

# LINUX ADMIN 1

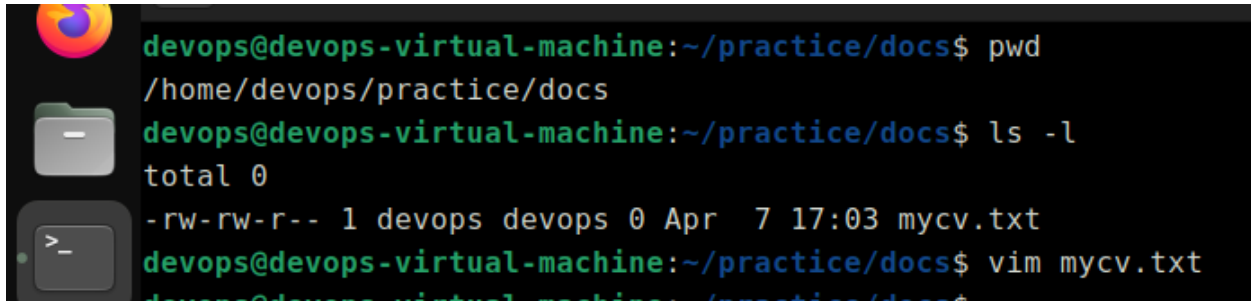
LAB 3  
EDITORS  
PROCESSES



**PREPARED FOR**  
Ahmed Ibrahim  
**PREPARED BY**  
Islam Khaled

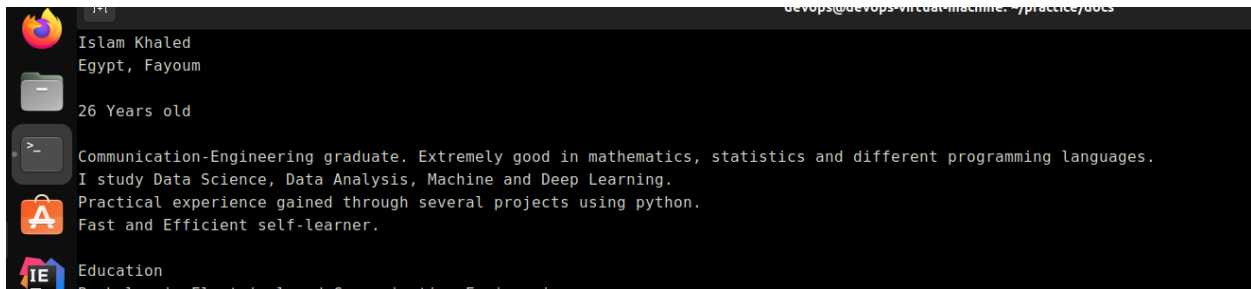
**14 Abril 2023**

**1. Using vi, write your CV in the file mycv. Your CV should include your name, age, school, college, experience,...**



```
devops@devops-virtual-machine:~/practice/docs$ pwd
/home/devops/practice/docs
devops@devops-virtual-machine:~/practice/docs$ ls -l
total 0
-rw-rw-r-- 1 devops devops 0 Apr  7 17:03 mycv.txt
devops@devops-virtual-machine:~/practice/docs$ vim mycv.txt
devops@devops-virtual-machine:~/practice/docs$
```

**2. Open mycv file using vi command then: Without using arrows state how to:**



```
Islam Khaled
Egypt, Fayoum

26 Years old

Communication-Engineering graduate. Extremely good in mathematics, statistics and different programming languages.
I study Data Science, Data Analysis, Machine and Deep Learning.
Practical experience gained through several projects using python.
Fast and Efficient self-learner.

Education
Bachelor in Electrical and Communication Engineering
```

