

# Système, Script et Sécurité

## 1. Création d'une VM Debian

La création d'un utilisateur avec une majuscule n'étant pas autorisée lors de l'installation, j'ai commencé par créer un compte titou avec comme mdp admin.

- suppression de l'utilisateur titou via **"userdel -r "**

```
root@debian:~# userdel -r titou
```

- création de l'utilisateur la plateforme via **"useradd "** et création du mot de passe via **"passwd "**

```
root@debian:~# useradd La_Plateforme
root@debian:~# passwd La_Plateforme
Nouveau mot de passe :
Retapez le nouveau mot de passe :
Les mots de passe ne correspondent pas.
Mot de passe : Erreur de manipulation du jeton d'authentification
passwd : mot de passe inchangé
root@debian:~# passwd La_Plateforme
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
root@debian:~# █
```

Les étapes suivantes ont été nécessaires car le home de l'utilisateur n'a pas été créé avec useradd, de ce fait je n'arrivais pas à me connecter à ce nouvel utilisateur.

Création du Shell de l'utilisateur.

```
root@debian:~# cat /etc/passwd
```

```
root@debian:~# usermod -s /bin/bash La_Plateforme
```

Pour finir la création de son home.

```
root@debian:~# ls -ld /home/La_Plateforme
ls: impossible d'accéder à '/home/La_Plateforme': Aucun fichier ou dossier de ce type
root@debian:~# mkdir /home/La_Plateforme
root@debian:~# chown La_Plateforme:La_Plateforme /home/La_Plateforme
root@debian:~# chmod 700 /home/La_Plateforme
root@debian:~# █
```

## 2.Commandes de recherche avancée

Création du fichier via “**nano**”

Copie du fichier dans les divers dossier via “**cp**” suivi du répertoire de destination

Puis utilisation de “**mv**” pour déplacer le fichier initial dans le dernier dossier

```
La_Plateforme@debian:~$ nano mon_texte.txt
La_Plateforme@debian:~$ cp mon_texte.txt /Bureau
cp: impossible de créer le fichier standard '/Bureau': Permission non accordée
La_Plateforme@debian:~$ cp mon_texte.txt ~/Bureau
La_Plateforme@debian:~$ cp mon_texte.txt ~/Documents
La_Plateforme@debian:~$ cp mon_texte.txt ~/Téléchargements
La_Plateforme@debian:~$ cp mon_texte.txt ~/Vidéos
La_Plateforme@debian:~$ mv mon_texte.txt ~/Images
La_Plateforme@debian:~$
```

Recherche des fichiers contenant le mot “force” via “**grep**”

```
La_Plateforme@debian:~$ grep -r "force" .
./Bureau/mon_texte.txt:Que la force soit avec toi
./Téléchargements/mon_texte.txt:Que la force soit avec toi
./Vidéos/mon_texte.txt:Que la force soit avec toi
./Documents/mon_texte.txt:Que la force soit avec toi
./Images/mon_texte.txt:Que la force soit avec toi
La_Plateforme@debian:~$
```

### 3.Compression et décompression de fichiers

- création du répertoire Plateforme

```
La_Plateforme@debian:~$ mkdir Plateforme
```

- copie du fichier mon\_texte.txt

```
La_Plateforme@debian:~$ cd Images/  
La_Plateforme@debian:~/Images$ cp mon_texte.txt ~/Plateforme  
La_Plateforme@debian:~/Images$ cd ..  
La_Plateforme@debian:~$ cd Plateforme/  
La_Plateforme@debian:~/Plateforme$ ls -l  
total 4  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars 10:07 mon_texte.txt
```

- duplication dans le dossier Plateforme

```
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars 10:07 mon_texte.txt  
La_Plateforme@debian:~/Plateforme$ cp mon_texte.txt mon_texte1.txt  
La_Plateforme@debian:~/Plateforme$ cp mon_texte.txt mon_texte2.txt  
La_Plateforme@debian:~/Plateforme$ cp mon_texte.txt mon_texte3.txt  
La_Plateforme@debian:~/Plateforme$ ls -l  
total 16  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars 10:14 mon_texte1.txt  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars 10:15 mon_texte2.txt  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars 10:15 mon_texte3.txt  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars 10:07 mon_texte.txt  
La_Plateforme@debian:~/Plateforme$
```

