user; 'o'/'0' other criteria
n,#,^0 . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'0' other filter(s)
V,v     . Toggle: 'V' forest view; 'v' hide/show forest view children

```

```

k,r      Manipulate tasks: 'k' kill; 'r' renice
d or s   Set update interval
W,Y,!   Write config file 'W'; Inspect other output 'Y'; Combine Cpus '!'
q        Quit

```

( commands shown with '.' require a visible task display window )

Press 'h' or '?' for help with Windows,

Type 'q' or <Esc> to continue





**Important** - To return to the previous display, press **q** or **escape**.

When launched top's refresh rate is 3 seconds. To change this to 1 second, use the **s** key:

```
top - 14:42:15 up 2 days, 1:57, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 199 total, 1 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7685.1 total, 4518.6 free, 1105.4 used, 2375.3 buff/cache
MiB Swap: 5120.0 total, 5120.0 free, 0.0 used. 6579.8 avail Mem
Change delay from 3.0 to 1
...
```

To sort the list by memory usage, press **M** :

```
[root@redhat9 ~]# top
top - 14:43:12 up 2 days, 1:58, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 199 total, 1 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7685.1 total, 4518.6 free, 1105.4 used, 2375.3 buff/cache
MiB Swap: 5120.0 total, 5120.0 free, 0.0 used. 6579.8 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 6049 gdm        20   0 4453852 242772 120572 S   0.0   3.1   0:23.39  gnome-shell
 6617 gdm        20   0  578604  54332  46592 S   0.0   0.7   0:00.05  ibus-x11
 6206 gdm        20   0  166808  45312  38144 S   0.0   0.6   0:00.05  Xwayland
 5707 root       20   0  350204  44404  18884 S   0.0   0.6   0:00.38  firewalld
 6444 gdm        20   0  678348  33608  21888 S   0.0   0.4   0:00.44  gsd-color
 5675 polkitd   20   0 2713852  31804  19792 S   0.0   0.4   0:00.54  polkitd
 6462 gdm        20   0  661452  29948  22276 S   0.0   0.4   0:00.13  gsd-media-keys
 6674 gdm        20   0 2866068  28596  21120 S   0.0   0.4   0:00.06  gjs
 6486 gdm        20   0  665628  28568  20996 S   0.0   0.4   0:00.12  gsd-power
 6432 gdm        20   0 2931604  27964  20864 S   0.0   0.4   0:00.06  gjs
 6439 gdm        20   0  590924  27396  19716 S   0.0   0.3   0:00.12  gsd-wacom
```

6451	gdm	20	0	590224	26680	19584	S	0.0	0.3	0:00.10	gsd-keyboard
6571	gdm	20	0	598424	24032	17280	S	0.0	0.3	0:00.02	gsd-printer
853	root	20	0	475084	23056	18120	S	0.0	0.3	0:04.02	NetworkManager
5975	gdm	20	0	734740	22260	15744	S	0.0	0.3	0:00.10	gnome-session-b
7620	apache	20	0	2423380	21372	5632	S	0.0	0.3	0:14.02	httpd
7431	apache	20	0	2423380	21316	5632	S	0.0	0.3	0:14.10	httpd
6455	gdm	20	0	594384	20668	16128	S	0.0	0.3	0:00.04	gsd-datetime
7898	apache	20	0	2357844	19356	5632	S	0.0	0.2	0:13.86	httpd
1	root	20	0	190992	17880	10644	S	0.0	0.2	0:13.89	systemd
7621	apache	20	0	2226708	17344	5760	S	0.0	0.2	0:12.83	httpd
7899	apache	20	0	2226708	17292	5760	S	0.0	0.2	0:11.94	httpd
7432	apache	20	0	2226708	17240	5632	S	0.0	0.2	0:12.81	httpd
7622	apache	20	0	2161172	17236	5760	S	0.0	0.2	0:12.70	httpd
7433	apache	20	0	2226708	17196	5632	S	0.0	0.2	0:12.77	httpd
7900	apache	20	0	2226708	17096	5632	S	0.0	0.2	0:12.27	httpd
6381	gdm	9	-11	545148	16128	12800	S	0.0	0.2	0:00.05	wireplumber
5742	root	20	0	395124	15672	11636	S	0.0	0.2	0:00.13	udisksd
6452	gdm	20	0	467300	14876	11136	S	0.0	0.2	0:00.02	gsd-print-notif
6640	colord	20	0	530424	14164	9980	S	0.0	0.2	0:00.06	colord
5851	gdm	20	0	22976	13936	10752	S	0.0	0.2	0:00.11	systemd
9365	trainee	20	0	22984	13812	10624	S	0.0	0.2	0:00.09	systemd
6454	gdm	20	0	603444	13636	10112	S	0.0	0.2	0:04.03	gsd-smartcard
6477	gdm	20	0	530684	13456	8320	S	0.0	0.2	0:00.01	gsd-sound
6599	gdm	20	0	600368	13304	6656	S	0.0	0.2	0:00.07	ibus-daemon
629	root	20	0	36112	13120	9216	S	0.0	0.2	0:00.48	systemd-udev
5787	root	20	0	387252	13092	9472	S	0.0	0.2	0:00.02	gdm-session-wor
6436	gdm	20	0	678244	12720	9216	S	0.0	0.2	0:00.01	gsd-sharing
5704	root	20	0	391468	12632	10712	S	0.0	0.2	0:00.05	ModemManager
6482	gdm	20	0	524704	12100	7168	S	0.0	0.2	0:01.08	gsd-housekeepin
7429	root	20	0	20368	11608	9176	S	0.0	0.1	0:04.09	httpd
7618	root	20	0	20368	11540	9108	S	0.0	0.1	0:04.28	httpd
6377	gdm	9	-11	326096	11520	8192	S	0.0	0.1	0:00.02	pipewire
5671	root	20	0	21044	11480	9088	S	0.0	0.1	0:02.60	systemd-logind
7896	root	20	0	20368	11464	9032	S	0.0	0.1	0:04.17	httpd