**a. Move the cursor down one line at a time.**

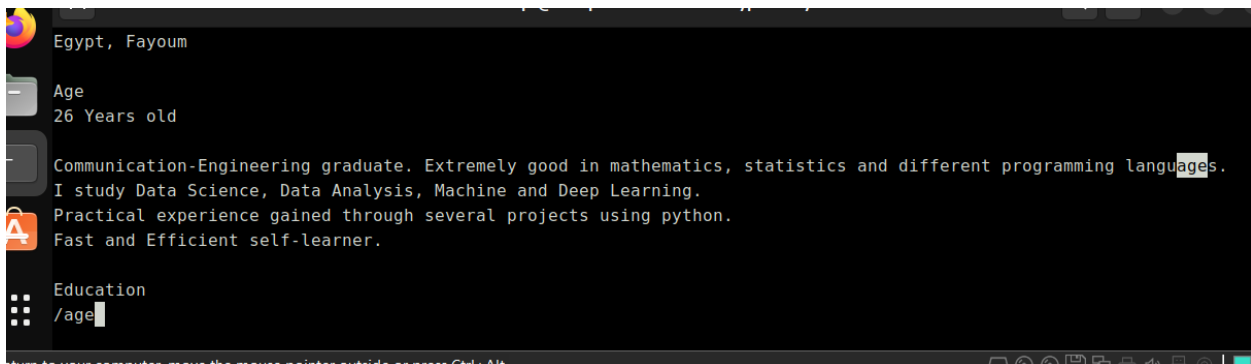
In command more : j

**b. Move the cursor up one line at a time.**

In command mode : k

**c. Search for word age**

In command mode : /age



```
Egypt, Fayoum

Age
26 Years old

Communication-Engineering graduate. Extremely good in mathematics, statistics and different programming languages.
I study Data Science, Data Analysis, Machine and Deep Learning.
Practical experience gained through several projects using python.
Fast and Efficient self-learner.

Education
/age
```

**d. Step to line 5 (assuming that you are in line 1 and the file is more than 5 lines).**

In command mode : 5G

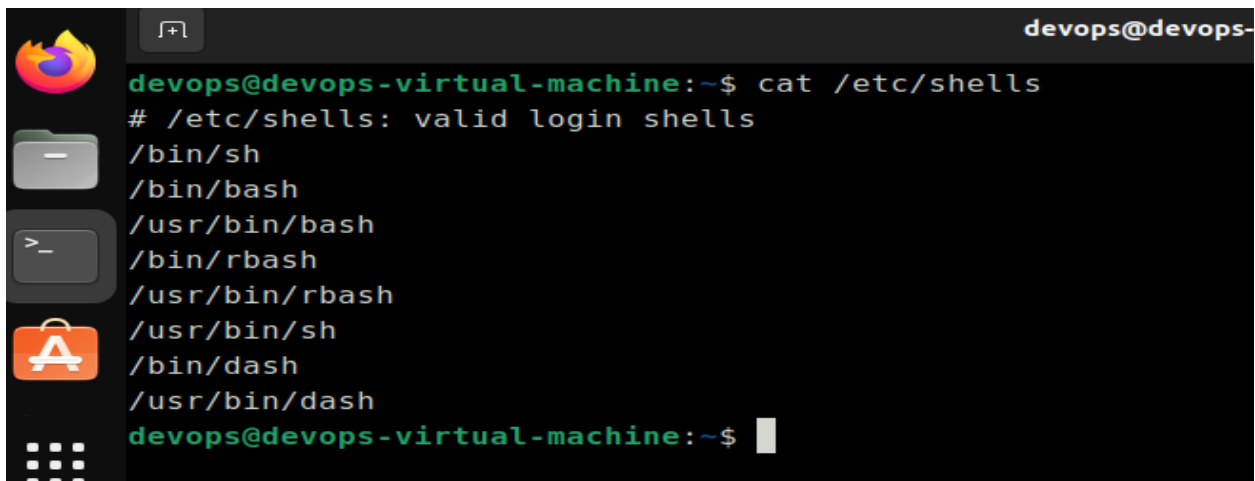
**e. Delete the line you are on and line 5.**

In command mode : dd (notice that I'm on line 5)