- création de l'archive
- **tar** : L'utilitaire de création d'archives.
- **-czvf** : Les options utilisées avec **tar** :
  - **c** : Crée une nouvelle archive.
  - **z** : Comprime l'archive avec gzip.
  - **v** : Affiche les détails pendant l'archivage.
  - **f** : Spécifie le nom du fichier d'archive.

```
La_Plateforme@debian:~$ tar -czvf Plateforme.tar.gz Plateforme/
Plateforme/
Plateforme/mon_texte2.txt
Plateforme/mon_texte.txt
Plateforme/mon_texte1.txt
Plateforme/mon_texte3.txt
La_Plateforme@debian:~$ ls -l
total 40
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:59 Bureau
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:59 Documents
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 10:00 Images
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:47 Modèles
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:47 Musique
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 10:15 Plateforme
-rw-r--r-- 1 La_Plateforme La_Plateforme 244 18 mars 10:17 Plateforme.tar.gz
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:47 Public
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:59 Téléchargements
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars 09:59 Vidéos
La_Plateforme@debian:~$
```

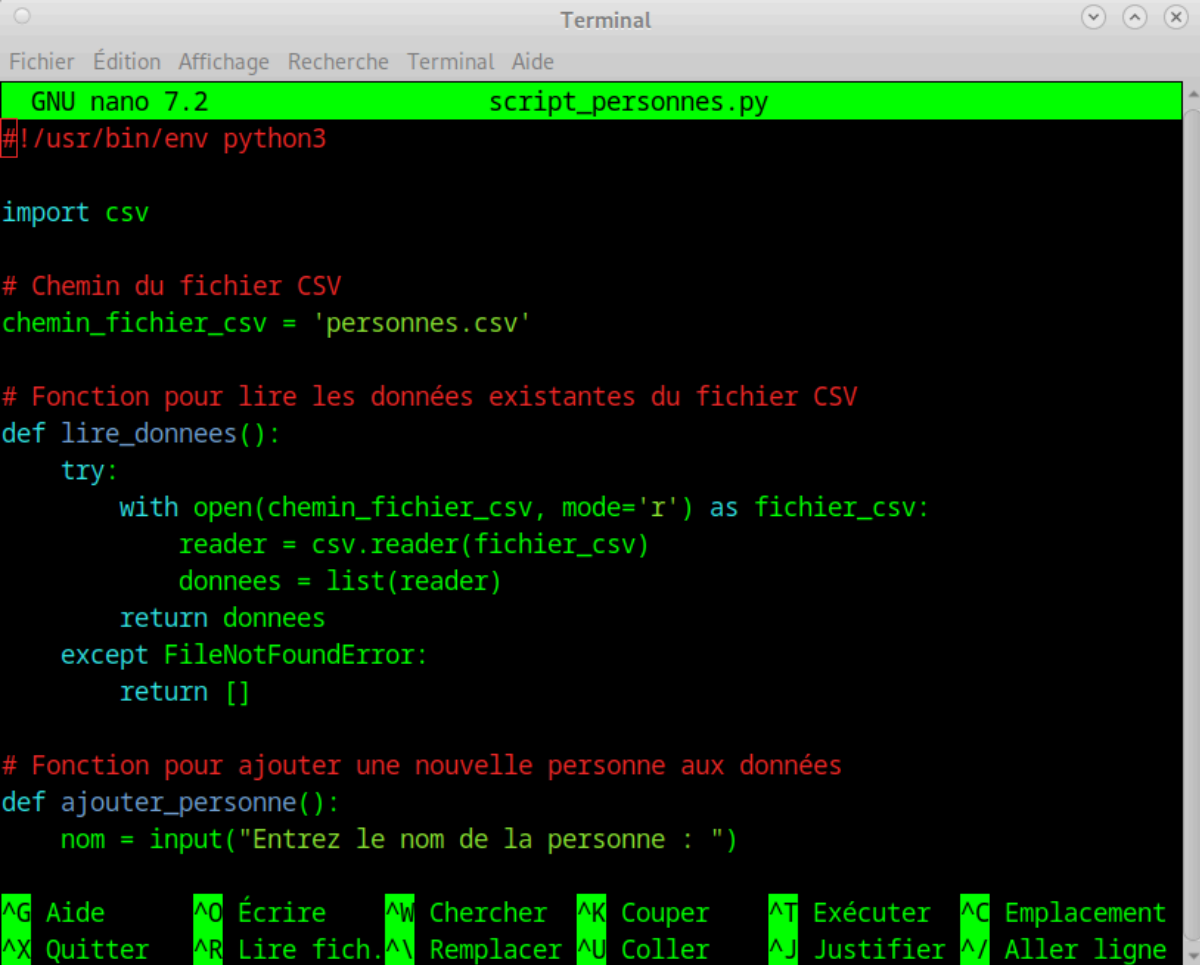
Extraction de l'archive dans le dossier Bureau

