Experiment No # 05

Experiment Name # Process handling in Linux

Aim and Objects:

The Linux terminal has a number of useful commands that can display running processes, kill them, and change their priority level. An instance of a program is called Process. In this lab we will be able to how to handle process, kill process etc.

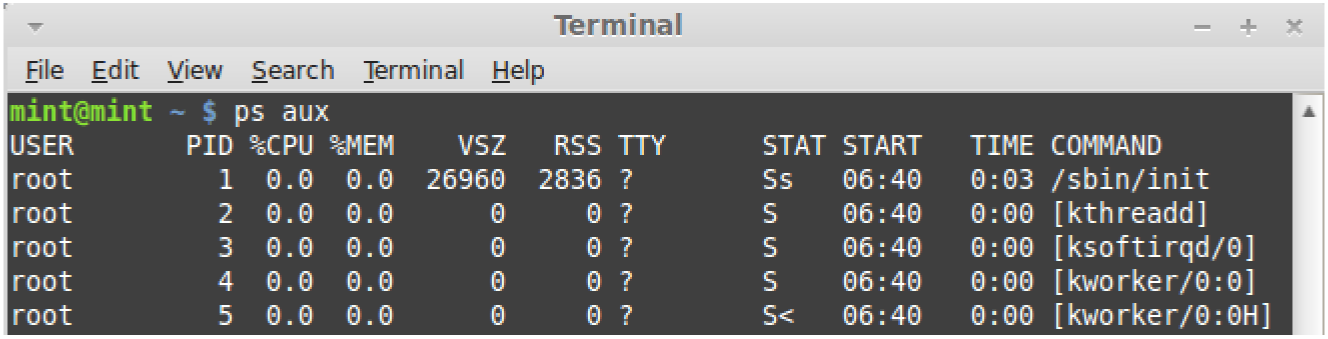
Commands Use for Process Handling :

The commands for handling process in Linux is given below :

1) Ps

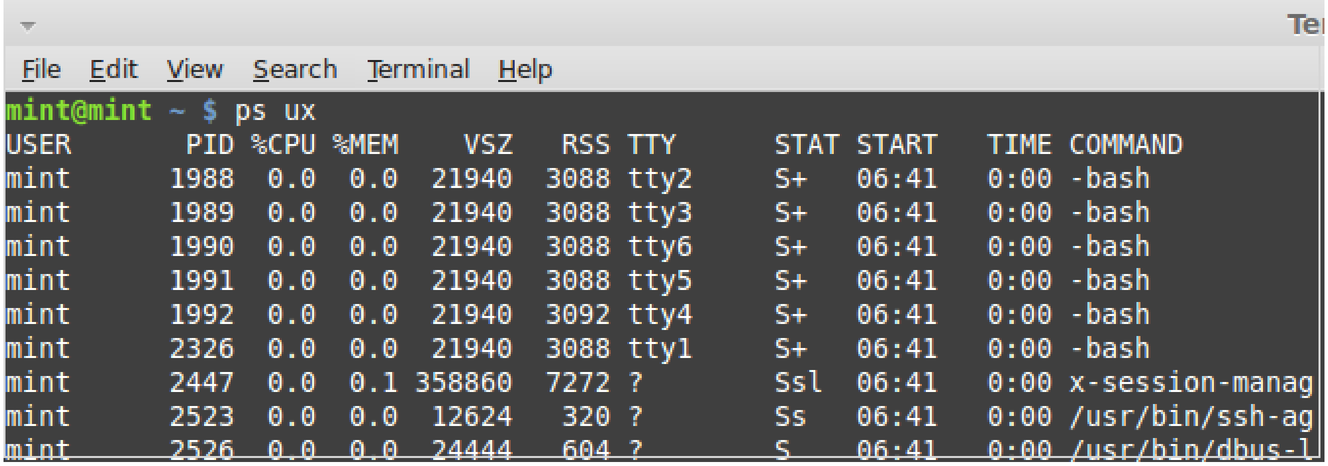
This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different.
To check all the processes running under a user, use the command -

