

UNIX - Introduction

X. Roirand – H. Evenas

Prénom Nom : CARRE

Date : 08/11/24

Groupe : 1D2

Présentation

Unix est né dans les laboratoires BELL qui dépendent d'AT&T (American Telephone & Telegraph) sous l'impulsion de 2 personnes :

- Ken Thompson
- Dennis Ritchie

Unix désigne une famille de systèmes d'exploitation non propriétaires, issus plus ou moins directement de différentes versions diffusées par AT&T. Certains membres de cette famille sont de complètes réécritures qui ne doivent rien à AT&T tout en restant compatibles au niveau des interfaces.

L'originalité de la démarche ayant présidé à sa conception repose sur deux aspects :

- le système a été conçu par des utilisateurs pour leurs propres besoins
- la conception et le développement ont été faits par une très petite équipe de très haut niveau scientifique.

Une première version, mono-utilisateur, a été réalisée en assembleur au cours des années 1969-1970, suivie dès 1971 d'une nouvelle version autorisant la multiprogrammation. L'objectif initial était essentiellement de fournir sur des petites machines un système interactif offrant un environnement de programmation comparable à ceux des grands systèmes.

Unix fut réécrit pour 90% en C en 1973. C'était le premier système d'exploitation à ne pas être entièrement écrit en assembleur. C'était aussi le premier système, en principe, portable.

L'année 1975 marque la date de première commercialisation du système.

Le système Unix est maintenant devenu une "norme" en matière de SE (système d'exploitation) et a été installé sur de nombreuses machines qui vont du micro-ordinateur au super-calculateur. Il est "simple" et "facile" à utiliser, propose un bon environnement de programmation et possède des outils dont ne disposent pas toujours des systèmes beaucoup plus performants.

Unix s'appuie sur trois grands principes :

1 - Le système de fichiers

Le système de fichiers permet à l'utilisateur de conserver les informations en les nommant. Il doit supporter les pannes matérielles et doit comporter un mécanisme de sécurité contre les accès non autorisés. Il distingue trois types de fichiers différents :

- le fichier ordinaire qui est une suite de caractères
- le répertoire qui contient des noms de fichiers ou de répertoires
- les fichiers spéciaux qui correspondent aux périphériques d'entrée/sortie.

2 - Les processus

Tous les travaux des utilisateurs sont réalisés par des processus.

Un processus est une séquence d'actions qui est caractérisée par la taille mémoire nécessaire à son exécution et par les fichiers auxquels il accède.

3 - Le shell

Le shell est un langage de programmation qui permet d'accéder au SE. Il exécute des commandes qui peuvent provenir soit d'un terminal, soit d'un fichier.

En résumé, Unix présente les caractéristiques suivantes :

- multi-utilisateurs
plusieurs utilisateurs connectés simultanément se partagent les ressources par découpage du temps et mise à profit des temps morts d'entrées/sorties ; le système possède un mécanisme d'identification et de protection des utilisateurs
- multi-tâches
un utilisateur peut lancer plusieurs tâches en même temps
- système de fichiers hiérarchisé arborescent
les entrées/sorties sont généralisées, vues par l'utilisateur à travers des périphériques considérés comme des fichiers

Principales commandes simplifiées (pour plus de détails voir l'Aide en ligne : commande *man* ou *help*) :

Connexion/Deconnexion

login	connexion login [nom utilisateur]
exit	quitte l'interpréteur de commandes et permet une déconnexion exit
su	changement d'identité ("-" force un nouvel environnement) su [-] id_utilisateur

Manipulation de répertoires

pwd	affiche le nom du répertoire courant pwd
cd	change le répertoire courant cd [[chemin d'accès] répertoire]
mkdir	crée un répertoire mkdir [chemin d'accès] répertoire
rmdir	supprime un sous-répertoire (s'il est vide) rmdir [chemin_d'accès] répertoire
ls	affiche la liste des fichiers et sous-répertoires d'un répertoire ls [chemin d'accès]
pushd	empile le chemin du répertoire courant et se déplace dans le répertoire indiqué pushd . (empile le chemin du répertoire courant et y reste)
popd	dépile le chemin du répertoire enregistré avec pushd et y retourne popd