**f. How to step to the end of line and change to writing mode in one-step.**

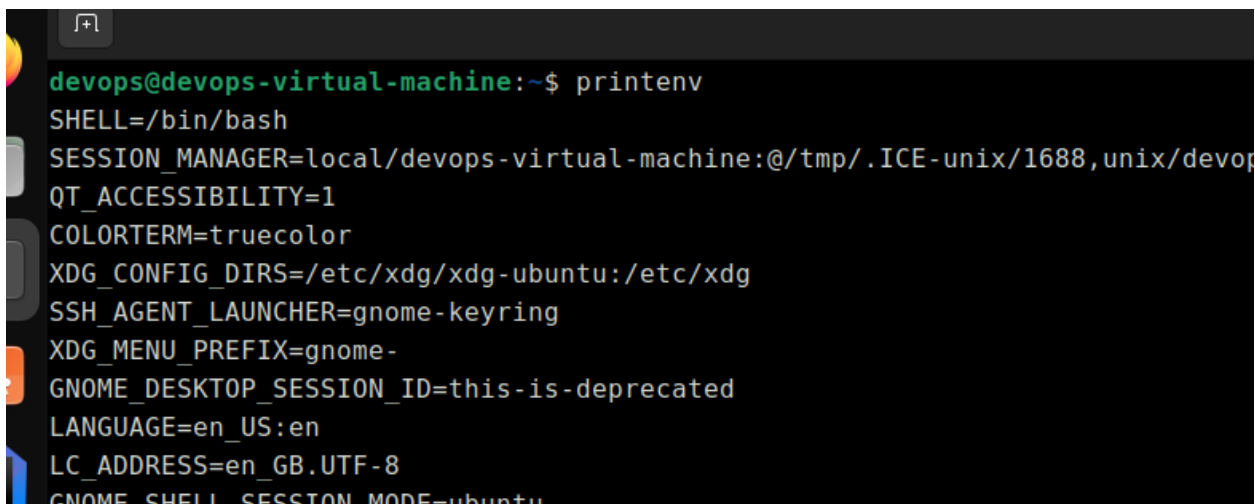
In command mode : A

### 3. List the available shells in your system.



```
devops@devops-
devops@devops-virtual-machine:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/usr/bin/sh
/bin/dash
/usr/bin/dash
devops@devops-virtual-machine:~$
```

### 4. List the environment variables in your current shell.



```
devops@devops-virtual-machine:~$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/devops-virtual-machine:@/tmp/.ICE-unix/1688,unix/devops-
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
LANGUAGE=en_US:en
LC_ADDRESS=en_GB.UTF-8
GNOME_SHELL_SESSION_MODE=ubuntu
```

## 5. List all of the environment variables for the bash shell.

```
devops@devops-virtual-machine:~$ set
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extglob:extquote:force_fig
BASH_ALIASES=()
BASH_ARGC=( [0]="0" )
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_VERSINFO=( [0]="2" [1]="11" )
BASH_LINENO=()
BASH_REMATCH=()
```

```
devops@devops-virtual-machine:~$ compgen -v
BASH
BASHOPTS
BASHPID
BASH_ALIASES
BASH_ARGC
BASH_ARGV
BASH_ARGV0
BASH_CMDS
```

## 6. What are the commands that list the value of a specific variable?

```
devops@devops-virtual-machine:~$ echo $HOME
/home/devops
devops@devops-virtual-machine:~$ printenv SHELL
/bin/bash
devops@devops-virtual-machine:~$ env | grep PATH
PATH=/opt/apache-maven-3.6.3/bin:/opt/jdk-13.0.1/bin:/usr/local/sbin:/usr/local
```

## 7. Display your current shell name.

```
devops@devops-virtual-machine:~$ printenv SHELL
/bin/bash
```

## 1. List the user commands and redirect the output to /tmp/commands.list

```
devops@devops-virtual-machine:~$ compgen -c > /tmp/commands.list
devops@devops-virtual-machine:~$ ls -l /tmp/commands.list
-rw-rw-r-- 1 devops devops 40188 Apr 13 16:28 /tmp/commands.list
devops@devops-virtual-machine:~$ head -n10 /tmp/commands.list
alert
egrep
fgrep
grep
l
la
ll
ls
if
then
devops@devops-virtual-machine:~$
```