- **tar** : L'utilitaire de manipulation d'archives.
- **-x** : Indique à **tar** d'extraire les fichiers de l'archive.
- **-z** : Spécifie que l'archive est compressée avec gzip.
- **-v** : Affiche les fichiers pendant l'extraction.
- **-f** : Spécifie le nom du fichier d'archive.

```
La_Plateforme@debian:~$ tar -xzvf Plateforme.tar.gz -C ~/Bureau/  
Plateforme/  
Plateforme/mon_texte2.txt  
Plateforme/mon_texte.txt  
Plateforme/mon_texte1.txt  
Plateforme/mon_texte3.txt  
La_Plateforme@debian:~$ cd Bureau/  
La_Plateforme@debian:~/Bureau$ ls -l  
total 8  
-rw-r--r-- 1 La_Plateforme La_Plateforme  27 18 mars  09:59 mon_texte.txt  
drwxr-xr-x 2 La_Plateforme La_Plateforme 4096 18 mars  10:15 Plateforme  
La_Plateforme@debian:~/Bureau$ cd Plateforme/  
La_Plateforme@debian:~/Bureau/Plateforme$ ls -l  
total 16  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars  10:14 mon_texte1.txt  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars  10:15 mon_texte2.txt  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars  10:15 mon_texte3.txt  
-rw-r--r-- 1 La_Plateforme La_Plateforme 27 18 mars  10:07 mon_texte.txt  
La_Plateforme@debian:~/Bureau/Plateforme$
```

## 4.Manipulation de texte

- Ecriture du script



```
Terminal
Fichier Édition Affichage Recherche Terminal Aide
GNU nano 7.2 script_personnes.py
#!/usr/bin/env python3

import csv

# Chemin du fichier CSV
chemin_fichier_csv = 'personnes.csv'

# Fonction pour lire les données existantes du fichier CSV
def lire_donnees():
    try:
        with open(chemin_fichier_csv, mode='r') as fichier_csv:
            reader = csv.reader(fichier_csv)
            donnees = list(reader)
            return donnees
    except FileNotFoundError:
        return []

# Fonction pour ajouter une nouvelle personne aux données
def ajouter_personne():
    nom = input("Entrez le nom de la personne : ")

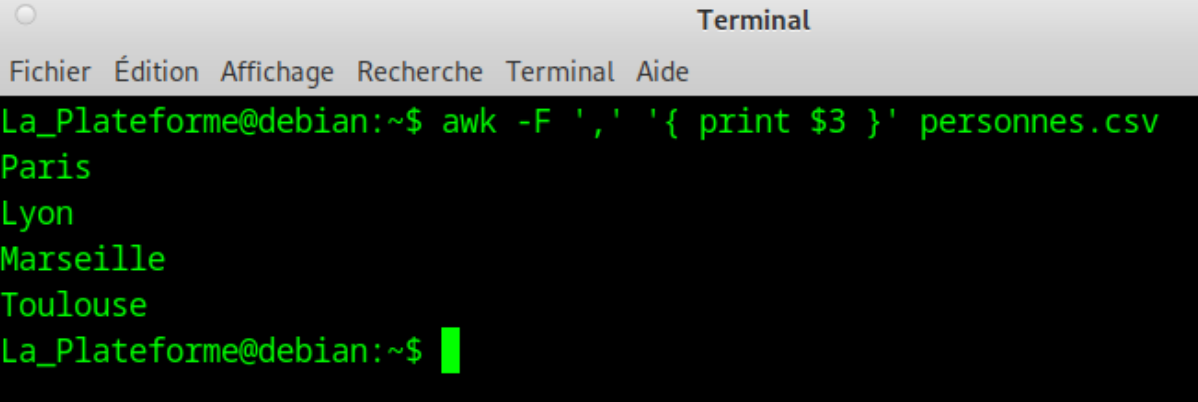
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^/ Aller ligne
```