Manipulation de fichiers

cat	affiche le contenu d'un fichier texte cat [chemin d'accès] [fichier]
touch	change les dates d'accès et de modification d'un fichier ou en crée un nouveau
chgrp	changement du groupe d'un fichier chgrp groupe [chemin d'accès] fichier
chmod	modifie la protection (droits) d'un fichier chmod mode [chemin d'accès] fichier
chown	changement du propriétaire d'un fichier chown propriétaire [chemin_d'accès] fichier
cp	copie un fichier source dans un fichier destination cp [chemin_d'accès] f_source {chemin_d'accès [f destination] f_destination}
cut	découpe une ligne en champ numéroté de 1 à n suivant un délimiteur cut -d[délimiteur] -f[numéro du champ] [chemin d'accès] fichier
head	affiche le début d'un fichier head [-nombre de lignes] [chemin d'accès] fichier
more	affiche le contenu d'un fichier texte page par page more [chemin d'accès] fichier
mv	change le nom d'un fichier ou le déplace mv [chemin d'accès] f_source {chemin_d'accès [f_destination] f_destination}
pg	similaire à more
rm	supprime un ou plusieurs fichiers, et des sous-répertoires rm [chemin d'accès] fichier
tail	affiche le début ou la fin d'un fichier tail [{+} -]nombre [lbc] [chemin d'accès] fichier
umask	positionnement des droits par défaut umask [mode]
wc	compte les mots, lignes ou caractères d'un fichier wc [chemin d'accès] fichier

Gestion des processus

ps	affichage les informations sur les processus en mémoire ps
kill	envoie d'un signal kill [-signal] n°_processus

Gestion des utilisateurs

who	affiche tous les utilisateurs connectés sur une machine who
whoami	renvoie les information sur l'utilisateur connecté whoami
groups	affiche les groupes d'appartenance de l'utilisateur connecté groups

Commande diverses

man	affiche de l'information sur les commandes Unix man commande
clear	efface l'écran clear
env	affiche les variables d'environnement env
alias	définition de synonymes alias [nom_alias=commande]
ln	lie un fichier à un autre ln [chemin_d'accès] fichier lié [[chemin_d'accès] lien]
grep	recherche d'une chaîne dans un fichier fichier grep expression [[chemin_d'accès] fichier]
diff	compare les contenus de deux fichiers diff [chemin d'accès] fichier 1 [chemin d'accès] fichier2
date	affiche ou modifie la date courante date
nano, vi, emacs	lance l'éditeur
sed	éditeur en mode ligne de commande
du	espace occupé sur le disque
find	recherche d'un fichier
less, more	filtre permettant l'affichage page par page

Nommage des fichiers et des répertoires

Les noms sont limités à 14 caractères au plus et une différenciation minuscule/majuscule est faite.

Les caractères à proscrire sont les suivants :

\ > < | \$? & [] * ! " ' ' () @ ~

Le point (".") joue un rôle particulier dans la mesure où les fichiers dont le nom commence par lui sont "invisibles" (ils sont dits "cachés").

TP

Tous les exercices sont à réaliser en ligne de commande dans un terminal (en utilisant l'application *Terminal*). Pour toutes les commandes à taper, il faut taper le nom de la commande, puis <Entrée> ou <Return> afin que l'interpréteur de commande, ici le shell, puisse prendre en compte votre commande.

1/ Analysez la description de la commande *grep* et déduisez-en le sens de [...]

Pour avoir la signification de la commande, utilisez la commande *man*, en tapant :

man ...

ou ... est le nom de la commande pour laquelle on veut le manuel

[...] signifie : *c'est argument qui est optionnel lors de l'écriture de la commande*

2/ Familiarisation avec l'usage des commandes UNIX : pour chacune des commandes données ci-après, précisez la signification de son abréviation (tout d'abord en anglais pour le nom complet, puis en français pour le rôle) puis décrivez son usage.

man, clear, cp, rm, mv, pwd, cd, mkdir, rmdir, ls, cat, touch

Exemple : *man* est mis pour *manual* (manuel) et lance l'aide en ligne

Vous donnerez le résultat sous la forme suivante :