```

 9357 root      20   0   19432   11264   9600 S   0.0   0.1   0:00.02 sshd
 5736 root      20   0  247824   11136   9088 S   0.0   0.1   0:00.04 cupsd
  614 root      20   0   27084   10880   9728 S   0.0   0.1   0:00.88 systemd-journal
 5765 root      20   0  453120   10828   7808 S   0.0   0.1   0:00.02 gdm
...

```

To see the zombie and waiting processes, use the **i** key:

```

[root@redhat9 ~]# top
top - 14:44:10 up 2 days, 1:59, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 199 total, 1 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.5 us, 0.0 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.2 hi, 0.0 si, 0.0 st
MiB Mem : 7685.1 total, 4518.6 free, 1105.4 used, 2375.3 buff/cache
MiB Swap: 5120.0 total, 5120.0 free, 0.0 used. 6579.8 avail Mem

   PID USER      PR  NI   VIRT   RES    SHR S  %CPU  %MEM    TIME+  COMMAND
   6049 gdm       20   0 4458012 242772 120572 S   1.0   3.1   0:23.41 gnome-shell
  10744 root      20   0  226032   4096   3328 R   1.0   0.1   0:00.59 top

```

To exit top, press **q**.

The command line switches for this command are :

```

[root@redhat9 ~]# top --help
top: inappropriate '-help'
Usage:
  top -hv | -bcEeHiOSs1 -d secs -n max -u|U user -p pid(s) -o field -w [cols]

```

## 1.5 - The fg, bg and jobs commands

Normally commands are executed in the foreground of a terminal. However you can also execute a command in what is known as the background by adding **(space)&** to the end of the command:

```
# sleep 9999 &
```



**Important** - Note that a background process is said to be **asynchronous** because it continues independently of its parent, which is the shell. The foreground the process is said to be **synchronous**.

Linux identifies processes sent to the back ground by **job numbers**.

The **jobs** command shows a list of all the current jobs associated with the current terminal:

```
[root@redhat9 ~]# sleep 9999 &
[1] 10749

[root@redhat9 ~]# jobs -l
[1]+ 10749 Running                  sleep 9999 &
```



**Important** - The job number is between square brackets whilst the PID is not. The + sign indicates that this is the last job to have been modified.

If you wish to send a process into the background to free up the shell for other commands, you must first suspend the process in question. Normally we suspend a process using the key combination CtrlZ.

For example:

```
[root@redhat9 ~]# sleep 1234
^Z
[2]+  Stopped                  sleep 1234
```

Once suspended, use the **bg** (background) command followed by % and the job number to send the process to the background:

```
[root@redhat9 ~]# bg %2
[2]+ sleep 1234 &

[root@redhat9 ~]# jobs -l
[1]- 10749 Running          sleep 9999 &
[2]+ 10750 Running          sleep 1234 &
```



**Important** - Note that when the process is sent to the background it resumes normal execution. The - character which follows the [1] indicates that this is that last but one job to have been modified.

To bring the job back to the foreground you cannot suspend it by using the `CtrlZ` keys. In this case you will have to send a signal to the process. Signals are sent to processes by using the **kill** command:

```
[root@redhat9 ~]# kill -s stop %2

[2]+  Stopped                  sleep 1234
```

To bring the process back to the foreground, use the `fg` command:

```
[root@redhat9 ~]# fg %2
sleep 1234
^C
```



**Important** - Note that we have used the `CtrlC` keys to kill the process once in the foreground.

The command line switches for the jobs command are :