i. **ps aux :** To display the owner of the processes along with the processes



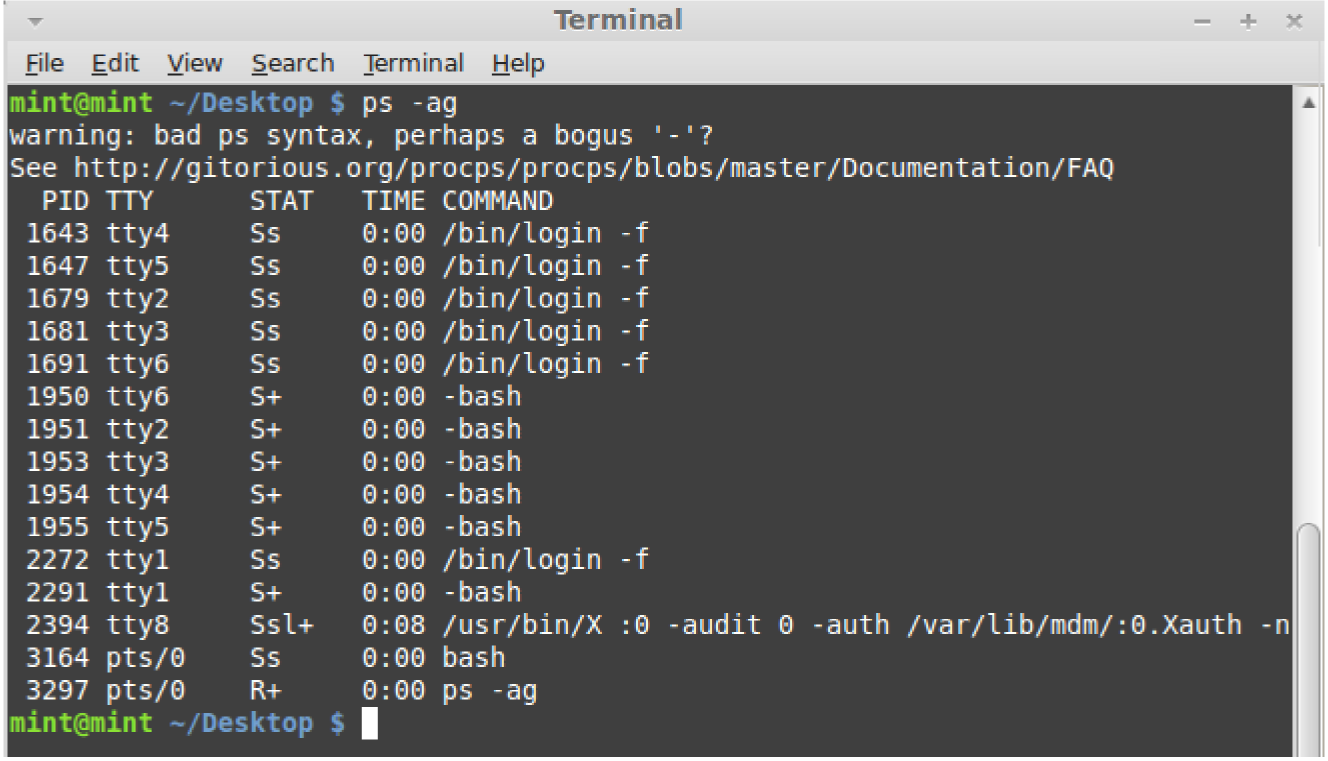
```
mint@mint ~ $ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  26960  2836 ?        Ss   06:40   0:03 /sbin/init
root         2  0.0  0.0      0     0 ?        S    06:40   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    06:40   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S    06:40   0:00 [kworker/0:0]
root         5  0.0  0.0      0     0 ?        S<   06:40   0:00 [kworker/0:0H]
```