## 2. Count the number of user commands

```
devops@devops-virtual-machine:~$ less /tmp/commands.list
devops@devops-virtual-machine:~$ man wc
devops@devops-virtual-machine:~$ wc -l /tmp/commands.list
3940 /tmp/commands.list
devops@devops-virtual-machine:~$ wc -w /tmp/commands.list
3940 /tmp/commands.list
devops@devops-virtual-machine:~$
```

## 3. Get all the users' names whose first character in their login is 'g'.

```
devops@devops-virtual-machine:~$ awk -F':' '/^g/ {print $1}' /etc/passwd
games
gnats
geoclue
gnome-initial-setup
gdm
devops@devops-virtual-machine:~$ cut -d: -f1 /etc/passwd | grep "^g"
games
gnats
geoclue
gnome-initial-setup
gdm
devops@devops-virtual-machine:~$
```

#### 4. Get the logins name and full names (comment) of logins starts with “g”.

```
devops@devops-virtual-machine:~$ cut -d: -f1,5 /etc/passwd | grep "^g"
games:games
gnats:Gnats Bug-Reporting System (admin)
geoclue:
gnome-initial-setup:
gdm:Gnome Display Manager
devops@devops-virtual-machine:~$ awk -F: '/^g/ {print $1,$5}' /etc/passwd
games games
gnats Gnats Bug-Reporting System (admin)
geoclue
gnome-initial-setup
gdm Gnome Display Manager
devops@devops-virtual-machine:~$
```

`awk -F'delimiter' '/pattern/ {print $feild1, $feild2, ...}' file-location`

#### 5. Save the output of the last command sorted by their full names in a file.

```
devops@devops-virtual-machine:~/practice/docs$ awk -F: '/^g/ {print $1,$5}' /etc/passwd | sort -k2 > sorted-awk-output
devops@devops-virtual-machine:~/practice/docs$ cat sorted-awk-output
geoclue
gnome-initial-setup
games games
gnats Gnats Bug-Reporting System (admin)
gdm Gnome Display Manager
devops@devops-virtual-machine:~/practice/docs$
```

#### 6. Write two commands:

**First: to search for all files on the system that is named `.bash_profile`.**

```
devops@devops-virtual-machine:~/practice/docs$ sudo find / -name ".bash_profile"
find: '/proc/1039/task/1039/net': Invalid argument
find: '/proc/1039/net': Invalid argument
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
devops@devops-virtual-machine:~/practice/docs$
```

```
devops@devops-virtual-machine:~/practice/docs$ su -
Password:
root@devops-virtual-machine:~# find / -name .bash_profile 2> /home/devops/practice/docs/errors2.txt
root@devops-virtual-machine:~# cat /home/devops/practice/docs/errors2.txt
find: '/proc/1039/task/1039/net': Invalid argument
find: '/proc/1039/net': Invalid argument
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
root@devops-virtual-machine:~# su - devops
devops@devops-virtual-machine:~$
```

```
devops@devops-virtual-machine:~/practice/docs$ sudo apt install plocate
Reading package lists... Done
Building dependency tree... Done
```

```
devops@devops-virtual-machine:~/practice/docs$ man plocate
devops@devops-virtual-machine:~/practice/docs$ locate .bash_profile
/usr/share/doc/adduser/examples/adduser.local.conf.examples/skel/dot.bash_profile
devops@devops-virtual-machine:~/practice/docs$
```