- Test du script

```
Terminal
Fichier Édition Affichage Recherche Terminal Aide
Entrez le nom de la personne : Jean
Entrez l'âge de la personne : 25
Entrez la ville de la personne : Paris
La nouvelle personne a été ajoutée au fichier CSV personnes.csv avec succès !
La_Plateforme@debian:~$ ./script_personnes.py
Entrez le nom de la personne : Marie
Entrez l'âge de la personne : 30
Entrez la ville de la personne : Lyon
La nouvelle personne a été ajoutée au fichier CSV personnes.csv avec succès !
La_Plateforme@debian:~$ ./script_personnes.py
Entrez le nom de la personne : Pierre
Entrez l'âge de la personne : 22
Entrez la ville de la personne : Marseille
La nouvelle personne a été ajoutée au fichier CSV personnes.csv avec succès !
La_Plateforme@debian:~$ ./script_personnes.py
Entrez le nom de la personne : Sophie
Entrez l'âge de la personne : 35
Entrez la ville de la personne : Toulouse
La nouvelle personne a été ajoutée au fichier CSV personnes.csv avec succès !

La_Plateforme@debian:~$ cat personnes.csv
Nom,Age,Ville
Jean,25 ans,Paris
Marie,30 ans,Lyon
Pierre,22 ans,Marseille
Sophie,35 ans,Toulouse
La_Plateforme@debian:~$
```

- Utilisation de awk



```
Terminal
Fichier Édition Affichage Recherche Terminal Aide
La_Plateforme@debian:~$ awk -F ',' '{ print $3 }' personnes.csv
Paris
Lyon
Marseille
Toulouse
La_Plateforme@debian:~$
```

- `-F ','`: L'option `-F` spécifie le délimiteur de champ à utiliser. Dans cet exemple, le délimiteur de champ est défini comme une virgule (','), ce qui signifie que `awk` va diviser chaque ligne en champs en utilisant la virgule comme séparateur. Cela est utile pour traiter des fichiers CSV où les données sont généralement séparées par des virgules.
- `'{ print $3 }'`: C'est la partie principale de la commande `awk`. Cette instruction est écrite entre apostrophes (ou guillemets simples) pour la distinguer du reste de la commande. Elle spécifie l'action à exécuter sur chaque ligne du fichier.
  - `{ }`: Les accolades délimitent le bloc d'instructions à exécuter pour chaque ligne.
  - `print $3`: C'est l'instruction à exécuter. `print` est une commande `awk` qui affiche le contenu spécifié. `$3` est une variable spéciale en `awk` qui fait référence au troisième champ de la ligne. Dans le contexte de cette commande, cela signifie que nous imprimons le contenu du troisième champ de chaque ligne, qui contient les informations sur la ville dans le fichier CSV.



## 5. Gestion de processus

La commande `top` affiche en temps réel les informations sur les processus et l'utilisation des ressources système. Elle est très interactive et fournit des informations détaillées. Vous pouvez l'exécuter simplement en tapant `top` dans votre terminal.