ii. **ps ux :**



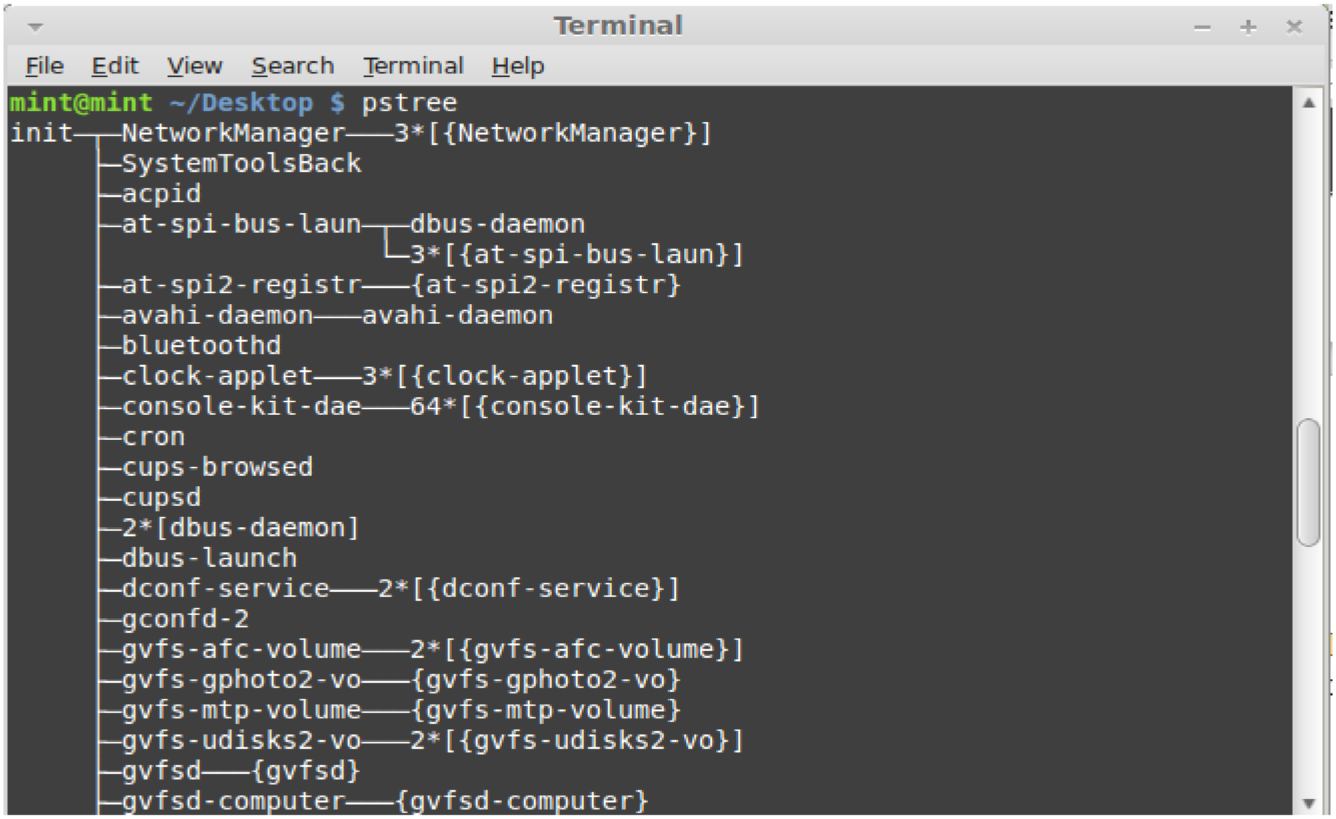
```
mint@mint ~ $ ps ux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
mint      1988  0.0  0.0  21940  3088 tty2     S+   06:41   0:00 -bash
mint      1989  0.0  0.0  21940  3088 tty3     S+   06:41   0:00 -bash
mint      1990  0.0  0.0  21940  3088 tty6     S+   06:41   0:00 -bash
mint      1991  0.0  0.0  21940  3088 tty5     S+   06:41   0:00 -bash
mint      1992  0.0  0.0  21940  3092 tty4     S+   06:41   0:00 -bash
mint      2326  0.0  0.0  21940  3088 tty1     S+   06:41   0:00 -bash
mint      2447  0.0  0.1 358860  7272 ?        Ssl  06:41   0:00 x-session-manag
mint      2523  0.0  0.0  12624   320 ?        Ss   06:41   0:00 /usr/bin/ssh-ag
mint      2526  0.0  0.0  24444   604 ?        S    06:41   0:00 /usr/bin/dbus-l
```

