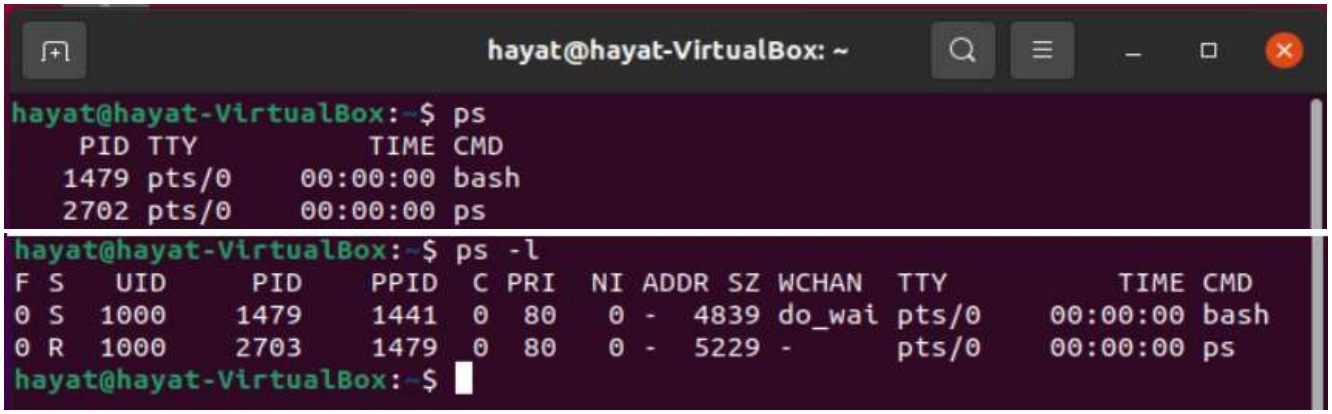
TP N°3 d'Administration Linux

❖ *EXERCICE 1 : « Gestion des Processus »*

1. Affichage de la liste des processus associés :



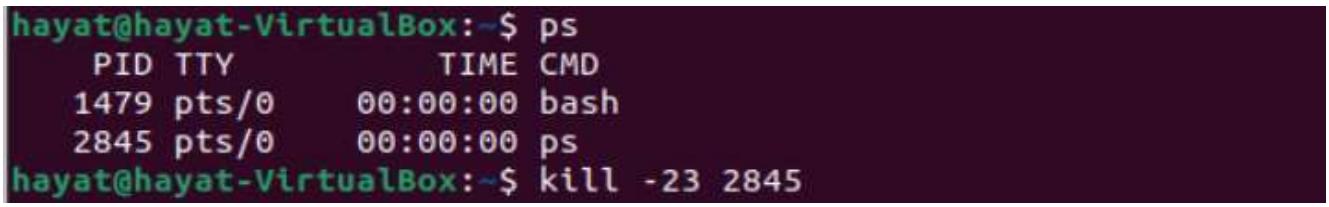
On constate que PPID du processus ps = PID du processus bash cela veut dire que **le processus bash est le processus père du ps**

2. Exécution des commandes suivants impliquent :

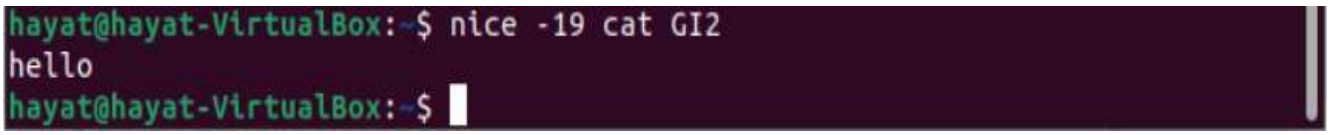
sigkill (9)
Le processus est tué sans pouvoir de défendre.

sigstop (23)
Le processus s'interrompt.

sigcont (25)
Le processus interrompu redémarre.



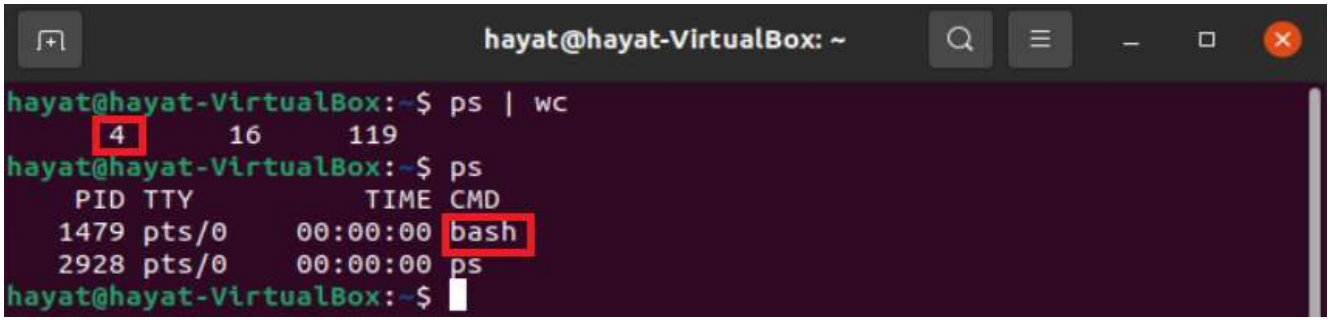
3. La commande **nice** pour lancer des commandes ayant une faible priorité où **19 désigne le nombre du processus ⇔ Demande de suspension imbloquable**



4. Affichage de **pstree** : on remarque que le processus **systemd** est la racine où plusieurs processeurs lui héritent comme **ModemManager, NetworkManager, cron** et n'importe quel processus lancé par le système. L'identifiant de **processus parent (parent process ID), ou ppid,** est simplement l'identifiant du père de processus.

5. La commande **ps | wc** compte deux processus associés aux commandes **ps et wc** que vous venez de lancer en utilisant la pipe | lorsqu'on lance la commande on a :
wc : word count est une commande qui permet de compter le nombre des mots **ps | wc** désigne le nombre de mots trouvés lorsqu'on lance la commande **ps**

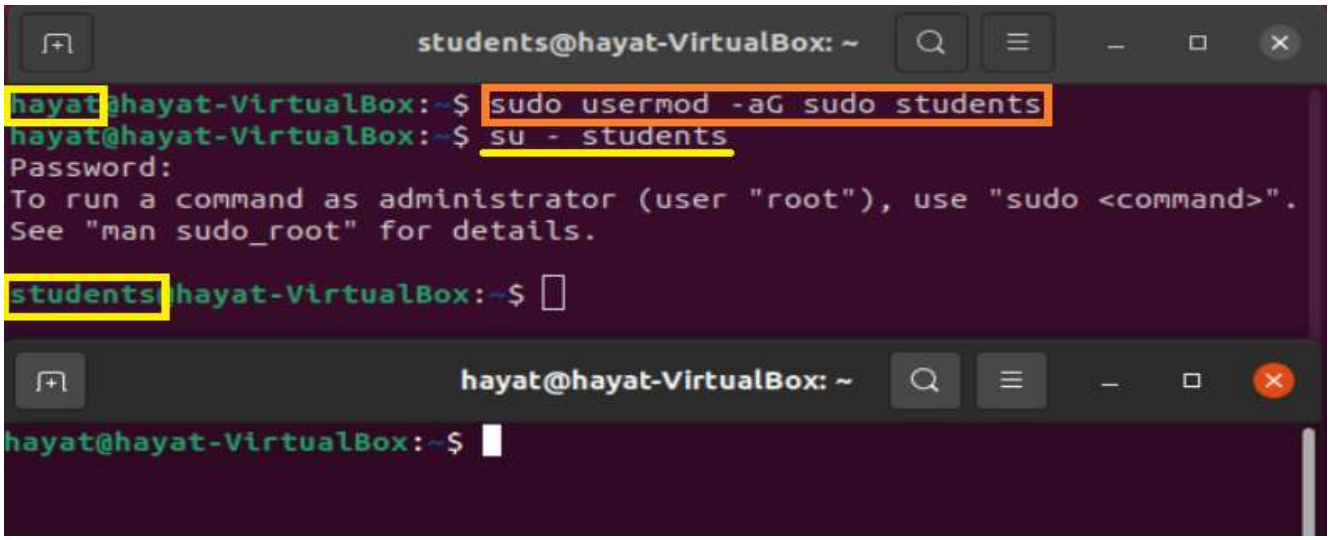
   - **4** : désigne le nombre maximal des lettres du nom du processus
   - **16** : désigne le nombre de bits pour lesquels ps est utilisé
   - **119** : le nombre total des lignes de la liste des processus lancés, utilisé par l'utilisateur et par le système