I don't know why there are no files called `.bash_profile` in my VM. To resolve this and go on with the lab, I replaced `.bash_profile` by `.profile`.

```
devops@devops-virtual-machine:~$ sudo find / -name .profile
find: '/proc/1038/task/1038/net': Invalid argument
find: '/proc/1038/net': Invalid argument
/home/devops/.profile
/root/.profile
/snap/core/14946/etc/skel/.profile
/snap/core/14946/root/.profile
/snap/core22/583/etc/skel/.profile
/snap/core22/583/root/.profile
/snap/core22/607/etc/skel/.profile
/snap/core22/607/root/.profile
/snap/core20/1852/etc/skel/.profile
/snap/core20/1852/root/.profile
/snap/core20/1695/etc/skel/.profile
/snap/core20/1695/root/.profile
/snap/core18/2721/etc/skel/.profile
/snap/core18/2721/root/.profile
/snap/core18/2620/etc/skel/.profile
/snap/core18/2620/root/.profile
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
/etc/skel/.profile
devops@devops-virtual-machine:~$
```



**Second: sorts the output of ls command on / recursively, Saving their output and error in 2 different files and sending them to the background.**

```
devops@devops-virtual-machine:~/practice/docs$ pwd && ls
/home/devops/practice/docs
sorted-awk-output
devops@devops-virtual-machine:~/practice/docs$ ls -R / > ls-root.txt 2> ls-root-error.txt &
[1] 5814
devops@devops-virtual-machine:~/practice/docs$ jobs
[1]+  Running                  ls --color=auto -R / > ls-root.txt 2> ls-root-error.txt &
devops@devops-virtual-machine:~/practice/docs$ ls -l
total 14284
-rw-rw-r-- 1 devops devops 122266 Apr 14 17:14 ls-root-error.txt
-rw-rw-r-- 1 devops devops 14498467 Apr 14 17:14 ls-root.txt
-rw-rw-r-- 1 devops devops 109 Apr 14 17:07 sorted-awk-output
[1]+  Exit 1                  ls --color=auto -R / > ls-root.txt 2> ls-root-error.txt
devops@devops-virtual-machine:~/practice/docs$ jobs
devops@devops-virtual-machine:~/practice/docs$
```

**7. Display the number of users who are logged now to the system.**

```
devops@devops-virtual-machine:~/practice/docs$ who | wc -l
1
devops@devops-virtual-machine:~/practice/docs$
```

**8. Display lines 7 to line 10 of /etc/passwd file**

```
devops@devops-virtual-machine:~/practice/docs$ head -n 10 /etc/passwd | tail -n 4
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
devops@devops-virtual-machine:~/practice/docs$ man sed
devops@devops-virtual-machine:~/practice/docs$ sed -n '7,10p' /etc/passwd
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
devops@devops-virtual-machine:~/practice/docs$ awk 'NR>=7 && NR<=10' /etc/passwd
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
devops@devops-virtual-machine:~/practice/docs$
```



## 9. What happens if you execute:

### • `cat filename1 | cat filename2`

```
devops@devops-virtual-machine:~/practice/docs$ ls
cat-pipe  ls-root-error.txt  ls-root.txt  sorted-awk-output
devops@devops-virtual-machine:~/practice/docs$ cat sorted-awk-output | cat cat-pipe
This is the second file in the pipe
filename2
devops@devops-virtual-machine:~/practice/docs$
```

Second cat will ignore the standard input and take its argument as an input then prints out its content.

### • `ls | rm`

```
devops@devops-virtual-machine:~/practice/docs$ ls
cat  cat-pipe  ls-root-error.txt  ls-root.txt  sorted-awk-output
devops@devops-virtual-machine:~/practice/docs$ ls | rm
rm: missing operand
Try 'rm --help' for more information.
devops@devops-virtual-machine:~/practice/docs$
```

rm does not read any input from standard input, the rm command will simply execute without any arguments resulting in error.

### • `ls /etc/passwd | wc -l`

```
devops@devops-virtual-machine:~/practice/docs$ ls /etc/passwd | wc -l
1
devops@devops-virtual-machine:~/practice/docs$ ls /etc/passwd
/etc/passwd
devops@devops-virtual-machine:~/practice/docs$
```

ls command produces a single line of output.