iii. ps –ag : To get information about all running process



```
mint@mint ~/Desktop $ ps -ag
warning: bad ps syntax, perhaps a bogus '-'?
See http://gitorious.org/procps/procps/blobs/master/Documentation/FAQ
  PID TTY      STAT   TIME COMMAND
 1643 tty4     Ss     0:00 /bin/login -f
 1647 tty5     Ss     0:00 /bin/login -f
 1679 tty2     Ss     0:00 /bin/login -f
 1681 tty3     Ss     0:00 /bin/login -f
 1691 tty6     Ss     0:00 /bin/login -f
 1950 tty6     S+     0:00 -bash
 1951 tty2     S+     0:00 -bash
 1953 tty3     S+     0:00 -bash
 1954 tty4     S+     0:00 -bash
 1955 tty5     S+     0:00 -bash
 2272 tty1     Ss     0:00 /bin/login -f
 2291 tty1     S+     0:00 -bash
 2394 tty8     Ssl+   0:08 /usr/bin/X :0 -audit 0 -auth /var/lib/mdm/:0.Xauth -n
 3164 pts/0    Ss     0:00 bash
 3297 pts/0    R+     0:00 ps -ag
mint@mint ~/Desktop $ 
```

iv. pstree : The **pstree** command is another way of visualizing processes. It displays them in tree format.



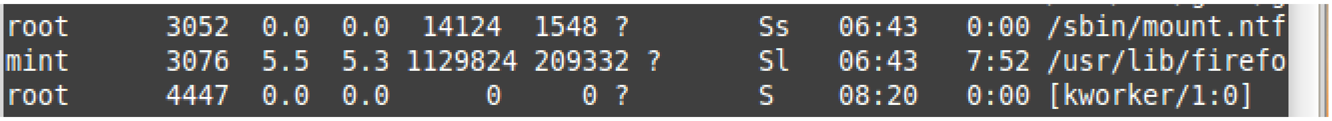
```
mint@mint ~/Desktop $ pstree
init─┬─NetworkManager───3*[{NetworkManager}]
     ├─SystemToolsBack
     ├─acpid
     ├─at-spi-bus-laun─┬─dbus-daemon
     │                 └─3*[{at-spi-bus-laun}]
     ├─at-spi2-registr───{at-spi2-registr}
     ├─avahi-daemon───avahi-daemon
     ├─bluetoothd
     ├─clock-applet───3*[{clock-applet}]
     ├─console-kit-dae───64*[{console-kit-dae}]
     ├─cron
     ├─cups-browsed
     ├─cupsd
     ├─2*[dbus-daemon]
     ├─dbus-launch
     ├─dconf-service───2*[{dconf-service}]
     ├─gconfd-2
     ├─gvfs-afc-volume───2*[{gvfs-afc-volume}]
     ├─gvfs-gphoto2-vo───{gvfs-gphoto2-vo}
     ├─gvfs-mtp-volume───{gvfs-mtp-volume}
     ├─gvfs-udisks2-vo───2*[{gvfs-udisks2-vo}]
     ├─gvfsd───{gvfsd}
     ├─gvfsd-computer───{gvfsd-computer}
```

2) Kill

The kill command can kill a process. This command terminates running processes on a Linux machine. To use these utilities I need to know the PID (process id) of the process I want to kill.
Example :

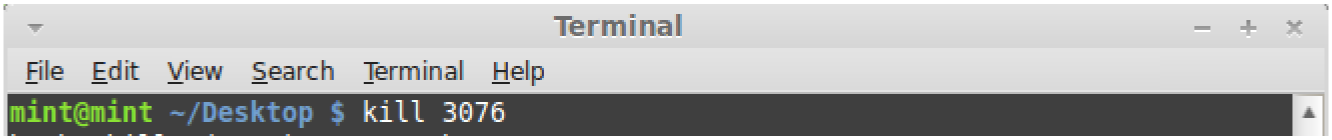
Kill PID:

Before Kill Command :