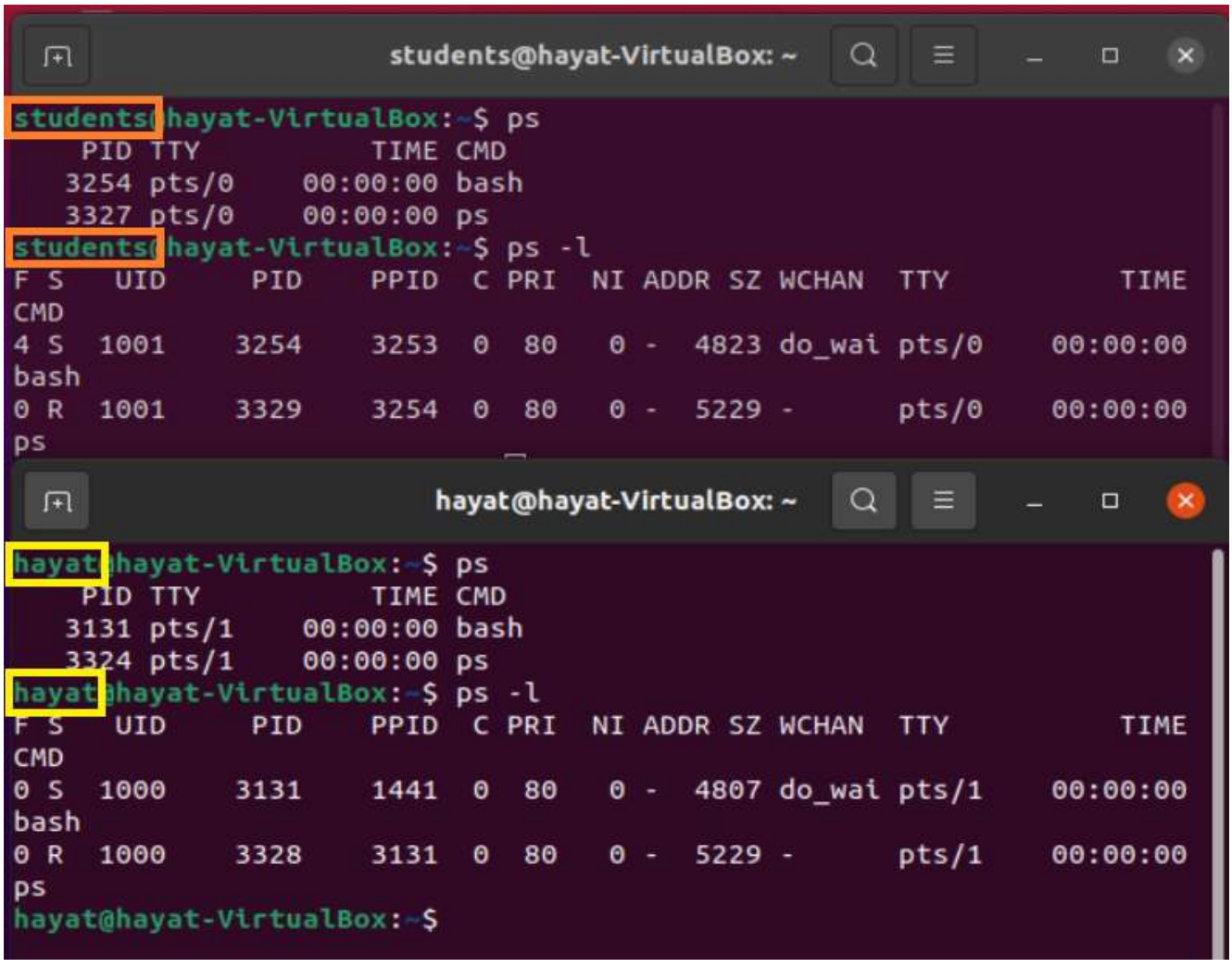
6. Les commandes qui permettent de reprendre l'exécution d'une instruction interrompue par un ^Z sont : **fg et bg. fg permet de poursuivre le déroulement du programme. bg le fait aussi mais il relance le programme**

❖ ***EXERCICE 2 : « VISUALISATION DES PROCESSUS »***

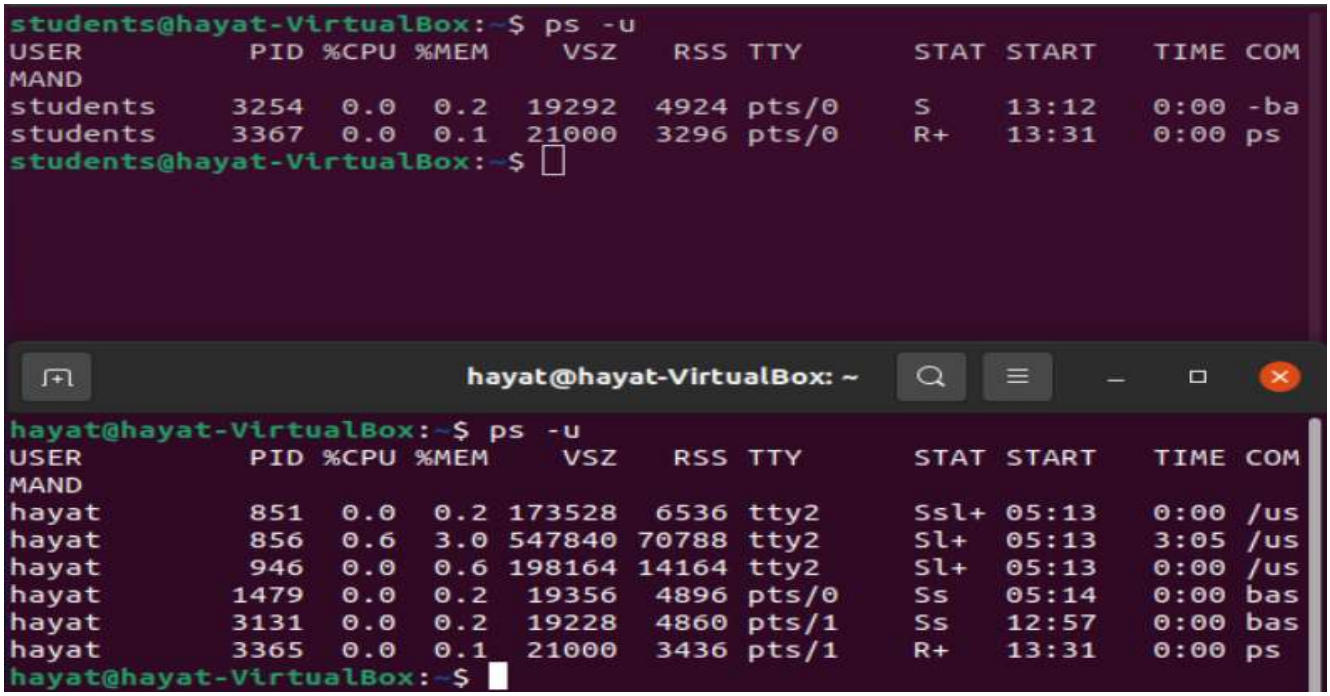
2.1) D'après le TP précédent on a créé un nouvel utilisateur nommé **« students »** il faut ajouter cet utilisateur au groupe sudo par cette commande **« sudo usermod -aG sudo students »** puis on l'accède par cette commande **su - students**