Nom de commande (en général abréviation)	Nom complet	Rôle
man	manual	<i>Pour trouver des informations sur un commande</i>
clear	<i>clear</i>	<i>Effacer l'écran actuelle</i>
cp	<i>copy</i>	<i>Copie un fichier source dans un fichier de destination</i>
rm	<i>remove</i>	<i>Supprime un ou plusieurs fichiers/sous-répertoires</i>
mv	<i>move</i>	<i>Déplace ou change un fichier</i>
pwd	<i>print working directory</i>	<i>Affiche le nom du répertoire courant</i>
cd	<i>change directory</i>	<i>Permet de se déplacer dans les répertoires (de changer de répertoire courant)</i>
mkdir	<i>make directory</i>	<i>Permet de créer un répertoire</i>

rmdir	<i>remove directory</i>	<i>Permet de retirer un répertoire s'il est vide</i>
ls	<i>list</i>	<i>Permet de lister les répertoires et fichiers d'un répertoire</i>
cat	<i>concatenate</i>	<i>Permet d'afficher le contenu d'un fichier texte</i>
touch	touch	<i>Change les dates d'accès d'un fichier ou permet d'en créer un nouveau</i>

3/ Indiquez la validité ou l'invalidité des différents noms de fichier proposés ci-après (en cas d'invalidité indiquez-en la cause) :

Essai
123essai.txt
<essai>
Essai.123essai
.123essai

Pour réaliser cela, vous utiliserez la commande *touch* suivi du nom du fichier à tester, puis vous vérifierez que ce fichier a été ou non créé grâce à la commande *ls*.

touch
ls ...

<i>touch Essai</i>	<i>ça marche, le fichier est créé</i>
<i>touch 123essai.txt</i>	<i>ça marche, le fichier est créé</i>
<i>touch <essai></i>	<i>ça ne marche pas, à cause des symboles « < » et « > »</i>
<i>touch Essai.123essai</i>	<i>ça marche, le fichier est créé</i>
<i>touch .123essai</i>	<i>ça marche, le fichier est créé mais ne se voit pas avec un simple ls et se voit avec ls -a</i>

On considère pour la question 4, que la position de départ est votre répertoire d'accueil. Si vous n'êtes pas sûr de votre répertoire de départ, vous pouvez taper la commande :

cd

4/ Créez à partir de votre répertoire courant, un répertoire appelé SYS et vérifiez son existence à l'aide de la commande *ls*.

Indiquez la commande utilisée pour créer le répertoire :

mkdir SYS
ls

5/ Déplacez-vous dans le répertoire SYS et créez-y 2 répertoires appelés *Documents* et *Images*, puis vérifiez leur existence.

Indiquez la ou les commandes utilisées pour créer les 2 répertoires et la commande utilisée pour vérifier l'existence de ces répertoires :

```
cd SYS/  
mkdir Documents  
mkdir Images  
ls
```

6/ Déplacez-vous dans Documents et créez-y un fichier appelé *mondoc.txt*

Pour créer ce fichier qui sera vide, vous pouvez simplement utiliser la commande *touch* qui permet de créer un fichier vide avec le nom que vous voulez. Chercher avec man ou sur internet comment utiliser la commande touch.

Indiquez la commande utilisée pour créer le fichier *mondoc.txt* :

```
cd Documents/  
touch mondoc.txt
```

7/ Affichez le contenu du fichier *mondoc.txt* (souvenez-vous qu'il est vide...)

Pour afficher le contenu d'un fichier, il existe plusieurs commandes. La 1^{ère} affiche le contenu complet du fichier. Donc si il est long l'affichage peut prendre un peu de temps. Les 2 suivantes affichent le fichier avec une pagination. C'est à dire que si le contenu du fichier dépasse la capacité d'affichage de l'écran utilisé (nombre de ligne) alors une pause est faite et il faut appuyer sur entrée pour afficher la ligne suivante ou sur espace pour afficher la page suivante.

Donnez les 3 commandes possibles.

```
cat mondoc.txt  
head mondoc.txt  
more mondoc.txt
```

8/ Créez un fichier appelé *image.png* dans le répertoire *Documents*

Bien-sûr le fichier *image.png* ne sera pas un vrai fichier image, il aura juste le nom.