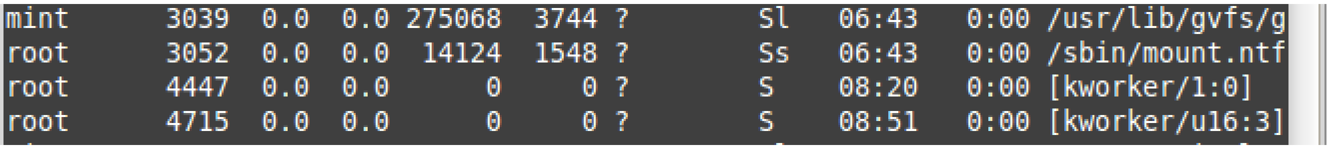
```
root      3052  0.0  0.0  14124  1548 ?        Ss   06:43   0:00 /sbin/mount.ntf
mint      3076  5.5  5.3 1129824 209332 ?      Sl   06:43   7:52 /usr/lib/firefo
root      4447  0.0  0.0      0     0 ?        S    08:20   0:00 [kworker/1:0]
```

Kill Command:



```
mint@mint ~/Desktop $ kill 3076
```

After Kill Command:

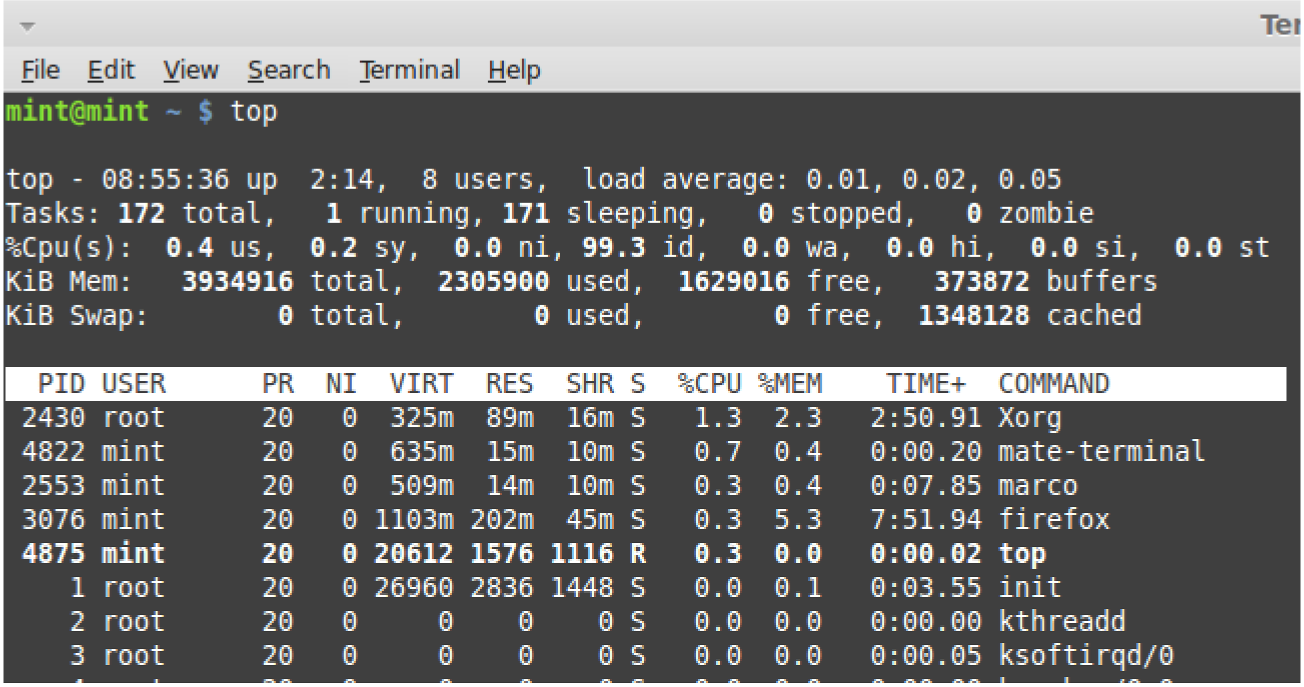


```
mint      3039  0.0  0.0 275068  3744 ?        Sl   06:43   0:00 /usr/lib/gvfs/g
root      3052  0.0  0.0  14124  1548 ?        Ss   06:43   0:00 /sbin/mount.ntf
root      4447  0.0  0.0      0     0 ?        S    08:20   0:00 [kworker/1:0]
root      4715  0.0  0.0      0     0 ?        S    08:51   0:00 [kworker/u16:3]
```