```
[root@redhat9 ~]# help jobs
jobs: jobs [-lnprs] [jobspec ...] or jobs -x command [args]
  Display status of jobs.
  Lists the active jobs. JOBSPEC restricts output to that job.
  Without options, the status of all active jobs is displayed.
  Options:
    -l      lists process IDs in addition to the normal information
    -n      lists only processes that have changed status since the last
            notification
    -p      lists process IDs only
    -r      restrict output to running jobs
    -s      restrict output to stopped jobs
  If -x is supplied, COMMAND is run after all job specifications that
  appear in ARGS have been replaced with the process ID of that job's
  process group leader.
  Exit Status:
  Returns success unless an invalid option is given or an error occurs.
  If -x is used, returns the exit status of COMMAND.
```

## 1.6 - The wait command

The **wait** command makes an asynchronous process react like a synchronous process:

```
[root@redhat9 ~]# jobs -l
[1]+ 10749 Running                  sleep 9999 &

[root@redhat9 ~]# wait %1
^C

[root@redhat9 ~]# jobs -l
```

```
[1]+ 10749 Running                sleep 9999 &
```



**Important** - Note that using `CtrlC` kills the process generated by the **wait** command and not the process generated by the **sleep** command.

## 1.7 - The nice command

This command is used to change the process priority. The default nice value when a process is started is **10**. The highest value of nice is **-20**, whilst the lowest value is **19**:

```
[root@redhat9 ~]# nice -n -20 sleep 1234
^Z
[2]+  Stopped                  nice -n -20 sleep 1234

[root@redhat9 ~]# ps lx | grep sleep
0    0    10749    10066    20    0  220952    1792  hrtime S    pts/0    0:00  sleep 9999
4    0    10775    10066    0 -20  220952    1792  do_sig T<   pts/0    0:00  sleep 1234
0    0    10777    10066    20    0  221664    2304  pipe_r S+   pts/0    0:00  grep --color=auto sleep

[root@redhat9 ~]# nice -n 19 sleep 5678
^Z
[3]+  Stopped                  nice -n 19 sleep 5678

[root@redhat9 ~]# ps lx | grep sleep
0    0    10749    10066    20    0  220952    1792  hrtime S    pts/0    0:00  sleep 9999
4    0    10775    10066    0 -20  220952    1792  do_sig T<   pts/0    0:00  sleep 1234
0    0    10778    10066    39  19  220952    1792  do_sig TN   pts/0    0:00  sleep 5678
0    0    10780    10066    20    0  221664    2304  pipe_r S+   pts/0    0:00  grep --color=auto sleep
```

As you can see the 6th column contains the nice value that is applied to the 5th column.



**Important** - Note that only root can give a process a value between 0 and -20.

The command line switches for this command are :

```
[root@redhat9 ~]# nice --help
Usage: nice [OPTION] [COMMAND [ARG]...]
Run COMMAND with an adjusted niceness, which affects process scheduling.
With no COMMAND, print the current niceness. Niceness values range from
-20 (most favorable to the process) to 19 (least favorable to the process).

Mandatory arguments to long options are mandatory for short options too.
  -n, --adjustment=N  add integer N to the niceness (default 10)
  --help             display this help and exit
  --version          output version information and exit

NOTE: your shell may have its own version of nice, which usually supersedes
the version described here. Please refer to your shell's documentation
for details about the options it supports.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/nice>
or available locally via: info '(coreutils) nice invocation'
```

## 1.8 - The renice command

This command is used to change the process priority of an already running process. Only the process owner and root can renice a process:

```
[root@redhat9 ~]# jobs -l
[1]  10749 Running          sleep 9999 &
[2]- 10775 Stopped          nice -n -20 sleep 1234
```

```
[3]+ 10778 Stopped                nice -n 19 sleep 5678

[root@redhat9 ~]# bg %2
[2]-  nice -n -20 sleep 1234 &

[root@redhat9 ~]# bg %3
[3]+  nice -n 19 sleep 5678 &

[root@redhat9 ~]# jobs -l
[1]  10749 Running                sleep 9999 &
[2]- 10775 Running                nice -n -20 sleep 1234 &
[3]+ 10778 Running                nice -n 19 sleep 5678 &

[root@redhat9 ~]# renice +5 10775
10775 (process ID) old priority -20, new priority 5

[root@redhat9 ~]# renice -5 10778
10778 (process ID) old priority 19, new priority -5

[root@redhat9 ~]# ps lx | grep sleep
0    0    10749    10066    20    0 220952    1792 hrtime S    pts/0    0:00 sleep 9999
4    0    10775    10066    25    5 220952    1792 do_sys SN   pts/0    0:00 sleep 1234
0    0    10778    10066    15   -5 220952    1792 do_sys S<  pts/0    0:00 sleep 5678
0    0    10790    10066    20    0 221796    2304 pipe_r S+   pts/0    0:00 grep --color=auto sleep
```



**Important** -Note that only root can decrease the priority value with the renice command.

The command line switches for this command are :