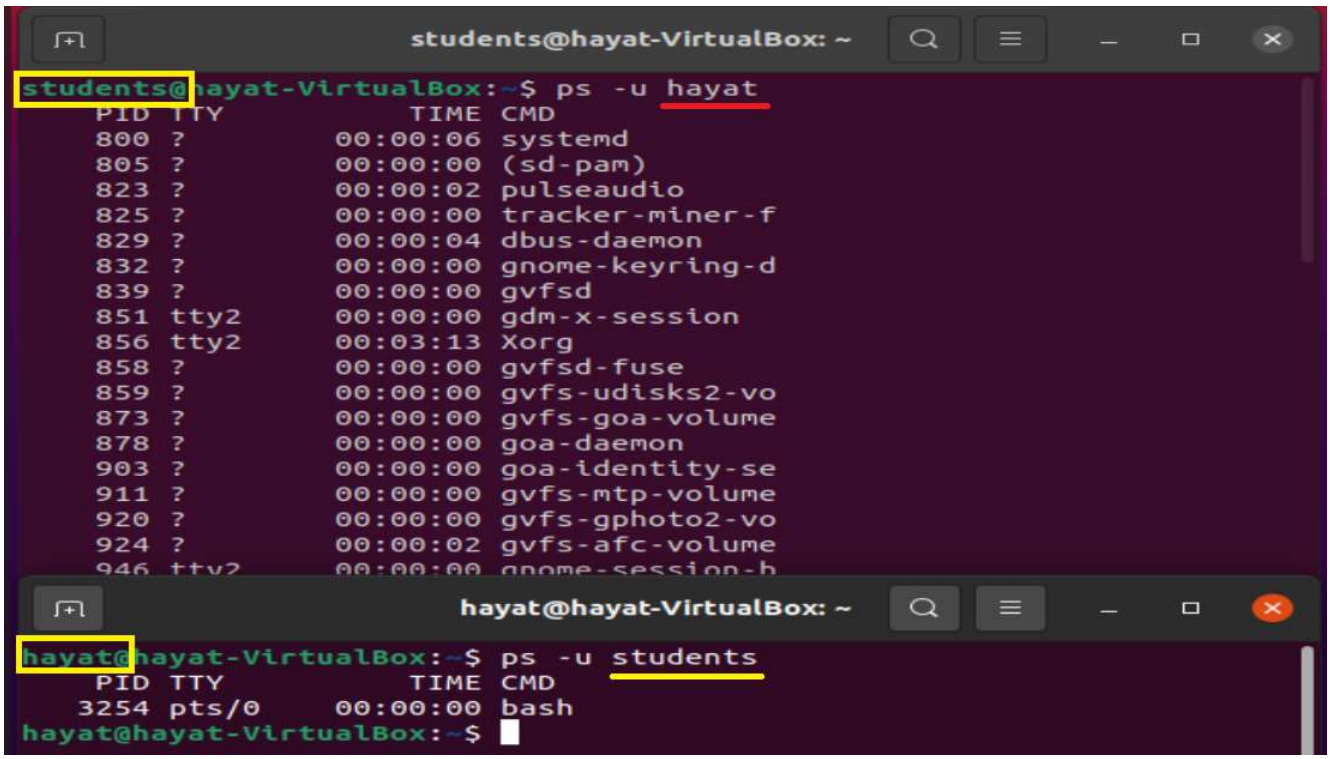
2.2) Affichage de la liste des processus pour chaque user : **PID c'est le numéro d'identification du processus PPID le numéro d'identification du processus père**



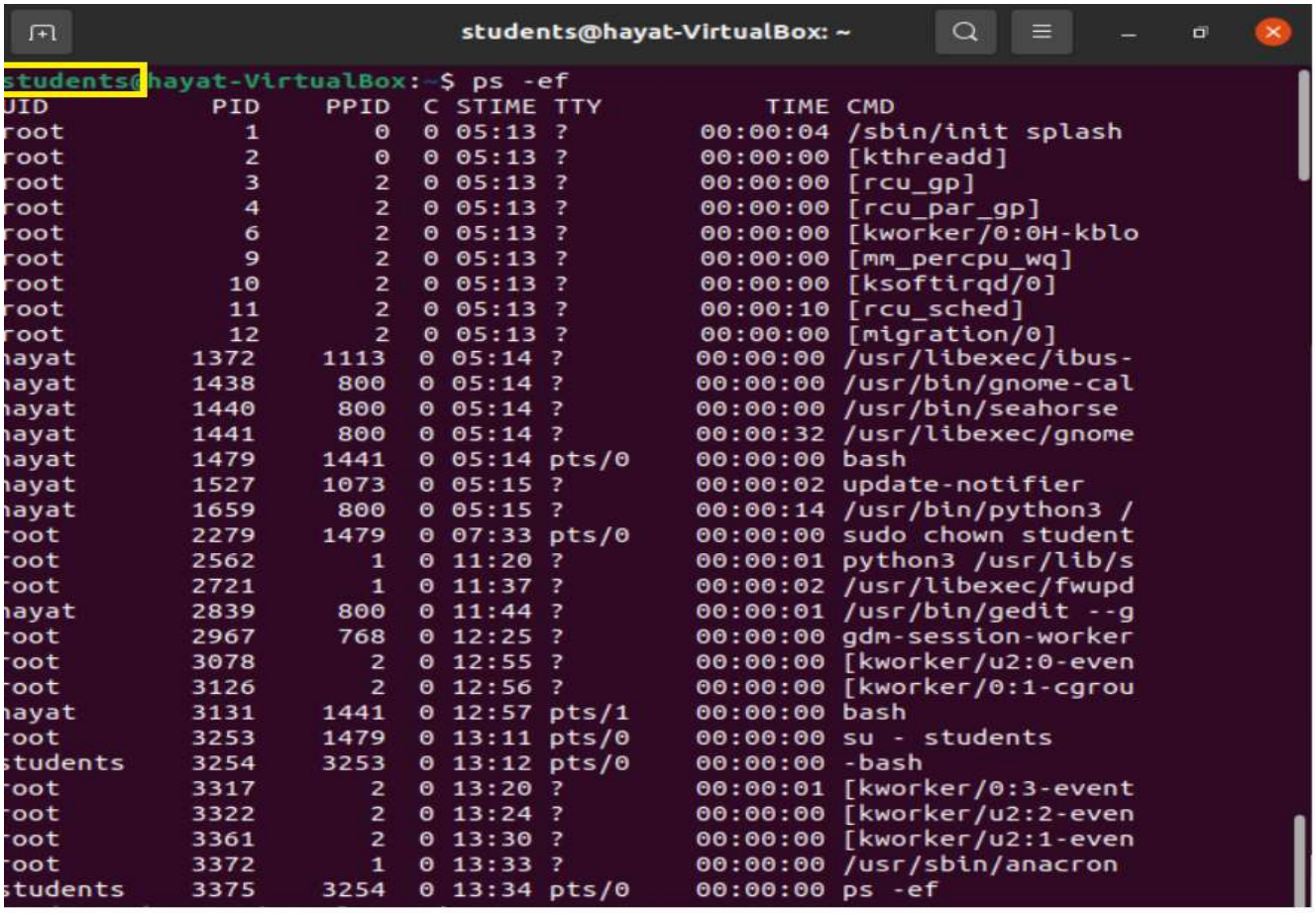
2.3) L’affichage des processus relatifs à votre terminal tout en affichant le nom de l’utilisateur

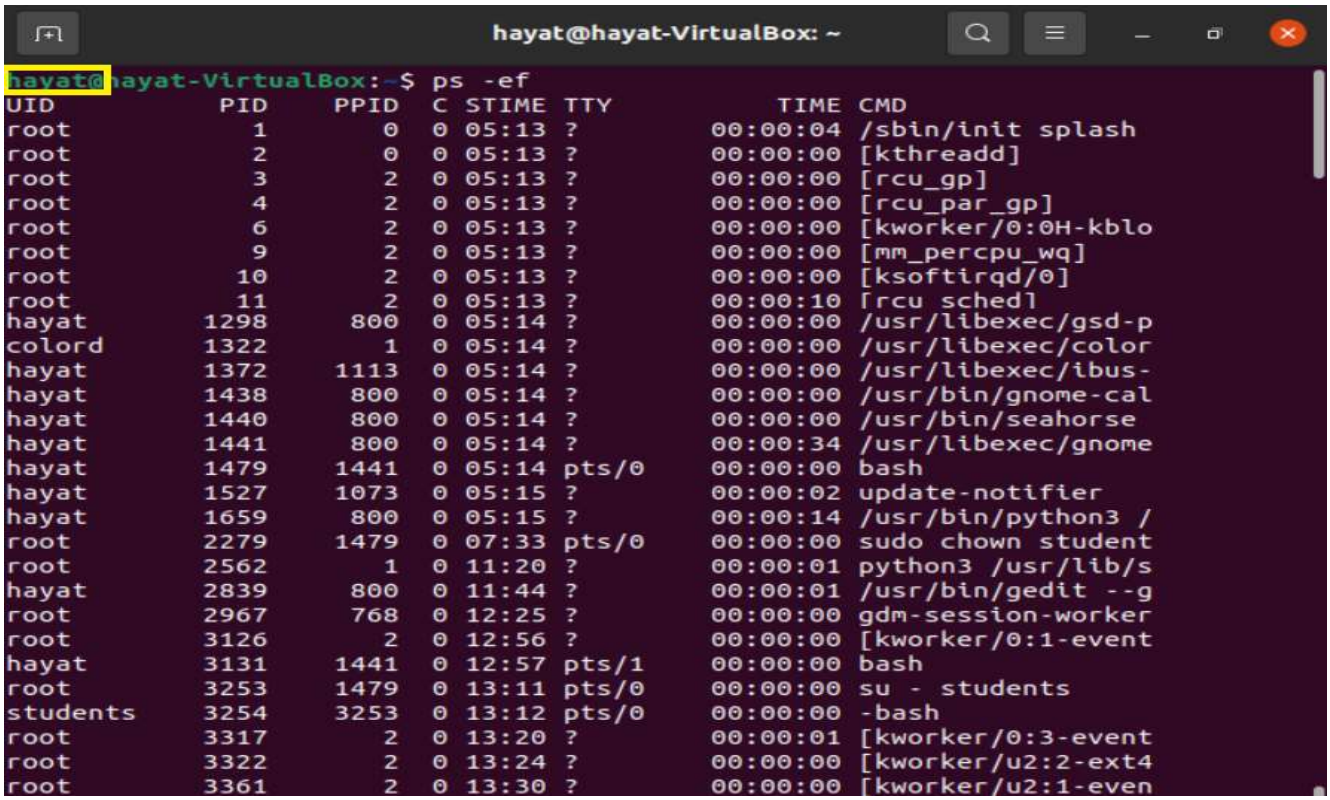


2.4) Affichage les processus de l'utilisateur 2 depuis le terminal de l'utilisateur 1 et vice versa