Indiquez la commande utilisée pour créer le fichier *image.png* :

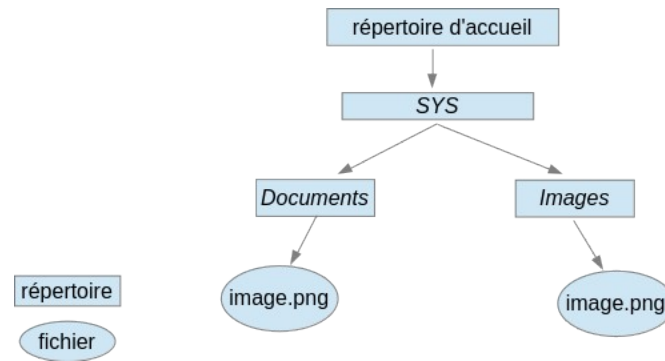
```
touch image.png
```

9/ Copiez le fichier *image.png* du répertoire *Documents* dans le répertoire *Images*

Indiquez la commande utilisée pour copier le fichier *image.png* :

```
cp image.png ../Images/
```

Après toutes ces commandes que vous avez tapées, l'arborescence de vos répertoires et fichiers peut être dessinée comme ceci :



A noter que le fichier `mondoc.txt` est aussi présent dans le répertoire `Documents`.

10/ Afficher l'arborescence complète à partir de votre répertoire d'accueil.

L'utilitaire `tree` permet d'afficher les arborescences complètes.

Indiquez la commande utilisée pour afficher l'arborescence :

Note : Il se peut que `tree` ne soit pas installé sur votre ordinateur, dans ce cas il faut l'installer avec la commande « `sudo apt install tree` »