## 10. Issue the command `sleep 100`.

## 11. Stop the last command.

## 12. Resume the last command in the background

## 13. Issue the jobs command and see its output.

**14. Send the sleep command to the foreground and send it again to the background.**

**15. Kill the sleep command.**

```
devops@devops-virtual-machine: ~/practice/docs
devops@devops-virtual-machine:~/practice/docs$ ### sleep 100; stop; resume bg; jobs; fg then bg; kill ###
devops@devops-virtual-machine:~/practice/docs$
devops@devops-virtual-machine:~/practice/docs$ sleep 100
^Z
[1]+  Stopped                  sleep 100
devops@devops-virtual-machine:~/practice/docs$ bg %1
[1]+ sleep 100 &
devops@devops-virtual-machine:~/practice/docs$ jobs
[1]+  Running                  sleep 100 &
devops@devops-virtual-machine:~/practice/docs$ fg "%sleep"
sleep 100
^Z
[1]+  Stopped                  sleep 100
devops@devops-virtual-machine:~/practice/docs$ bg "%sleep"
[1]+ sleep 100 &
devops@devops-virtual-machine:~/practice/docs$ ps aux | grep sleep
devops      6266  0.0  0.0 17116  972 pts/0    S   18:12   0:00  sleep 100
devops      6272  0.0  0.1 17956  2328 pts/0    S+  18:13   0:00  grep --color=auto  sleep
devops@devops-virtual-machine:~/practice/docs$ kill -9 6266
devops@devops-virtual-machine:~/practice/docs$ jobs
[1]+  Killed                   sleep 100
devops@devops-virtual-machine:~/practice/docs$
```

I could've used `pkill sleep`, it will work properly in this case but it's more recommended to use PID as some processes names overlap.

**16. Display your processes only**

```
devops@devops-virtual-machine:~$ ps ux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
devops    1606  0.0  0.4 25228 9932 ?        Ss   16:25   0:05 /lib/systemd/systemd --user
devops    1607  0.0  0.0 170084  84 ?        S    16:25   0:00 (sd-pam)
devops    1613  0.0  0.0 48320  476 ?        S<sl 16:25   0:00 /usr/bin/pipewire
devops    1614  0.0  0.0 32204  604 ?        Ssl  16:25   0:00 /usr/bin/pipewire-media-session
devops    1615  0.0  0.2 1430988 5280 ?        S<sl 16:25   0:09 /usr/bin/pulseaudio --daemonize=no --log-t
devops    1617  0.0  0.0 76300  4 ?        Ss   16:25   0:06 /snap/snapd-desktop-integration/57/usr/bin
devops    1624  0.0  0.1 249656 2784 ?        Sl   16:25   0:00 /usr/bin/gnome-keyring-daemon --daemonize
devops    1629  0.0  0.3 14988 7232 ?        Ss   16:25   0:05 /usr/bin/dbus-daemon --session --address=s
devops    1633  0.0  0.0 249380 1736 ?        Ssl  16:25   0:00 /usr/libexec/gvfsd
devops    1638  0.0  0.0 380884 1688 ?        Sl   16:25   0:00 /usr/libexec/gvfsd-fuse /run/user/1000/gvf
devops    1646  0.0  0.0 171128 1676 tty2    Ssl+ 16:25   0:00 /usr/libexec/gdm-wayland-session env GNOME
devops    1651  0.0  0.0 231772  840 tty2    Sl+  16:25   0:00 /usr/libexec/gnome-session-binary --sessio
devops    1673  0.0  0.3 644344 6644 ?        SNsl 16:25   0:04 /usr/libexec/tracker-miner-fs-3
devops    1695  0.0  0.0 100648  244 ?        Ssl  16:25   0:00 /usr/libexec/gnome-session-ctl --monitor
devops    1709  0.0  0.1 601676 3088 ?        Ssl  16:25   0:01 /usr/libexec/gnome-session-binary --system
```