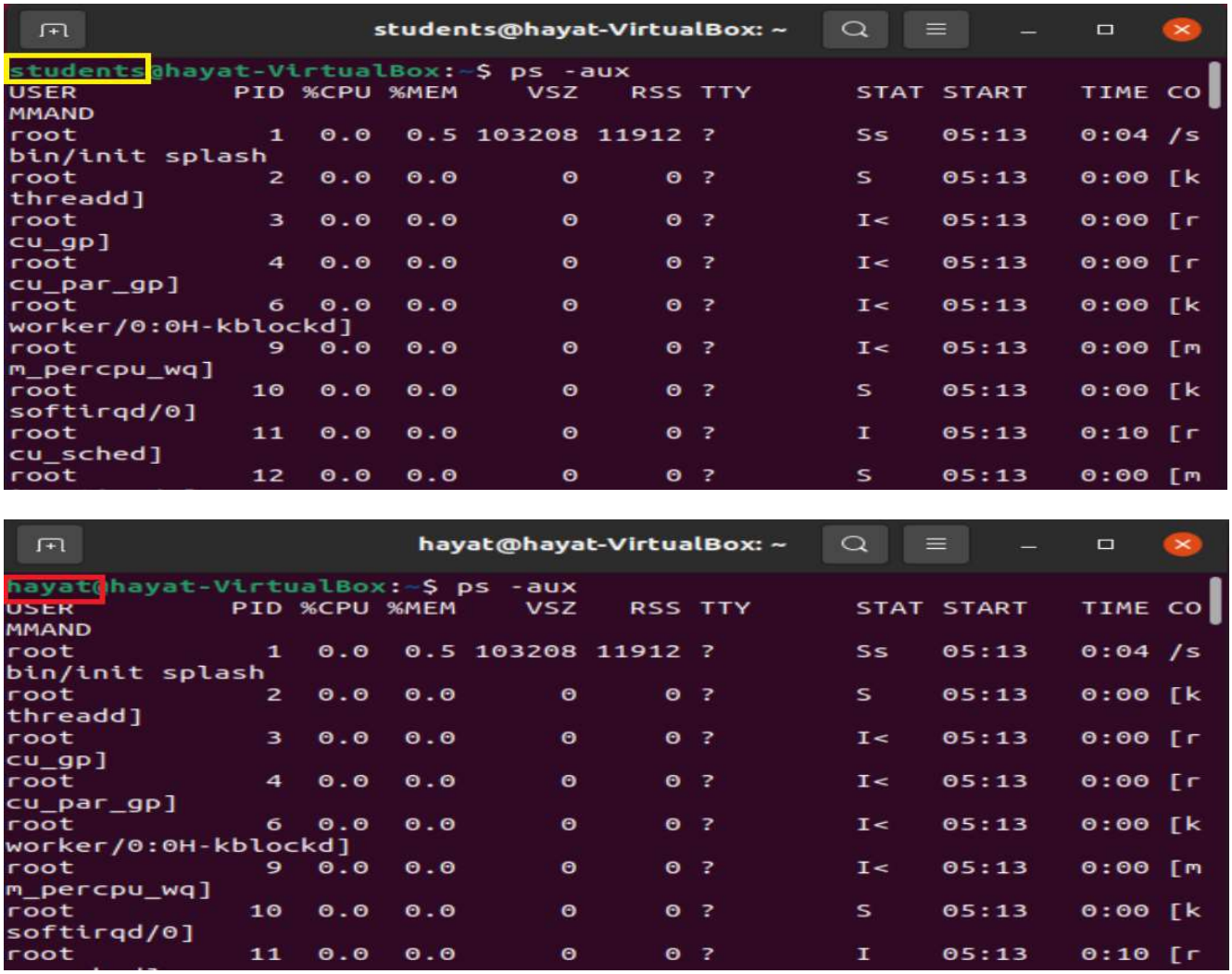


2.5) Affichage des processus de tous les utilisateurs

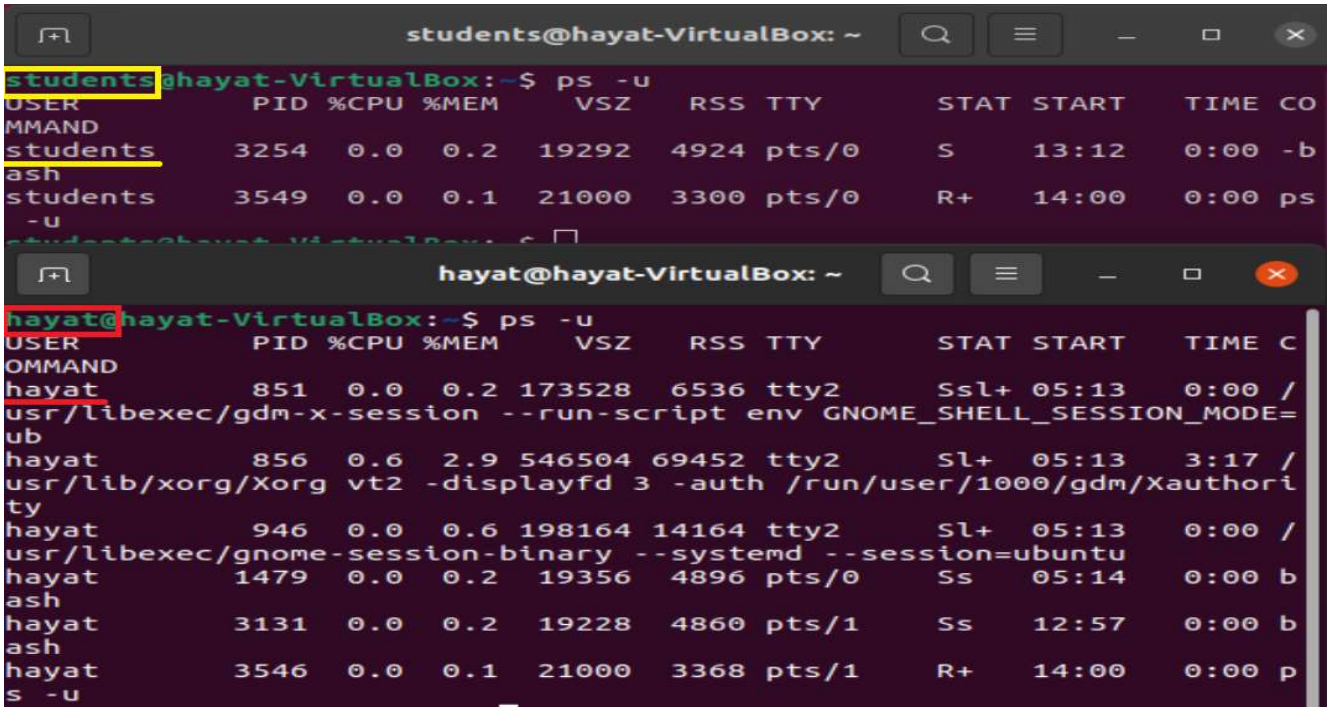

students@hayat-VirtualBox: ~
students@hayat-VirtualBox:~$ ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 05:13 ? 00:00:04 /sbin/init splash
root 2 0 0 05:13 ? 00:00:00 [kthreadd]
root 3 2 0 05:13 ? 00:00:00 [rcu_gp]
root 4 2 0 05:13 ? 00:00:00 [rcu_par_gp]
root 6 2 0 05:13 ? 00:00:00 [kworker/0:0H-kblo
root 9 2 0 05:13 ? 00:00:00 [mm_percpu_wq]
root 10 2 0 05:13 ? 00:00:00 [ksoftirqd/0]
root 11 2 0 05:13 ? 00:00:10 [rcu_sched]
root 12 2 0 05:13 ? 00:00:00 [migration/0]
hayat 1372 1113 0 05:14 ? 00:00:00 /usr/libexec/ibus-
hayat 1438 800 0 05:14 ? 00:00:00 /usr/bin/gnome-cal
hayat 1440 800 0 05:14 ? 00:00:00 /usr/bin/seahorse
hayat 1441 800 0 05:14 ? 00:00:32 /usr/libexec/gnome
hayat 1479 1441 0 05:14 pts/0 00:00:00 bash
hayat 1527 1073 0 05:15 ? 00:00:02 update-notifier
hayat 1659 800 0 05:15 ? 00:00:14 /usr/bin/python3 /
root 2279 1479 0 07:33 pts/0 00:00:00 sudo chown student
root 2562 1 0 11:20 ? 00:00:01 python3 /usr/lib/s
root 2721 1 0 11:37 ? 00:00:02 /usr/libexec/fwupd
hayat 2839 800 0 11:44 ? 00:00:01 /usr/bin/gedit --g
root 2967 768 0 12:25 ? 00:00:00 gdm-session-worker
root 3078 2 0 12:55 ? 00:00:00 [kworker/u2:0-even
root 3126 2 0 12:56 ? 00:00:00 [kworker/0:1-cgrou
hayat 3131 1441 0 12:57 pts/1 00:00:00 bash
root 3253 1479 0 13:11 pts/0 00:00:00 su - students
students 3254 3253 0 13:12 pts/0 00:00:00 -bash
root 3317 2 0 13:20 ? 00:00:01 [kworker/0:3-event
root 3322 2 0 13:24 ? 00:00:00 [kworker/u2:2-even
root 3361 2 0 13:30 ? 00:00:00 [kworker/u2:1-even