`sudo apt install tree`
`tree SYS`

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ tree SYS
SYS
├── Documents
│   ├── image.png
│   └── mondoc.txt
└── Images
    └── image.png
```

11/ Supprimez le fichier `image.png` du répertoire `Documents`

Indiquez la commande utilisée pour supprimer le fichier `image.png` :

`rm SYS/Documents/image.png`

12/ Affichez le contenu du répertoire `Documents`

Indiquez la commande utilisée pour afficher le contenu du répertoire :

`ls SYS/Documents/`

13/ Sans changer de répertoire, affichez le contenu du répertoire `Images`

Indiquez la commande utilisée pour afficher le contenu du répertoire `Images` :

`ls SYS/Images/`

14/ Cherchez avec la commande `man` comment afficher le contenu du répertoire `Images` sur une seule colonne

Indiquez la commande utilisée pour afficher le contenu du répertoire Images sur une seule colonne :

ls -1 SYS/Images/

Le point de départ de l'arborescence du système est un répertoire particulier qui s'appelle la racine (/). Votre répertoire d'accueil n'est qu'une sous-partie de l'arborescence globale.

15/ Depuis votre répertoire courant lancez la commande qui permet de visualiser le contenu de la racine du système

Indiquez la commande utilisée pour afficher le contenu de la racine système :

ls ../..

16/ Déplacez-vous à la racine du système et lancez cette même commande, mais sans argument. Que constatez-vous ?

ls
Ce sont les mêmes fichiers et répertoires qui sont affichés

17/ Y a t-il un intérêt à se déplacer dans le répertoire dont on veut lister le contenu ou pas ?

Non car l'on connaît le chemin d'accès on peut donc juste l'écrire

18/ Créez un fichier .test à la racine de votre home grâce à la commande touch.

Indiquez la commande utilisée pour créer le fichier :

touch home/aryouko/.test

19/ Utilisez la commande « ls » pour afficher le contenu de votre home, expliquez pourquoi vous ne voyez pas le fichier « .test » :

ls home/aryouko/
Les fichiers commençant par un « . » dans leur nom ne sont pas visibles avec un simple ls, ils sont dit « cachés » ou « invisibles ».

20/ Trouvez l'option dans la commande « ls » qui vous permet d'afficher le contenu de votre home mais qui fait bien apparaître le fichier « .test », et indiquez la commande utilisée :

ls -a home/aryouko/

21/ Retournez dans votre répertoire d'accueil aussi appelé « home directory » ou « home » en lançant la commande cd sans argument, puis, depuis ce répertoire, listez le contenu du répertoire SYS

Indiquez la commande utilisée pour afficher le contenu du répertoire SYS:

cd
ls SYS/

22/ Déplacez-vous dans SYS et testez la touche *Tab* pour la complémentation automatique (*ls D Tab* par exemple donnera *ls Documents*). Faites la même chose pour lister le répertoire Images. Manipulez les flèches montante et descendante pour visualiser l'historique des commandes.

ls Documents/
ls Images/

23/ Il existe une commande pour lister l'historique de toutes les commandes que vous avez déjà tapé dans un terminal (voir dans une session ou plusieurs sessions bash précédente). Trouvez cette commande et testez la.
Indiquez le nom et la syntaxe de cette commande.

La commande est History et ça syntaxe est :
history

24/ Essayer de rappeler une des commandes de l'historique, en utilisant « une astuce », qui permet d'exécuter une commande de l'historique sans avoir à retaper cette commande, mais juste son numéro.
Indiquez le nom et la syntaxe de cette commande.

!555
Le numéro étant la commande dans l'historique et le ! pour l'exécuter, c'est un désigne d'événement q permet de référencer à un événement passé dans l'historique

25/ Il existe une autre astuce (proche de la précédente), qui permet de rappeler une commande qui commence par la ou les même lettre qu'une précédente. Expliquez cette astuce.

!string string étant le début d'une commande et prenant la plus récente ressemblant

26/ Expliquez en quoi, dans certains cas, cette astuce peut-être dangereuse.

Si on se trompe de commande ou qu'une commande est pris malencontreusement à la place d'une autre

27/ Lancez une à une les différentes commandes nécessaires pour supprimer de votre espace toute l'arborescence créée au cours de ces différents exercices

Indiquez les commandes utilisées:

rm SYS/Documents/mondoc.txt
rm SYS/Images/image.png
rmdir SYS/Documents/
rmdir SYS/Images/
rmdir SYS/
rm 123essai.txt
rm Essai

rm Essai.123essai
rm .123essai
rm .test

Partie bonus

28/ Dans votre répertoire d'accueil créez le fichier *essai.txt* puis cherchez un moyen de le détruire **en forçant l'apparition de la demande de confirmation** (pensez à l'aide en ligne)

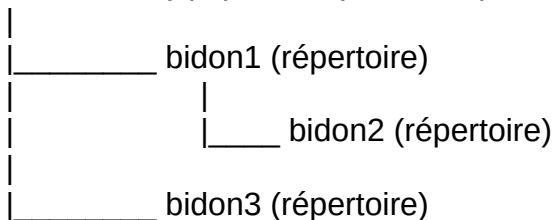
Indiquez la commande utilisée :

rm -i essai.txt

29/ Créez l'arborescence donnée ci-dessous (bidon1, bidon2 et bidon3 sont des répertoires vides), puis détruisez-la en une seule commande **en forçant l'apparition de la demande de confirmation** (utilisez à nouveau l'aide en ligne de la commande *rm*)

Pensez à regarder le manuel de *rm* (command *man*) pour voir comment supprimer des répertoires.

<home directory(répertoire personnel)>



Indiquez la commande utilisée :

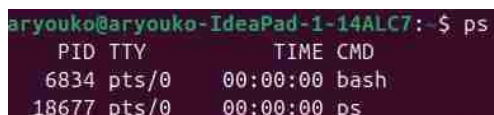
rm -i -r bidon1/ bidon3/

Travail sur les process :

Pour ce travail, vous ferez une copie d'écran pour chaque réponse avec les 10/15 premières lignes de la sortie écran de la commande ou tout si il y a moins de 10/15 lignes de sortie écran.

30/ Connaître la hierarchie des process est utile. Quelle commande pouvez-vous utiliser pour afficher la liste des process dans votre terminal ?

ps