Voilà ce que j'obtiens lorsque que je l'utilise.

```
top - 11:48:00 up 1:05, 1 user, load average: 0,71, 0,40, 0,30
Tâches: 175 total, 1 en cours, 174 en veille, 0 arrêté, 0 zombie
%Cpu(s): 4,8 ut, 1,9 sy, 0,0 ni, 93,0 id, 0,2 wa, 0,0 hi, 0,2 si, 0,0 st
MiB Mem : 1932,4 total, 413,8 libr, 1054,2 util, 635,2 tamp/cache
MiB Éch : 975,0 total, 975,0 libr, 0,0 util. 878,2 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
2113	La_Plat+	20	0	2815940	301112	150280	S	11,9	15,2	0:07.95	firefox+
1472	root	20	0	0	0	0	D	1,0	0,0	0:29.78	kworker+
2323	La_Plat+	20	0	11828	5584	3420	R	1,0	0,3	0:00.16	top
2231	La_Plat+	20	0	2427448	97132	77080	S	0,7	4,9	0:00.87	WebExte+
178	root	20	0	0	0	0	I	0,3	0,0	0:15.13	kworker+
415	root	20	0	167720	11492	7988	S	0,3	0,6	0:15.86	vmtoolsd
1183	La_Plat+	20	0	267692	24676	17104	S	0,3	1,2	0:00.45	mate-sc+
1206	La_Plat+	20	0	141660	40624	30812	S	0,3	2,1	0:14.19	vmtoolsd
1767	root	20	0	0	0	0	I	0,3	0,0	0:00.13	kworker+
2112	root	20	0	0	0	0	I	0,3	0,0	0:00.04	kworker+
2179	La_Plat+	20	0	2440684	119064	92784	S	0,3	6,0	0:01.09	Privile+
2273	La_Plat+	20	0	2398616	70028	58736	S	0,3	3,5	0:00.19	Web Con+
1	root	20	0	102180	12272	9224	S	0,0	0,6	0:02.23	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.03	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+
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	slub_fl+

1. **Identifier le PID du processus à arrêter** : Vous pouvez utiliser l'une des commandes mentionnées précédemment, comme `ps`, `top`, `htop`, `pgrep`, etc., pour trouver le PID du processus que vous souhaitez arrêter.
- 2.
3. **Utiliser la commande `kill`** : Une fois que vous avez identifié le PID du processus, vous pouvez utiliser la commande `kill` suivie du PID pour arrêter le processus.

Dans mon cas j'ai démarré une page internet avec firefox. Je vais donc taper "kill 2113".

```
5 root      0 -20      0      0      0 I    0,0    0,0    0:00.00 slub_fl+
6 root      0 -20      0      0      0 I    0,0    0,0    0:00.00 netns
10 root     0 -20      0      0      0 I    0,0    0,0    0:00.00 mm_perc+
11 root     20  0      0      0      0 I    0,0    0,0    0:00.00 rcu_tas+
La_Plateforme@debian:~$ kill 2113
La_Plateforme@debian:~$
```

Et je constate la disparition de ma page internet dans la barre des tâches.

Par défaut, `kill` envoie le signal SIGTERM, demandant au processus de se terminer proprement. Si le processus ne répond pas ou si vous voulez forcer l'arrêt, vous pouvez utiliser le signal SIGKILL en spécifiant `-9` avec la commande `kill`. Par exemple :

bash

```
kill -9 1234
```

Cela forcera l'arrêt immédiat du processus.

Commande supplémentaire:

```
La_Plateforme@debian:~$ pgrep firefox
3706
```


```
La_Plateforme@debian:~$ killall firefox-esr
```

## 6.Surveillance des ressources systèmes

Pour commencer j'installe Sysstat sur ma vm :

```
La_Plateforme@debian:~$ sudo apt install sysstat
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  isag
Les NOUVEAUX paquets suivants seront installés :
  sysstat
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 596 ko dans les archives.
Après cette opération, 1 828 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 sysstat amd64 12.6.1-1 [596 kB]
596 ko réceptionnés en 0s (3 636 ko/s)
Préconfiguration des paquets...
Sélection du paquet sysstat précédemment désélectionné.
(Lecture de la base de données... 165898 fichiers et répertoires déjà installés.
)
Préparation du dépaquetage de .../sysstat_12.6.1-1_amd64.deb ...
Dépaquetage de sysstat (12.6.1-1) ...
```

J'ai ensuite écrit un script qui permet de récupérer les informations de mon systemes et de les afficher dans le terminal :



```
GNU nano 7.2 system_info.sh
#!/bin/bash

# Récupération des statistiques d'utilisation du CPU
echo "Utilisation du CPU:"
mpstat -P ALL | grep -v "CPU"

# Récupération des statistiques d'utilisation de la mémoire
echo -e "\nUtilisation de la mémoire:"
free -h

# Récupération des statistiques d'utilisation du disque dur
echo -e "\nUtilisation du Disque Dur"
df -h |grep /sda1

# Récupération du nombre de processus en cours
echo -e "\nNombre de processus en cours:"
ps -e | wc -l

[ Lecture de 18 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^/ Aller ligne
```

Voilà ce que j'obtiens lorsque j'exécute le script:

```
Terminal
Fichier  Édition  Affichage  Recherche  Terminal  Aide

Nombre de processus en cours:
172
La_Plateforme@debian:~$ nano system_info.sh
La_Plateforme@debian:~$ ./system_info.sh
Utilisation du CPU:

13:38:34    all    1,95    0,01    0,82    0,78    0,00    0,04    0,00    0,00    0,00    96
,40
13:38:34      0    1,96    0,01    0,88    0,56    0,00    0,03    0,00    0,00    0,00    96
,57
13:38:34      1    1,95    0,00    0,77    1,00    0,00    0,05    0,00    0,00    0,00    96
,23

Utilisation de la mémoire:

          total      utilisé      libre      partagé  tamp/cache  disponible
Mem:      1,9Gi      816Mi      308Mi      23Mi      989Mi      1,1Gi
Échange:   974Mi      780Ki      974Mi

Utilisation du Disque Dur
/dev/sda1    48G    4,8G    41G    11% /

Nombre de processus en cours:
171
La_Plateforme@debian:~$
```

Par la suite, il m'a fallu manipuler les sorties obtenues via les différentes commandes, afin de les enregistrer dans un fichier csv. Pour réaliser cette manipulation j'ai enregistré mes sorties dans des variables.

```
# Récupération des statistiques d'utilisation du CPU
echo "Utilisation du CPU:"
output_cpu=$(mpstat -P ALL | grep -v "CPU")
echo "$output_cpu"

# Récupération des statistiques d'utilisation de la mémoire
echo -e "\nUtilisation de la mémoire:"
output_mem=$(free -h)
echo "$output_mem"

# Récupération des statistiques d'utilisation du disque dur
echo -e "\nUtilisation du Disque Dur"
output_dd=$(df -h | grep /sda1)
echo "$output_dd"
```

Puis j'ai les ai manipulé avec la commande " awk "

Dans un premier temps je n'ai pas utilisé la commande sed, ce qui me posait un problème par la suite pour l'écriture du fichier csv qui utilise les virgules comme séparateur.

C'est pourquoi dans le screen suivant j'utilise sed pour remplacer les , par des . et de ce fait garder mes valeurs décimales au bon format pour l'enregistrement CSV.

```
# Récupération de l'utilisation du CPU pour chaque cœur
cpu_all_dispo=$(echo "$output_cpu" | awk 'NR==2 {print $NF}' | sed 's/,././g')
cpu_1_dispo=$(echo "$output_cpu" | awk 'NR==3 {print $NF}' | sed 's/,././g')
cpu_2_dispo=$(echo "$output_cpu" | awk 'NR==4 {print $NF}' | sed 's/,././g')

# Récupération de la mémoire disponible
ram_dispo=$(echo "$output_mem" | grep "Mem" | awk '{print $7}' | sed 's/,././g')

# Récupération de l'espace disque utilisé
dd_use=$(echo "$output_dd" | awk '{print $5}')
```

Une fois les données récupérées et formatées à ma guise je n'ai plus qu'à les écrire dans un fichier csv.

```
# Création du fichier CSV
echo "Dispo Cpu(all),Dispo Cpu(1),Dispo Cpu(2),Ram dispo,Espace Disque Utilisé" > stats.csv
echo "$cpu_all_dispo,$cpu_1_dispo,$cpu_2_dispo,$ram_dispo,$dd_use" >> stats.csv

# Affichage des valeurs ecrites dans le fichier csv
echo -e "\nValeurs enregistrées dans stats.csv :"
cat stats.csv
```

Voila le script en totalité

```
#!/bin/bash

# ---- 1er Partie : Recupération des infos du systemes ---- #

# Récupération des statistiques d'utilisation du CPU
echo "Utilisation du CPU:"
output_cpu=$(mpstat -P ALL | grep -v "CPU")
echo "$output_cpu"

# Récupération des statistiques d'utilisation de la mémoire
echo -e "\nUtilisation de la mémoire:"
output_mem=$(free -h)
echo "$output_mem"

# Récupération des statistiques d'utilisation du disque dur
echo -e "\nUtilisation du Disque Dur"
output_dd=$(df -h | grep /sda1)
echo "$output_dd"

# Récupération du nombre de processus en cours
echo -e "\nNombre de processus en cours:"
ps -e | wc -l

# ---- Partie 2 : Formatage et création du fichier CSV ---- #

# Récupération de l'utilisation du CPU pour chaque cœur
cpu_all_dispo=$(echo "$output_cpu" | awk 'NR==2 {print $NF}' | sed 's/,./g')
cpu_1_dispo=$(echo "$output_cpu" | awk 'NR==3 {print $NF}' | sed 's/,./g')
cpu_2_dispo=$(echo "$output_cpu" | awk 'NR==4 {print $NF}' | sed 's/,./g')