## 17. Display all processes except yours

```
devops@devops-virtual-machine:~$ ps aux | grep -v devops
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.3	250128	6808	?	Ss	16:22	0:08	/sbin/init auto noprompt splash
root	2	0.0	0.0	0	0	?	S	16:22	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	16:22	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	16:22	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I<	16:22	0:00	[slub_flushwq]
root	6	0.0	0.0	0	0	?	I<	16:22	0:00	[netns]
root	8	0.0	0.0	0	0	?	I<	16:22	0:00	[kworker/0:0H-events_highpri]
root	10	0.0	0.0	0	0	?	I<	16:22	0:00	[mm_percpu_wq]

## 18. Use the pgrep command to list your processes only

```
devops@devops-virtual-machine:~$ pgrep -u devops
```

```
1606
1607
1613
1614
1615
1617
1624
1629
1633
1638
1646
1651
```

## 19. Kill your processes only.

```
kill -9 -u $(whoami) > killed
kill -15 -u $USER > Terminated
```

1. Which command gives an overview of all current shell jobs?

jobs

```
devops@devops-virtual-machine:~$ jobs
```


```
[1]-  Running                  xlogo &
[2]+  Running                  sleep 100 &
```

```
devops@devops-virtual-machine:~$
```

2. How do you stop the current shell job to continue running it in the background?

Ctrl - z then bg %(job-id or job-name)

```
devops@devops-virtual-machine:~$ xlogo
^Z
[1]+  Stopped                  xlogo
devops@devops-virtual-machine:~$ bg %xlogo
[1]+ xlogo &
devops@devops-virtual-machine:~$ jobs
[1]+  Running                  xlogo &
devops@devops-virtual-machine:~$
```



3. Which keystroke combination can you use to cancel the current shell job?

Ctrl - c

```
devops@devops-virtual-machine:~$ fg "%xlogo"
xlogo
^C
devops@devops-virtual-machine:~$
```

4. A user is asking you to cancel one of the jobs he has started. You cannot access the shell that user currently is working from. What can you do to cancel his job anyway?

```
devops@devops-virtual-machine:~$ su - jhon
Password:
jhon@devops-virtual-machine:~$ whoami
jhon
jhon@devops-virtual-machine:~$ jobs
jhon@devops-virtual-machine:~$ sleep 3600 &
[1] 6835
jhon@devops-virtual-machine:~$ jobs
[1]+  Running                  sleep 3600 &
jhon@devops-virtual-machine:~$
jhon@devops-virtual-machine:~$
jhon@devops-virtual-machine:~$ su - devops
Password:
devops@devops-virtual-machine:~$ ps aux | grep sleep
jhon      6835  0.0  0.0  17116  976 pts/0    S   20:14   0:00 sleep 3600
devops    6846  0.0  0.1  17956  2292 pts/0    S+  20:14   0:00 grep --color=auto sleep
devops@devops-virtual-machine:~$ sudo kill 6835
devops@devops-virtual-machine:~$ ps aux | grep sleep
devops    6852  0.0  0.1  17956  2316 pts/0    S+  20:14   0:00 grep --color=auto sleep
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$
devops@devops-virtual-machine:~$ su - jhon
Password:
jhon@devops-virtual-machine:~$ jobs
jhon@devops-virtual-machine:~$
```

**ps aux | grep user-job**

**Get the PID as the user id will be displayed**

**sudo kill PID (Terminate)**

5. Which command would you use to show parent-child relationships between processes?

~\$ **ps tree (options) (pid)**