```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps
  PID TTY          TIME CMD
  6834 pts/0    00:00:00 bash
 18677 pts/0    00:00:00 ps
```

31/ Trouvez l'option de *ps* qui va bien et donnez la commande pour afficher l'ensemble des process pour un utilisateur donné, en l'occurrence, l'utilisateur root ?

ps -U root -u root u

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps -U root -u root u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2 23948 14988 ?        Ss   09:45   0:03 /sbin/init splash
root         2  0.0  0.0      0     0 ?        S    09:45   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    09:45   0:00 [pool_workqueue_release]
root         4  0.0  0.0      0     0 ?        I<   09:45   0:00 [kworker/R-rcu_g]
root         5  0.0  0.0      0     0 ?        I<   09:45   0:00 [kworker/R-rcu_p]
root         6  0.0  0.0      0     0 ?        I<   09:45   0:00 [kworker/R-slub_]
root         7  0.0  0.0      0     0 ?        I<   09:45   0:00 [kworker/R-netns]
root        10  0.0  0.0      0     0 ?        I<   09:45   0:00 [kworker/0:0H-events_hig]
root        12  0.0  0.0      0     0 ?        I<   09:45   0:00 [kworker/R-mm_pe]
root        13  0.0  0.0      0     0 ?        I    09:45   0:00 [rcu_tasks_kthread]
root        14  0.0  0.0      0     0 ?        I    09:45   0:00 [rcu_tasks_rude_kthread]
root        15  0.0  0.0      0     0 ?        I    09:45   0:00 [rcu_tasks_trace_kthread]
root        16  0.0  0.0      0     0 ?        S    09:45   0:00 [ksoftirqd/0]
```

32/ On a toujours la possibilité de chercher le PPID (parent id) d'un process et ensuite de chercher les infos pour le PPID obtenu, et ainsi de suite pour connaître toute l'arborescence d'un process donné mais il existe une option qui permet de faire ça dans ps, trouvez là et faite la commande qui permet d'afficher l'arborescence (donc sous forme d'arbre) de tout le système. ATTENTION, il faut voir l'ensemble des process qui tournent sur votre machine, donc rajouter l'option pour afficher sous forme d'un arbre n'est pas assez, il faut en plus rajouter une option supplémentaire.

ps -A --forest ou ps -e --forest

-A : voir tout les process --forest : pour l'avoir en arbre

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps -e --forest
  PID TTY          TIME CMD
    2 ?           00:00:00 kthreadd
    3 ?           00:00:00 \_ pool_workqueue_release
    4 ?           00:00:00 \_ kworker/R-rcu_g
    5 ?           00:00:00 \_ kworker/R-rcu_p
    6 ?           00:00:00 \_ kworker/R-slub_
    7 ?           00:00:00 \_ kworker/R-netns
   10 ?           00:00:00 \_ kworker/0:0H-events_highpri
   12 ?           00:00:00 \_ kworker/R-mm_pe
   13 ?           00:00:00 \_ rcu_tasks_kthread
   14 ?           00:00:00 \_ rcu_tasks_rude_kthread
   15 ?           00:00:00 \_ rcu_tasks_trace_kthread
   16 ?           00:00:00 \_ ksoftirqd/0
   17 ?           00:00:17 \_ rcu_preempt
```

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ pstree
systemd--ModemManager--3*[{ModemManager}]
--NetworkManager--3*[{NetworkManager}]
--accounts-daemon--3*[{accounts-daemon}]
--apache2--5*[apache2]
--avahi-daemon--avahi-daemon
--bluetoothd
--colord--3*[{colord}]
--cron
--cups-browsed--3*[{cups-browsed}]
--cupsd
--dbus-daemon
--fwupd--5*[{fwupd}]
--gdm3--gdm-session-wor--gdm-wayland-ses--gnome-session-b--3*[{gnom+}
```

33/ On a parfois des process qui sont très consommateur de CPU ou de mémoire et il est utile de pouvoir les identifier. Donnez la commande qui permet d'afficher la liste de tous les process de la machine, en triant par occupation CPU, du plus gros consommateur au plus petit consommateur.

ps -e -o pid,comm,%cpu --sort=-%cpu

Ici c'est le --sort=-%cpu qui permet de trier le reste est de l'affichage avec -o pour les colonnes