root 3372 1 0 13:33 ? 00:00:00 /usr/sbin/anacron
students 3375 3254 0 13:34 pts/0 00:00:00 ps -ef


hayat@hayat-VirtualBox: ~
hayat@hayat-VirtualBox:~$ ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 05:13 ? 00:00:04 /sbin/init splash
root 2 0 0 05:13 ? 00:00:00 [kthreadd]
root 3 2 0 05:13 ? 00:00:00 [rcu_gp]
root 4 2 0 05:13 ? 00:00:00 [rcu_par_gp]
root 6 2 0 05:13 ? 00:00:00 [kworker/0:0H-kblo
root 9 2 0 05:13 ? 00:00:00 [mm_percpu_wq]
root 10 2 0 05:13 ? 00:00:00 [ksoftirqd/0]
root 11 2 0 05:13 ? 00:00:10 [rcu_sched]
hayat 1298 800 0 05:14 ? 00:00:00 /usr/libexec/gsd-p
colord 1322 1 0 05:14 ? 00:00:00 /usr/libexec/color
hayat 1372 1113 0 05:14 ? 00:00:00 /usr/libexec/ibus-
hayat 1438 800 0 05:14 ? 00:00:00 /usr/bin/gnome-cal
hayat 1440 800 0 05:14 ? 00:00:00 /usr/bin/seahorse
hayat 1441 800 0 05:14 ? 00:00:34 /usr/libexec/gnome
hayat 1479 1441 0 05:14 pts/0 00:00:00 bash
hayat 1527 1073 0 05:15 ? 00:00:02 update-notifier
hayat 1659 800 0 05:15 ? 00:00:14 /usr/bin/python3 /
root 2279 1479 0 07:33 pts/0 00:00:00 sudo chown student
root 2562 1 0 11:20 ? 00:00:01 python3 /usr/lib/s
hayat 2839 800 0 11:44 ? 00:00:01 /usr/bin/gedit --g
root 2967 768 0 12:25 ? 00:00:00 gdm-session-worker
root 3126 2 0 12:56 ? 00:00:00 [kworker/0:1-event
hayat 3131 1441 0 12:57 pts/1 00:00:00 bash
root 3253 1479 0 13:11 pts/0 00:00:00 su - students
students 3254 3253 0 13:12 pts/0 00:00:00 -bash
root 3317 2 0 13:20 ? 00:00:01 [kworker/0:3-event
root 3322 2 0 13:24 ? 00:00:00 [kworker/u2:2-ext4
root 3361 2 0 13:30 ? 00:00:00 [kworker/u2:1-even

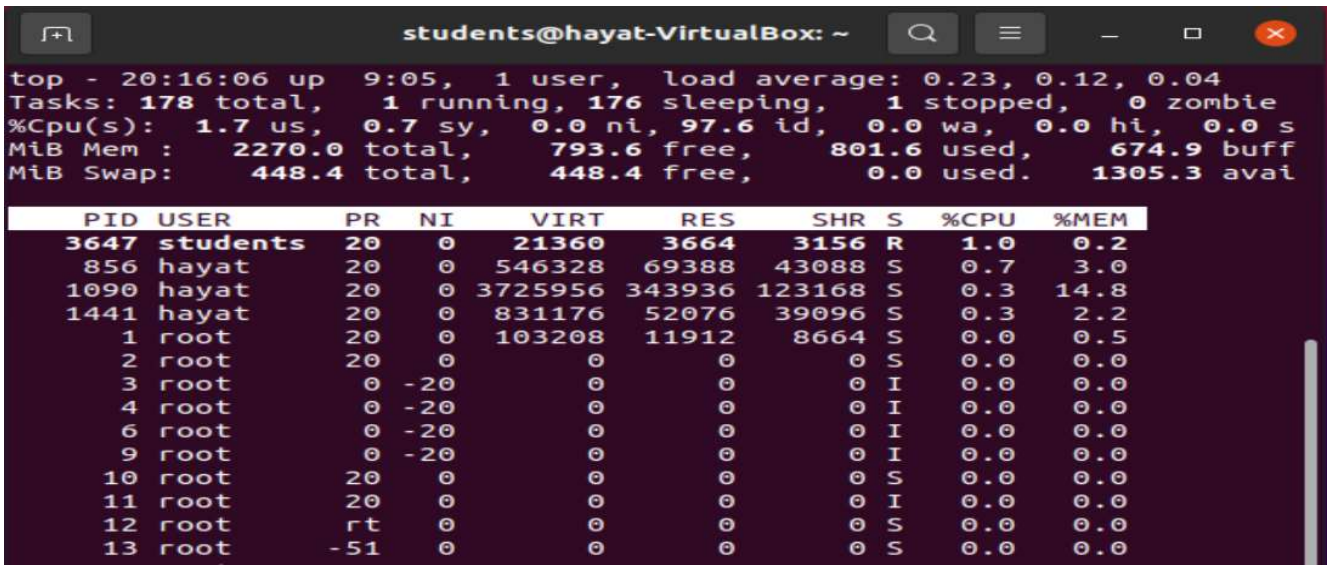
2.6) Les processus qui sont en cours d'exécution pour chaque utilisateur