```
devops@devops-virtual-machine:~$ ps aux | grep sbin
root      1  0.0  0.3 250128 6528 ?        Ss   16:22   0:09 /sbin/init auto noprompt splash
root      865  0.0  0.0   2812   0 ?        Ss   16:23   0:00 /usr/sbin/acpid
root      870  0.0  0.0   18240   672 ?        Ss   16:23   0:01 /usr/sbin/cron -f -P
root      872  0.0  0.2 269788 5056 ?        Ssl  16:23   0:05 /usr/sbin/NetworkManager --no-daemon
root      878  0.0  0.0   82828   656 ?        Ssl  16:23   0:01 /usr/sbin/irqbalance --foreground
syslog    882  0.0  0.0  222400   352 ?        Ssl  16:23   0:01 /usr/sbin/rsyslogd -n -iNONE
root      906  0.0  0.0   16496    20 ?        Ss   16:23   0:00 /sbin/wpa_supplicant -u -s -O /run/wpa_supplicant
root      975  0.0  0.1  244212  2104 ?        Ssl  16:23   0:00 /usr/sbin/ModemManager
root      981  0.0  0.1   81756  2288 ?        Ss   16:23   0:03 /usr/sbin/cupsd -l
root     1016  0.0  0.1  250096  2424 ?        Ssl  16:23   0:00 /usr/sbin/gdm3
root     1058  0.0  0.0   172612   612 ?        Ssl  16:23   0:00 /usr/sbin/cups-browsed
kernoops 1070  0.0  0.0   13080    720 ?        Ss   16:23   0:00 /usr/sbin/kerneloops --test
kernoops 1084  0.0  0.0   13080    776 ?        Ss   16:23   0:00 /usr/sbin/kerneloops
devops    7024  0.0  0.1   17956  2332 pts/0    S+   20:28   0:00 grep --color=auto sbin

devops@devops-virtual-machine:~$ ps tree -hpu 7024
devops@devops-virtual-machine:~$ ps tree -hpu 1084
kerneloops(1084,kernoops)
devops@devops-virtual-machine:~$ ps tree -hpu 1070
kerneloops(1070,kernoops)
devops@devops-virtual-machine:~$ ps tree -hpu 882
rsyslogd(882,syslog)
├─{rsyslogd}(919)
├─{rsyslogd}(920)
└─{rsyslogd}(921)
devops@devops-virtual-machine:~$
```

6. Which command enables you to change the priority of PID 1234 to a higher priority?

~\$ **renice -n -15 1234**

7. On your system, 20 **dd** processes are currently running. What is the easiest way to stop all of them?

~\$ **kill dd**

8. Which command enables you to stop the command with the name **mycommand**?

~\$ **kill -f mycommnd**

## 9. Which command do you use from **top** to kill a process?

~\$ top > press k > enter PID > enter signal

```
top - 20:39:17 up 4:16, 1 user, load average: 0.31, 0.16, 0.11
Tasks: 301 total, 1 running, 299 sleeping, 0 stopped, 1 zombie
%Cpu(s): 0.0 us, 0.2 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1940.6 total, 66.5 free, 1687.0 used, 187.1 buff/cache
MiB Swap: 3140.0 total, 1566.5 free, 573.5 used, 93.6 avail Mem
Send pid 7104 signal [15/sigterm] 15
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
35	root	20	0	0	0	0	S	1.0	0.0	0:27.93	kcompactd0
740	systemd+	20	0	14960	1084	920	S	0.7	0.1	0:25.01	systemd-oond
759	root	20	0	326116	2400	1980	S	0.3	0.1	0:41.59	vmtoolsd
1732	devops	20	0	4391340	142920	39088	S	0.3	7.2	2:59.56	gnome-shell
1981	devops	20	0	299856	10596	6340	S	0.3	0.5	0:37.70	vmtoolsd
6816	root	20	0	402404	11732	9480	S	0.3	0.6	0:02.54	rwu
7104	devops	20	0	21972	4088	3228	R	0.3	0.2	0:00.10	top
7104	root	20	0	250128	6148	3100	S	0.0	0.3	0:09.60	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04	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	0:00.02	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
11	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
14	root	20	0	0	0	0	S	0.0	0.0	0:01.34	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:09.78	rcu_preempt
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.37	migration/0
17	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
21	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.67	migration/1
23	root	20	0	0	0	0	S	0.0	0.0	0:01.93	ksoftirqd/1
25	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri

devops@devops-virtual-machine:~\$

## 10. What is required to select a performance profile that best matches your system needs?

The most common tool for this purpose is **tuned**, which is a system daemon that monitors system activity and can apply different performance profiles depending on the workload.