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps -e -o pid,comm,%cpu --sort=-%cpu
```

PID	COMMAND	%CPU
18136	soffice.bin	4.2
18021	Isolated Web Co	2.2
2531	gnome-shell	1.0
18223	gnome-terminal-	0.8
18695	kworker/u24:3-g	0.5
3804	firefox	0.4
19103	kworker/u26:4-f	0.2
19105	kworker/u26:6-e	0.2
18207	kworker/u26:0-e	0.1
18487	eog	0.1
238	irq/33-ELAN06FA	0.1
17124	kworker/u26:2-e	0.1
17863	kworker/u24:2-g	0.1

32/ Donnez une commande similaire mais qui affiche tous les process de la machine, triés par occupation mémoire, du plus petit au plus grand.

ps -e -o pid,comm,%cpu -sort=+%mem

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps -e -o pid,comm,%mem --sort=+%mem
```

PID	COMMAND	%MEM
2	kthreadd	0.0
3	pool_workqueue_	0.0
4	kworker/R-rcu_g	0.0
5	kworker/R-rcu_p	0.0
6	kworker/R-slub_	0.0
7	kworker/R-netns	0.0
10	kworker/0:0H-ev	0.0
12	kworker/R-mm_pe	0.0
13	rcu_tasks_kthre	0.0
14	rcu_tasks_rude_	0.0
15	rcu_tasks_trace	0.0
16	ksoftirqd/0	0.0
17	rcu_preempt	0.0

33/ Pour la prochaine question, nous avons besoin de connaître le PID du shell courant de votre terminal, trouvez comment afficher le PID de votre shell courant ?

Echo \$\$, \$\$ pour le PID du shell courant

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ echo $$
18231
```

34/ Maintenant, affichez, toujours avec la commande ps, le temps d'exécution de votre shell depuis qu'il a été créé ?

ps -p \$\$ -o etime

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps -p $$ -o etime
ELAPSED
29:27
```

35/ En utilisant le programme top, donnez la signification des 3 valeurs affichées pour le « load average » ?