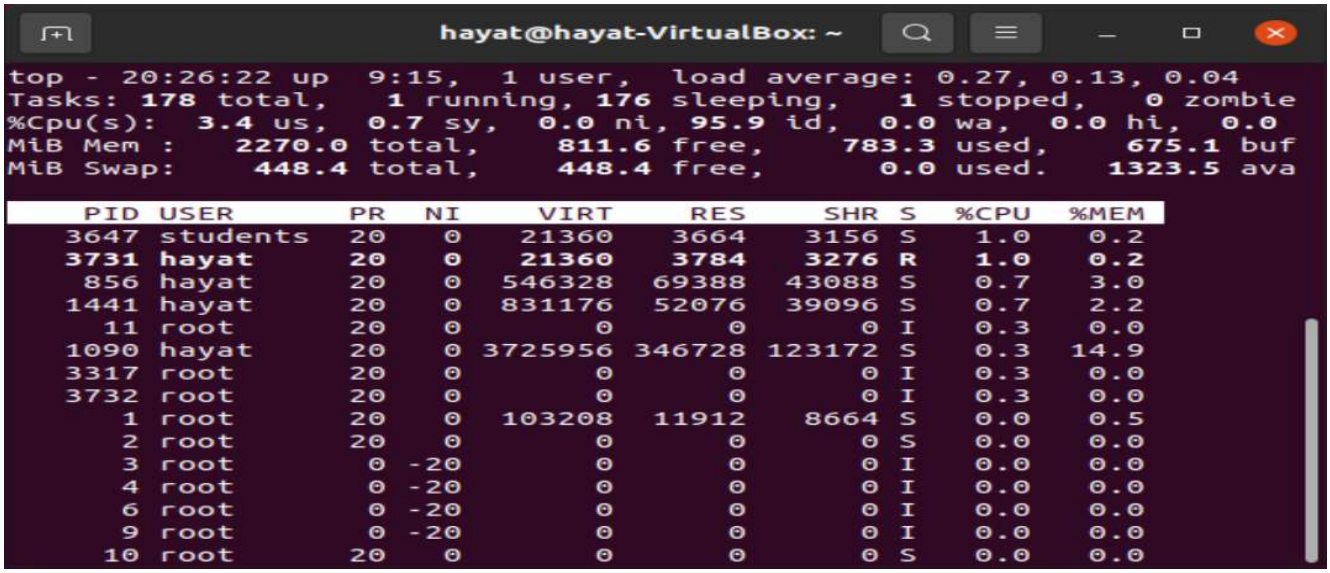


2.7) La liste des processus dont vous êtes des propriétaires



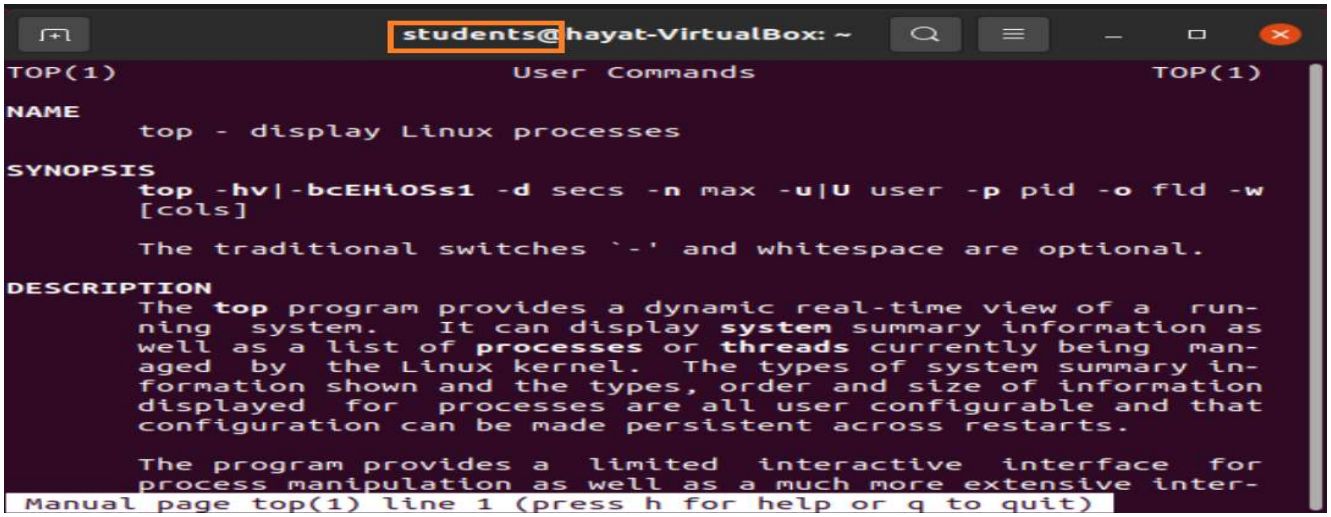
2.8) Voici l'exécution de la commande **top :**

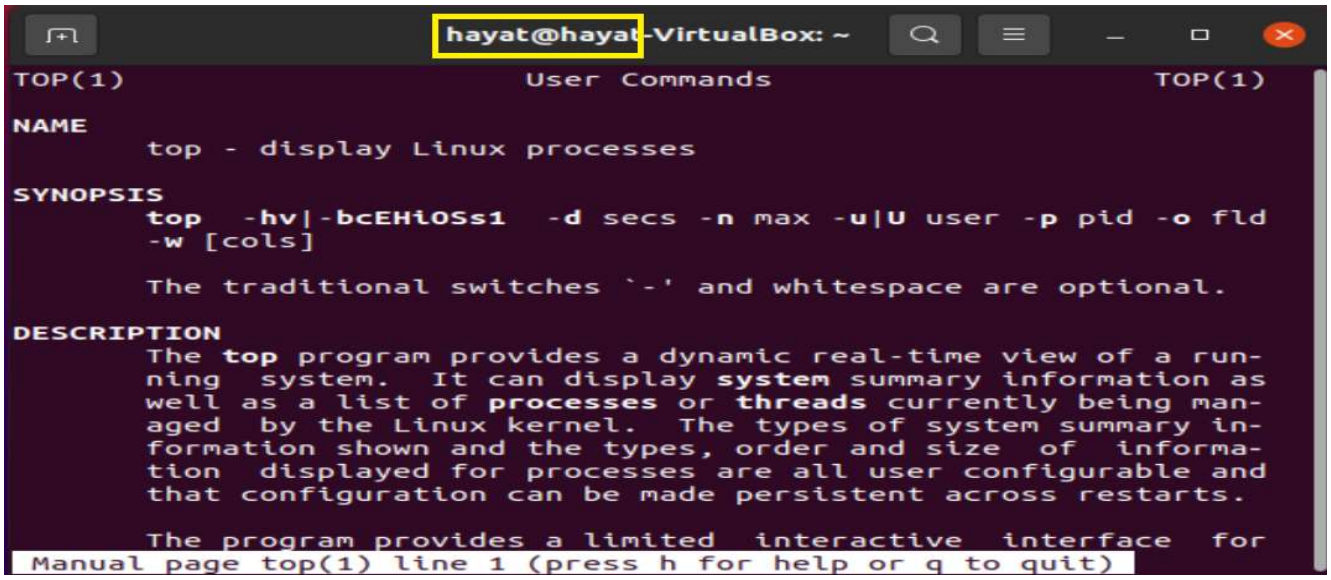




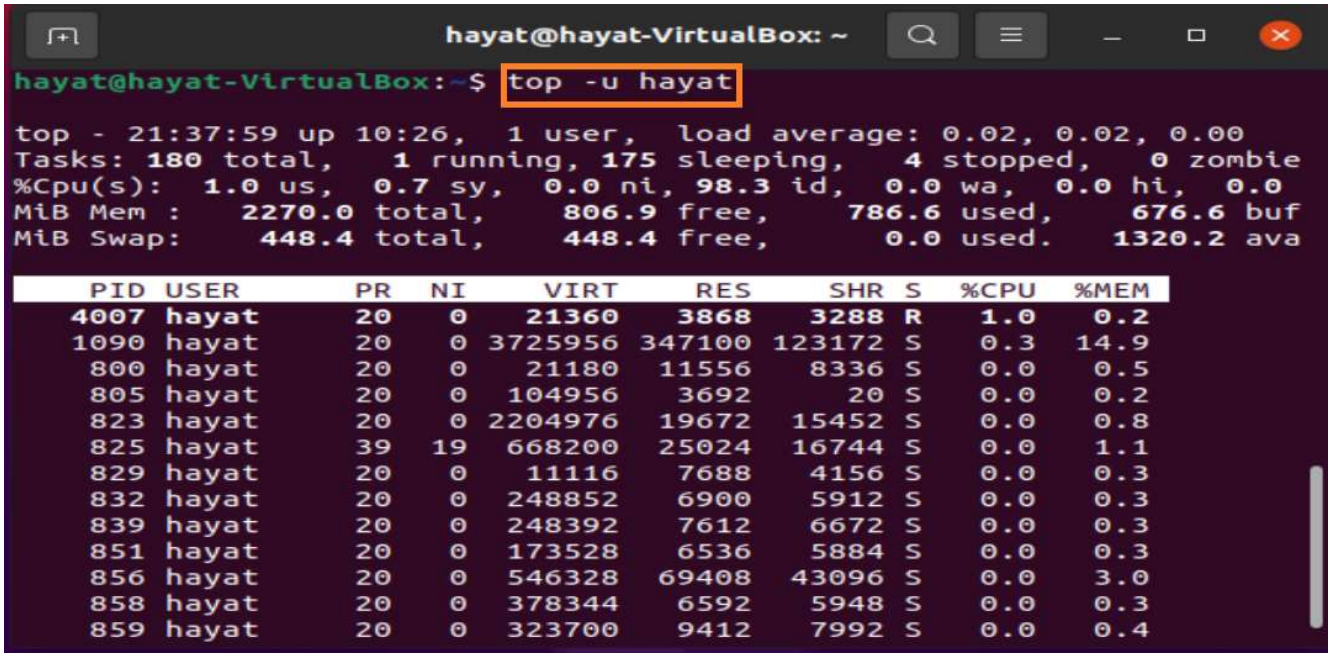
2.9) L'exécution de la commande **top :**