3) Top

This utility tells the user about all the running processes on the Linux machine

Example :

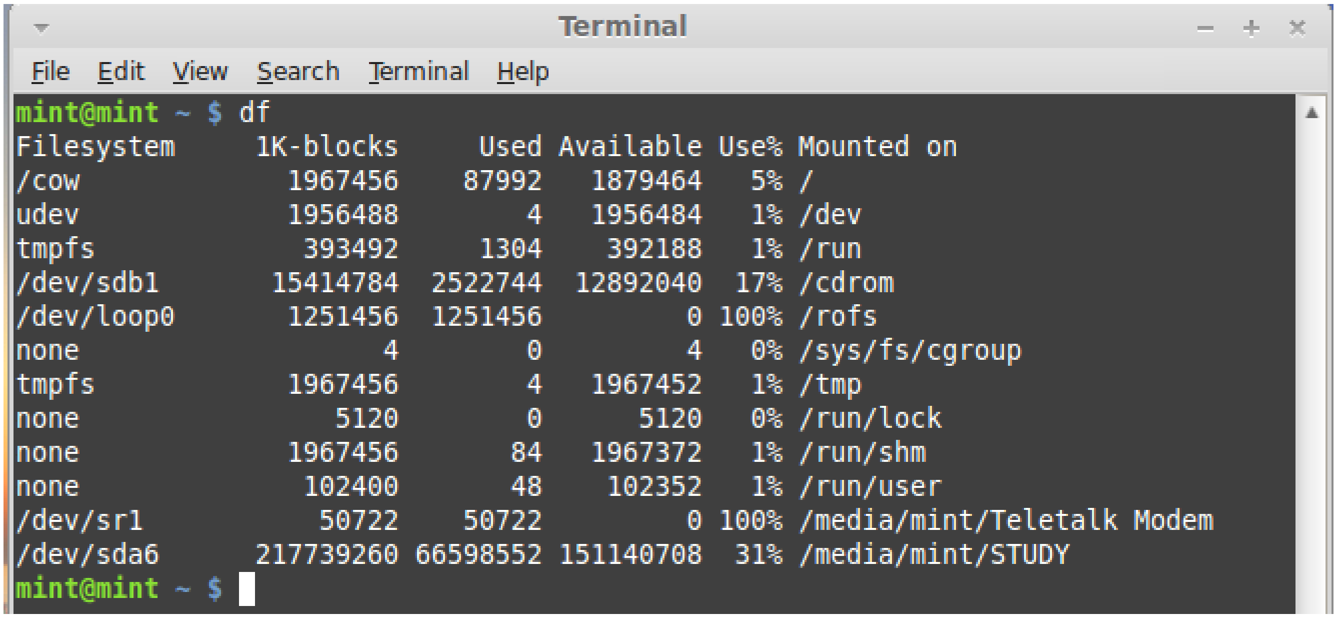


```
mint@mint ~ $ top

top - 08:55:36 up  2:14,  8 users,  load average: 0.01, 0.02, 0.05
Tasks: 172 total,   1 running, 171 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.4 us,  0.2 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   3934916 total,  2305900 used,  1629016 free,   373872 buffers
KiB Swap:        0 total,        0 used,        0 free,  1348128 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 2430 root      20   0  325m  89m  16m S   1.3  2.3   2:50.91 Xorg
 4822 mint      20   0  635m  15m  10m S   0.7  0.4   0:00.20 mate-terminal
 2553 mint      20   0  509m  14m  10m S   0.3  0.4   0:07.85 marco
 3076 mint      20   0 1103m 202m  45m S   0.3  5.3   7:51.94 firefox
 4875 mint      20   0 20612 1576 1116 R   0.3  0.0   0:00.02 top
    1 root      20   0 26960 2836 1448 S   0.0  0.1   0:03.55 init
    2 root      20   0     0    0    0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S   0.0  0.0   0:00.05 ksoftirqd/0
```