```
[root@redhat9 ~]# renice --help
```

**Usage:**

```
renice [-n] <priority> [-p|--pid] <pid>...
renice [-n] <priority> -g|--pgrp <pgid>...
renice [-n] <priority> -u|--user <user>...
```

Alter the priority of running processes.

**Options:**

```
-n, --priority <num>    specify the nice value
-p, --pid                interpret arguments as process ID (default)
-g, --pgrp               interpret arguments as process group ID
-u, --user                interpret arguments as username or user ID

-h, --help                display this help
-V, --version             display version
```

For more details see `renice(1)`.

## 1.9 - The nohup command

This command allows a process to continue after closing the terminal in which it was launched. A child process dies when the parent process dies or terminates. As a connection is a process, when you disconnect, your processes terminate. To avoid remaining connected after launching a long process, use the `nohup` command:

```
nohup lp sales.txt &
```

The command line switches for this command are :

```
[root@redhat9 ~]# nohup --help
Usage: nohup COMMAND [ARG]...
       or: nohup OPTION
Run COMMAND, ignoring hangup signals.
```

```
--help      display this help and exit
--version   output version information and exit
```

If standard input is a terminal, redirect it from an unreadable file.  
If standard output is a terminal, append output to 'nohup.out' if possible,  
'\$HOME/nohup.out' otherwise.  
If standard error is a terminal, redirect it to standard output.  
To save output to FILE, use 'nohup COMMAND > FILE'.

NOTE: your shell may have its own version of nohup, which usually supersedes  
the version described here. Please refer to your shell's documentation  
for details about the options it supports.

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>  
Full documentation <<https://www.gnu.org/software/coreutils/nohup>>  
or available locally via: info '(coreutils) nohup invocation'

## 1.10 - The kill command

The kill command sends signals to processes. The list of possible signals can be displayed using the **-l** option:

```
[root@redhat9 ~]# 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



**Important** - You will notice that each signal has a number. These signal numbers are used instead of options. For example, **-19** instead of the **-stop** option.

The most useful signals are as follows:

Numéro	Description
-1	A Hang Up is sent to all the child processus of the PID specified
-2	Equivalent to the <code>CtrlC</code> key combination
-3	The same signal as -2 but it generates de debug file
-9	A brutal way of killing a process
-15	The correct way of killing a process

The command line switches for this command are :

```
[root@redhat9 ~]# help kill
kill: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]
Send a signal to a job.
Send the processes identified by PID or JOBSPEC the signal named by
SIGSPEC or SIGNUM. If neither SIGSPEC nor SIGNUM is present, then
SIGTERM is assumed.
Options:
  -s sig    SIG is a signal name
  -n sig    SIG is a signal number
  -l        list the signal names; if arguments follow '-l' they are
            assumed to be signal numbers for which names should be listed
  -L        synonym for -l
Kill is a shell builtin for two reasons: it allows job IDs to be used
instead of process IDs, and allows processes to be killed if the limit
```

on processes that you can create is reached.  
Exit Status:  
Returns success unless an invalid option is given or an error occurs.

## 1.11 - The pkill command

The pkill command is used to send signals to processes identified by their name. For example, the following command forces syslog to reread its configuration file:

```
[root@redhat9 ~]# pkill -HUP rsyslogd
```

The command line switches for this command are :

```
[root@redhat9 ~]# pkill --help
```

### Usage:

```
pkill [options] <pattern>
```

### Options:

-<sig>, --signal <sig>	signal to send (either number or name)
-q, --queue <value>	integer value to be sent with the signal
-e, --echo	display what is killed
-c, --count	count of matching processes
-f, --full	use full process name to match
-g, --pgroup <PGID,...>	match listed process group IDs
-G, --group <GID,...>	match real group IDs
-i, --ignore-case	match case insensitively
-n, --newest	select most recently started
-o, --oldest	select least recently started
-O, --older <seconds>	select where older than seconds
-P, --parent <PPID,...>	match only child processes of the given parent
-s, --session <SID,...>	match session IDs
-t, --terminal <tty,...>	match by controlling terminal

```
-u, --euid <ID,...>    match by effective IDs
-U, --uid <ID,...>     match by real IDs
-x, --exact            match exactly with the command name
-F, --pidfile <file>   read PIDs from file
-L, --logpidfile       fail if PID file is not locked
-r, --runstates <state> match runstates [D,S,Z,...]
--ns <PID>            match the processes that belong to the same
                    namespace as <pid>
--nslist <ns,...>     list which namespaces will be considered for
                    the --ns option.
                    Available namespaces: ipc, mnt, net, pid, user, uts

-h, --help            display this help and exit
-V, --version         output version information and exit
```

For more details see `pgrep(1)`.