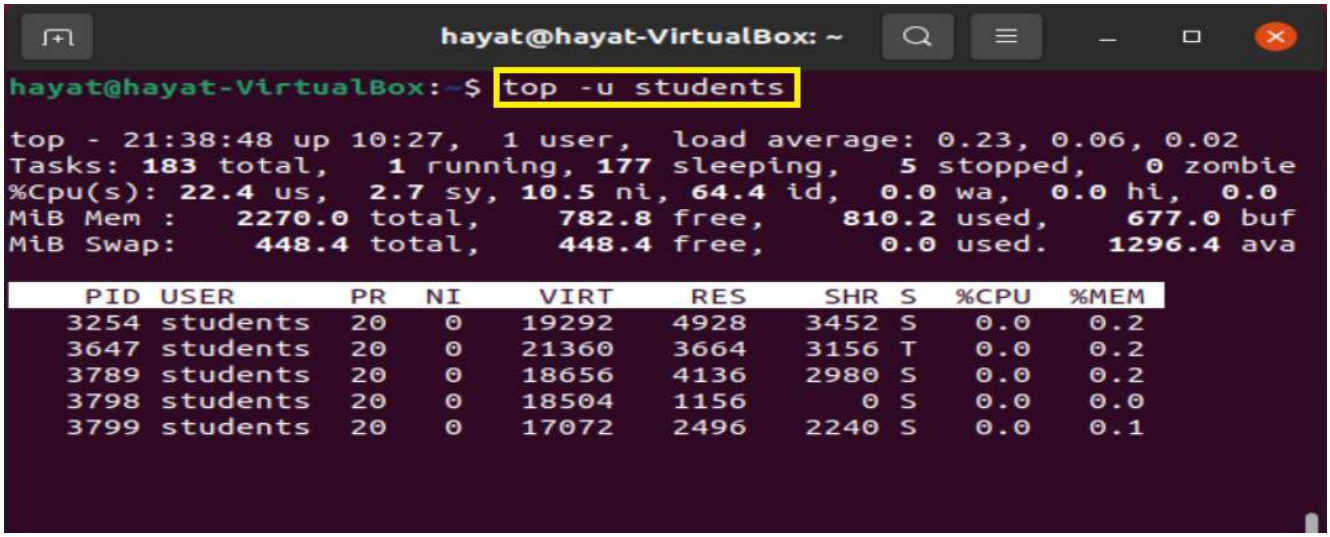
- ✓ **h : pour obtenir de l'aide sur la commande top.**
- ✓ **R : pour trier par numéro PID**
- ✓ **M : pour trier par l'utilisation de RAM**
- ✓ **u : appuyez sur u puis sur le nom d'utilisateur pour obtenir uniquement les détails du processus utilisateur**
- ✓ **k : pour tuer un processus, appuyez sur k puis sur le numéro PID puis entrez pour tuer un processus**
- ✓ **r : Pour renommer un processus, appuyez sur r puis sur le PID no puis sur la valeur de renice pour renommer un processus.**
- ✓ **q : Pour quitter la commande top.**



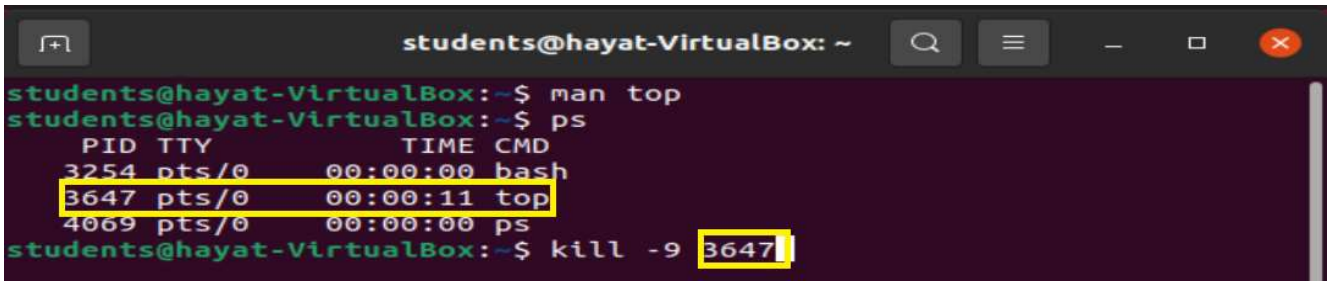


2.10) L'activité des processus pour un seul utilisateur :



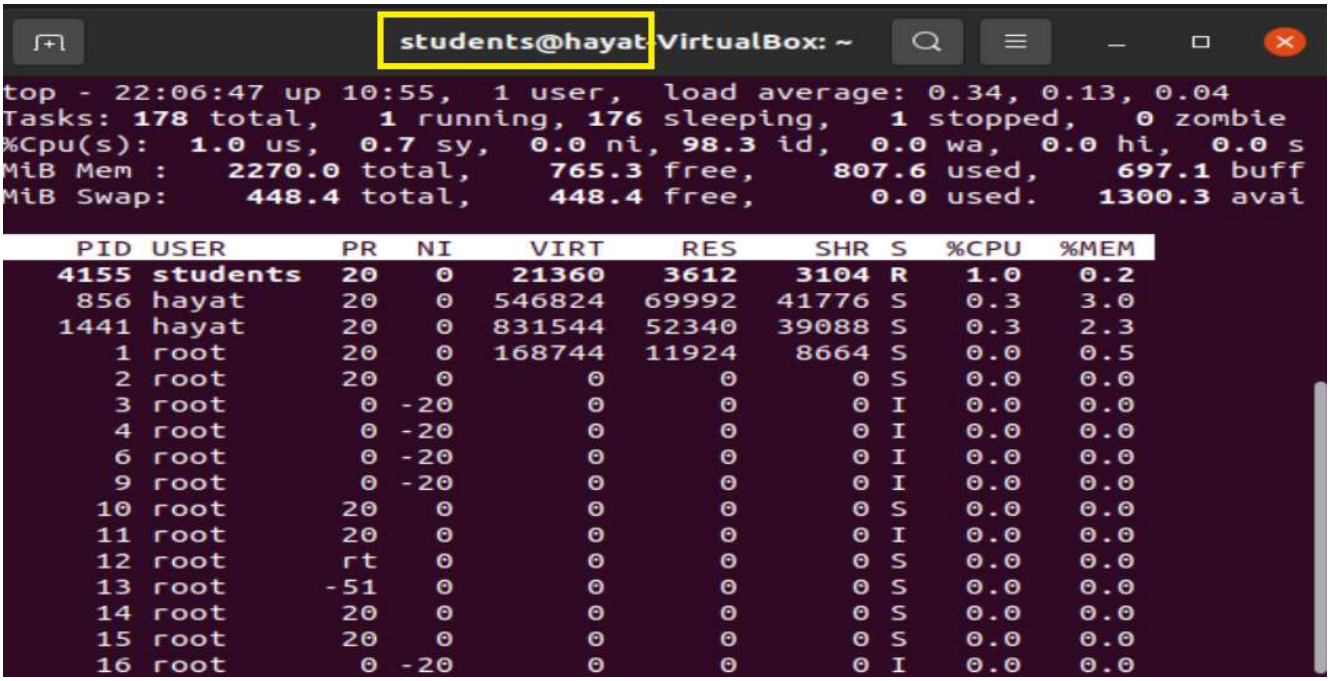


2.11) L'envoie du signal **SIGKIL=9** au processus **top**

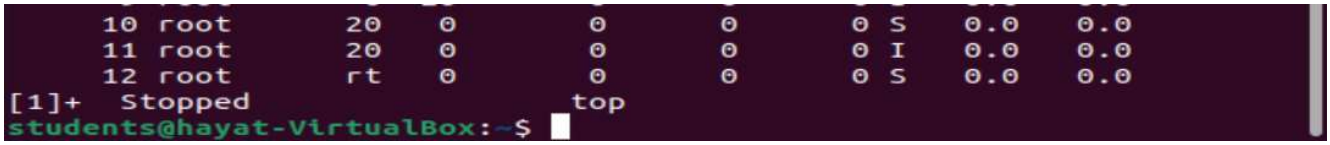


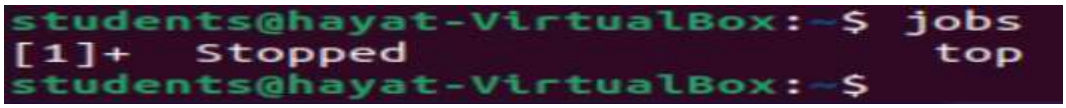
Exercice 3 : *« Gestion des jobs »*

3.1 Lancement de la commande **top**, on ne peut pas exécuter une nouvelle commande depuis la fenêtre dans laquelle on a lancé **top** :