4) df

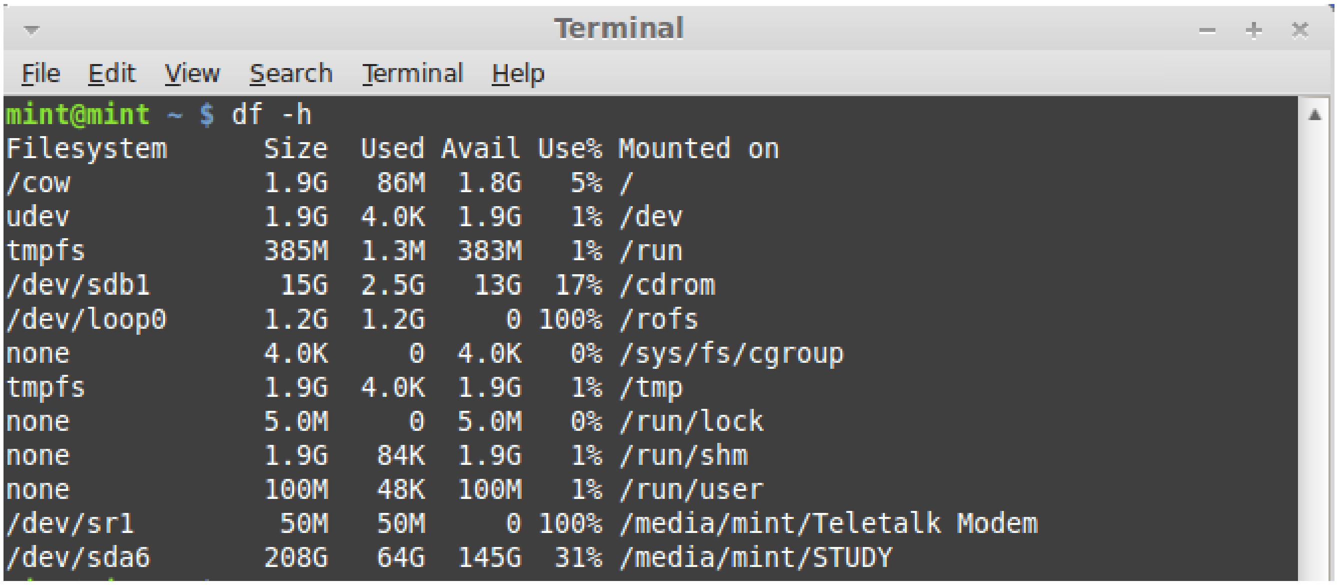
This utility reports the free disk space(Hard Disk) on all the file systems.

Example :



```
mint@mint ~ $ df
Filesystem     1K-blocks     Used Available Use% Mounted on
/cow             1967456    87992   1879464   5% /
udev             1956488        4   1956484   1% /dev
tmpfs             393492     1304    392188   1% /run
/dev/sdb1       15414784  2522744  12892040  17% /cdrom
/dev/loop0       1251456  1251456         0 100% /rofs
none                   4        0         4   0% /sys/fs/cgroup
tmpfs            1967456        4   1967452   1% /tmp
none                5120        0      5120   0% /run/lock
none             1967456       84   1967372   1% /run/shm
none              102400       48    102352   1% /run/user
/dev/sr1           50722    50722         0 100% /media/mint/Teletalk Modem
/dev/sda6      217739260 66598552 151140708  31% /media/mint/STUDY
mint@mint ~ $
```

If we want the above information in a readable format, then use the command : df –h