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ top
```

top - 17:27:27 up 1 day, 3:29, 1 user, load average: 0.34, 0.33, 0.32
Tâches: 342 total, 1 en cours, 341 en veille, 0 arrêté, 0 zombie
%Cpu(s): 0.3 ut, 0.2 sy, 0.0 ni, 99.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 5767.5 total, 376.3 libr, 3113.2 util, 2721.6 tamp/cache
MiB Éch: 4096.0 total, 4095.7 libr, 0.3 util. 2654.3 dispo Mem

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
2531	aryouko	20	0	5448916	364112	165500	S	3.0	6.2	19:02.96	gnome-s+
18223	aryouko	20	0	635932	59052	45520	S	2.0	1.0	0:11.07	gnome-t+
238	root	-51	0	0	0	0	S	0.7	0.0	2:27.79	irq/33-+
18657	aryouko	20	0	2570388	186020	90600	S	0.7	3.1	0:08.64	Isolate+
21182	root	20	0	0	0	0	I	0.7	0.0	0:00.80	kworker+
21407	aryouko	20	0	23404	6016	3840	R	0.7	0.1	0:00.57	top

Première valeur : charge moyenne du système sur la dernière minute
Deuxième valeur : charge moyenne du système sur les 5 dernières minutes
Troisième valeur : charge moyenne du système sur les 15 dernières minutes

36/ Toujours avec ce programme, donnez le nombre de process qui :

- dorment (sont en sleep) 341 sont en veille
- sont arrêtés 0 sont arrêtés
- sont des zombies 0 sont zombies
- s'exécutent 1 est en cours

36/ Toujours avec top, indiquez comment trier par ordre décroissant d'occupation du processeur ?

top -o=+%CPU ou SHIFT + P quand on est dans le top

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ top -o=+%CPU
top - 17:37:26 up 1 day, 3:39, 1 user, load average: 0,31, 0,28, 0,31
Tâches: 327 total, 1 en cours, 326 en veille, 0 arrêté, 0 zombie
%Cpu(s): 0,7 ut, 0,4 sy, 0,0 ni, 98,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 5767,5 total, 364,7 libr, 3122,4 util, 2723,9 tamp/cache
MiB Éch: 4096,0 total, 4095,7 libr, 0,3 util. 2645,1 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
2531	aryouko	20	0	5442776	364256	165584	S	6,0	6,2	19:31.78	gnome-s+
18223	aryouko	20	0	635932	59180	45520	S	4,3	1,0	0:13.80	gnome-t+
238	root	-51	0	0	0	0	S	1,7	0,0	2:31.65	irq/33+
20016	aryouko	20	0	2401052	73720	61324	S	0,7	1,2	0:00.52	Web Con+
21182	root	20	0	0	0	0	I	0,7	0,0	0:02.41	kworker+
835	systemd+	20	0	17556	7632	6728	S	0,3	0,1	0:11.63	systemd+

37/ Puis trouvez comment trier par ordre mémoire décroissant ?

top -o=+%MEM ou SHIFT + M quand on est dans le top

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ top -o=+%MEM
top - 17:46:12 up 1 day, 3:48, 1 user, load average: 0,59, 0,39, 0,33
Tâches: 329 total, 1 en cours, 328 en veille, 0 arrêté, 0 zombie
%Cpu(s): 0,5 ut, 0,3 sy, 0,0 ni, 99,2 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 5767,5 total, 519,8 libr, 2965,0 util, 2680,9 tamp/cache
MiB Éch: 4096,0 total, 4095,7 libr, 0,3 util. 2802,5 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
3804	aryouko	20	0	11,3g	488824	252708	S	0,0	8,3	8:18.81	firefox
18136	aryouko	20	0	1546988	428524	168428	S	0,0	7,3	2:07.09	soffice+
2531	aryouko	20	0	5444612	366616	167548	S	4,3	6,2	20:00.05	gnome-s+
4540	aryouko	20	0	2616900	213816	148976	S	0,0	3,6	1:19.40	Isolate+
4024	aryouko	20	0	2768332	211136	95120	S	0,0	3,6	0:28.54	Privile+
18021	aryouko	20	0	2626616	200160	102332	S	0,0	3,4	0:39.01	Isolate+

38/ Lancez maintenant 2 terminaux. Dans l'un vous tapez la commande « tail -f », juste pour rendre le terminal bloqué et dans l'autre affichez la liste de vos process. Identifiez le PID et donnez le ?

ps -e | grep tail

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ ps -e | grep tail
23032 pts/1    00:00:00 tail
```


39/ Envoyez un signal d'interruption (SIGINT) sur ce process, pour cela servez-vous de la commande kill et de son manuel, rien d'autre. **Indiquez cette commande et la copie des 2 écrans.**

Kill -2 23032

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ kill -2 23032
```

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ tail -f
```

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$
```

40/ Essayez d'envoyer un même signal sur un process appartenant à root, **quel est le résultat ? Pourquoi ?**

```
aryouko@aryouko-IdeaPad-1-14ALC7:~$ kill -2 1
bash: kill: (1) - Opération non permise
```

On ne peut pas faire la même chose car on a pas les permissions, autrement dit on est pas l'utilisateur root pour l'exécuter il faudrait l'être ou mettre sudo devant.

41/ Ecrivez un script demande à l'utilisateur 2 nombres de manière interactive, l'un après l'autre et qui vérifie que chaque donnée entrée est bien un nombre valide, puis les additionne et affiche le résultat. **Donnez le contenu du script ici ?**

```
function Addition() {
    local verif=1
    while ((verif)); do
        read -p "Entrez un numéro : " int1
        if [[ "$int1" =~ ^-?[0-9]+([.][0-9]+)?$ ]]; then
            verif=0
        else
            echo "Entrée invalide. Veuillez entrer un nombre entier."
        fi
    done
    verif=1
    while ((verif)); do
        read -p "Entrez un numéro : " int2
        if [[ "$int1" =~ ^-?[0-9]+([.][0-9]+)?$ ]]; then
            verif=0
        fi
    done
    local res=$(echo "$int1 + $int2" | bc)
    echo "Le résultat est : $res"
}
Addition
```