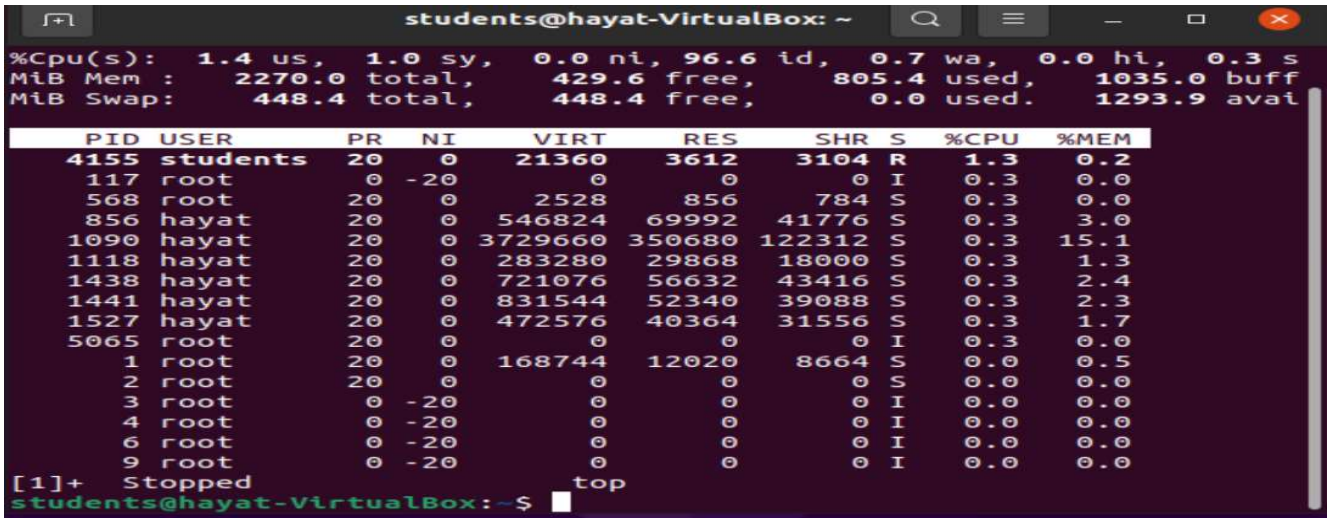


3.2 Après qu'on tape le Ctrl-z l'exécution de la commande **top** s'arrête**, la commande jobs nous permet de connaître les processus qui tournent en arrière-plan**



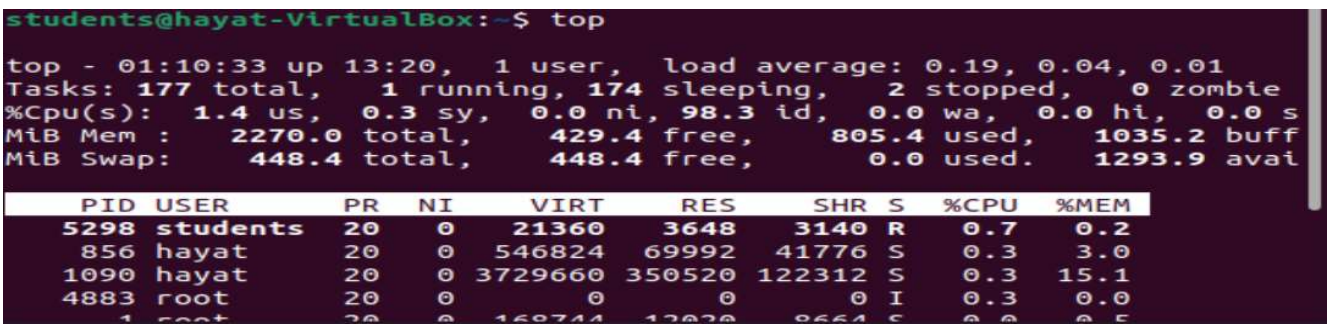


3.3 Lancement de la commande f**g** avec l'arrêt **de top par Ctrl Z**



3.4 Affichage de la commande : bg, non je n'arrive pas à lancer une nouvelle commande **via cette commande le processus de la commande top continue à s'exécuter mais en arrière plan**

3.5 Lancement du processus top



```
     12 root      rt   0       0      0      0 S   0.0   0.0
     13 root     -51   0       0      0      0 S   0.0   0.0
[2]+  Stopped                 top
students@hayat-VirtualBox:~$
```

3.6 La commande jobs permet de connaitre les processus qui tournent en arrière-plan

3.7 **Le 1% désigne le numéro de job du processus top**

```
students@hayat-VirtualBox:~$ kill -9 %1
[1]-  Killed                  top
students@hayat-VirtualBox:~$
```

3.8 Les voilà

```
students@hayat-VirtualBox:~$ ps
    PID TTY          TIME CMD
   3254 pts/0    00:00:00 bash
   5298 pts/0    00:00:03 top
   5370 pts/0    00:00:00 ps
students@hayat-VirtualBox:~$
```

3.9 Voici les différences entre les 2 commandes :

```
students@hayat-VirtualBox:~$ ps
    PID TTY          TIME CMD
   3254 pts/0    00:00:00 bash
   5298 pts/0    00:00:03 top
   5370 pts/0    00:00:00 ps
students@hayat-VirtualBox:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
students    3254    3253  0 21يناير pts/0 00:00:00 -bash
students    5298    3254  0 01:10 pts/0    00:00:03 top
students    5402    3254  0 01:29 pts/0    00:00:00 ps -f
students@hayat-VirtualBox:~$
```

3.10 le PID du processus top restant est : **5298**

3.11 Cette commande affiche les processus en cours d'exécution par tous les users

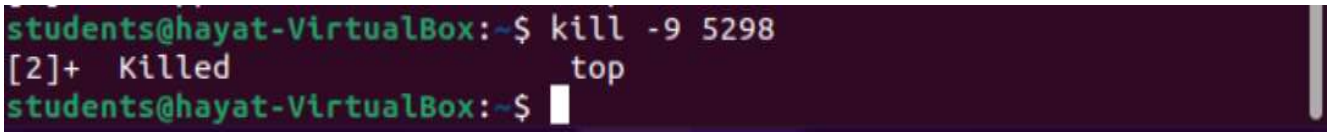
```
students@hayat-VirtualBox:~$ ps -af
UID          PID    PPID  C STIME TTY          TIME CMD
hayat        856     851  0 21يناير tty2 00:04:24 /usr/lib/xorg/
hayat        946     851  0 21يناير tty2 00:00:00 /usr/libexec/g
root        2279    1479  0 21يناير pts/0 00:00:00 sudo chown st
root        3253    1479  0 21يناير pts/0 00:00:00 su - students
students    3254    3253  0 21يناير pts/0 00:00:00 -bash
students    5298    3254  0 01:10 pts/0    00:00:03 top
students    5431    3254  0 01:36 pts/0    00:00:00 ps -af
students@hayat-VirtualBox:~$
```

3.12 Par exemple **3253 est le pid du processus su – students lancé par l'utilisateur root**

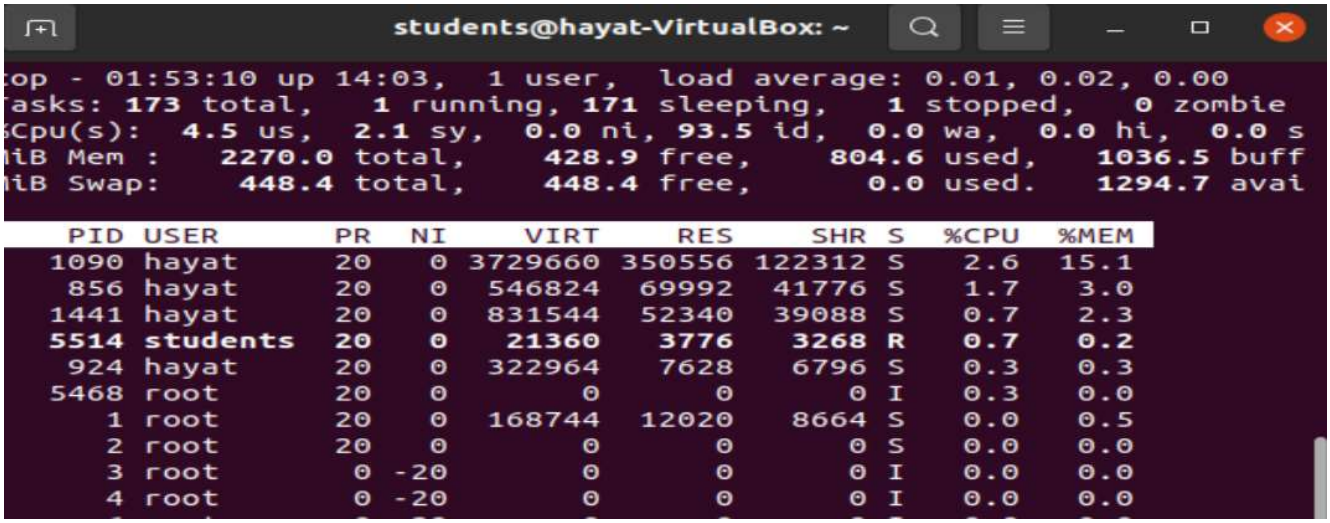
donc un processus de root on a pas le droit de le tuer

```
students@hayat-VirtualBox:~$ kill -9 3253
-bash: kill: (3253) - Operation not permitted
```

3.13 le voilà :



3.14 Non, on ne peut pas exécuter d'autre commande lors lancement du top



3.15 le voila

3.16 voilà le lancement

- ✓ **Bg : permet de faire passer le processus en arrière-plan (*background*)**
- ✓ **Fg : permt de reprendre un processus au premier plan (*foreground*)**