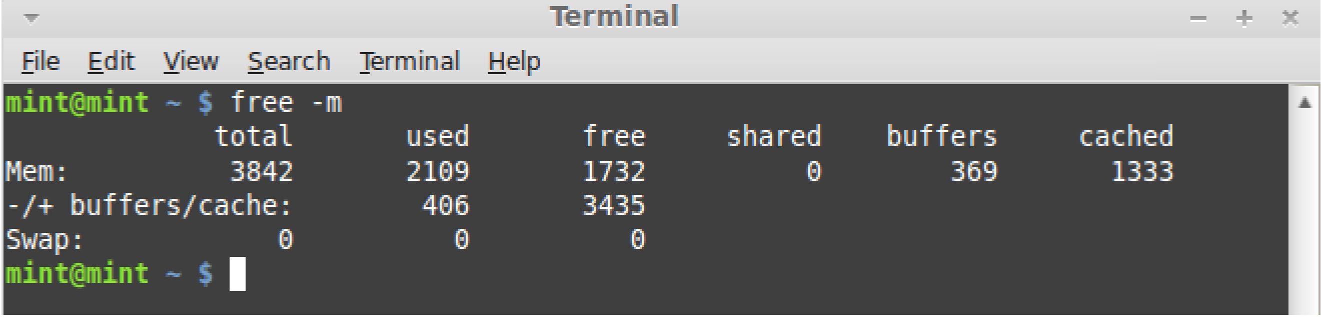


```
mint@mint ~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/cow            1.9G   86M  1.8G   5% /
udev            1.9G  4.0K  1.9G   1% /dev
tmpfs           385M  1.3M  383M   1% /run
/dev/sdb1        15G  2.5G   13G  17% /cdrom
/dev/loop0      1.2G  1.2G     0 100% /rofs
none            4.0K     0  4.0K   0% /sys/fs/cgroup
tmpfs           1.9G  4.0K  1.9G   1% /tmp
none            5.0M     0  5.0M   0% /run/lock
none            1.9G   84K  1.9G   1% /run/shm
none            100M   48K  100M   1% /run/user
/dev/sr1         50M   50M     0 100% /media/mint/Teletalk Modem
/dev/sda6       208G   64G  145G  31% /media/mint/STUDY
```

5) Free

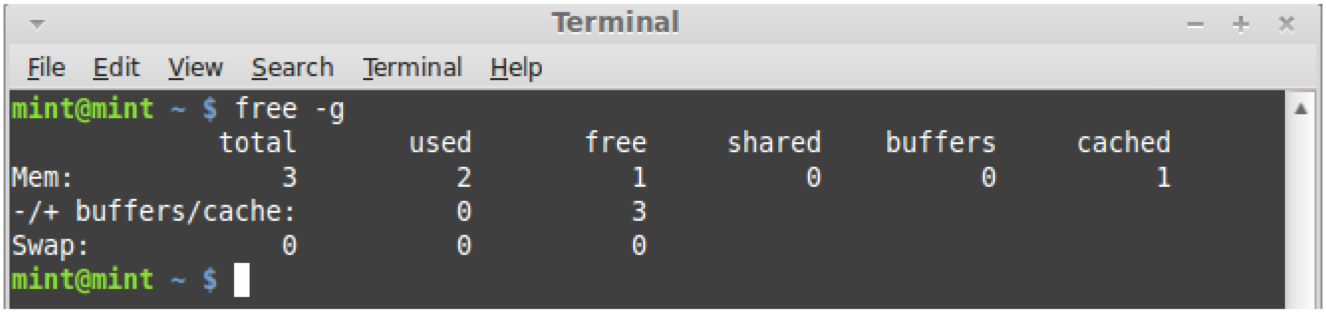
This command shows the free and used memory (RAM) on the Linux system.

free –m : to display output in MB



```
mint@mint ~ $ free -m
             total       used       free     shared    buffers     cached
Mem:          3842       2109       1732          0        369       1333
-/+ buffers/cache:        406       3435
Swap:            0          0          0
mint@mint ~ $ 
```

free –g : to display output in GB



```
mint@mint ~ $ free -g
             total       used       free     shared    buffers     cached
Mem:             3          2          1          0          0          1
-/+ buffers/cache:          0          3
Swap:            0          0          0
mint@mint ~ $ 
```

Conclusion :

The ps command on linux is one of the most basic commands for viewing the processes running on the system. It provides a snapshot of the current processes along with detailed information like user id, cpu usage, memory usage, command name etc. It does not display data in real time like top or htop commands. But even though being simpler in features and output it is still an essential process management/monitoring tool that every linux newbie should know about and learn well.