# Récupération de la mémoire disponible
ram_dispo=$(echo "$output_mem" | grep "Mem" | awk '{print $7}' | sed 's/,./g')

# Récupération de l'espace disque utilisé
dd_use=$(echo "$output_dd" | awk '{print $5}')

# Affichage des valeurs récupérées
echo -e "\nValeurs récupérées :"
echo "Dispo du CPU (tous les cœurs) : $cpu_all_dispo"
echo "Dispo du CPU (cœur 1) : $cpu_1_dispo"
echo "Dispo du CPU (cœur 2) : $cpu_2_dispo"
echo "RAM disponible : $ram_dispo"
echo "Espace disque utilisé : $dd_use"

# Création du fichier CSV
echo "Dispo Cpu(all),Dispo Cpu(1),Dispo Cpu(2),Ram dispo,Espace Disque Utilisé" > stats.csv
echo "$cpu_all_dispo,$cpu_1_dispo,$cpu_2_dispo,$ram_dispo,$dd_use" >> stats.csv

# Affichage des valeurs ecrites dans le fichier csv
echo -e "\nValeurs enregistrées dans stats.csv :"
cat stats.csv
```

Voila le rendu final de mon script lorsque que je l'exécute

```
La_Plateforme@debian:~$ ./system_info.sh
Utilisation du CPU:

11:06:50    all    0,20    0,00    0,19    0,04    0,00    0,02    0,00    0,00    0,00    99,54
11:06:50      0    0,19    0,00    0,21    0,02    0,00    0,02    0,00    0,00    0,00    99,57
11:06:50      1    0,22    0,00    0,18    0,06    0,00    0,03    0,00    0,00    0,00    99,51

Utilisation de la mémoire:
                total      utilisé      libre      partagé tamp/cache  disponible
Mem:           1,9Gi       809Mi       781Mi       26Mi          513Mi        1,1Gi
Échange:        974Mi         0B       974Mi

Utilisation du Disque Dur
/dev/sda1        48G    4,8G    41G  11% /

Nombre de processus en cours:
170

Valeurs récupérées :
Dispo du CPU (tous les cœurs) : 99.54
Dispo du CPU (coeur 1) : 99.57
Dispo du CPU (coeur 2) : 99.51
RAM disponible : 1.1Gi
Espace disque utilisé : 11%

Valeurs enregistrées dans stats.csv :
Dispo Cpu(all),Dispo Cpu(1),Dispo Cpu(2),Ram dispo,Espace Disque Utilisé
99.54,99.57,99.51,1.1Gi,11%
La_Plateforme@debian:~$
```



## 8.Scripting Avancé :

J'écris le script qui permet d'automatiser la sauvegarde

```
#!/bin/bash

# Définition des variables
current_date=$(date +%d-%m-%Y-%H:%M)

# Vérifie si le fichier Plateforme_backup.tar.gz existe
if [ -e "Plateforme_backup.tar.gz" ]; then
    # Décompresse l'archive si elle existe
    tar -xzf Plateforme_backup.tar.gz
else
    # Crée le répertoire Plateforme_backup s'il n'existe pas
    mkdir -p Plateforme_backup
fi

# Copie le contenu du répertoire Plateforme dans Plateforme_backup avec la date actuelle ajoutée au nom
cp -r Plateforme/ Plateforme_backup/Plateforme_copie_$current_date

tar -czf Plateforme_backup.tar.gz Plateforme_backup
rm -r Plateforme_backup
```

Voilà une autre version du script qui permet de ne pas avoir trop de fichier à compresser, puisque lors de la première version l'archivage se faisait sur tous les précédents backup

```
#!/bin/bash

# Définition des variables
current_date=$(date +%d-%m-%Y-%H:%M)

# Vérifie si le dossier Plateforme_backup existe
if [ ! -d "Plateforme_backup" ]; then
    # Crée le dossier Plateforme_backup s'il n'existe pas
    mkdir -p Plateforme_backup
    echo "Le dossier Plateforme_backup a été créé."
else
    echo "Le dossier Plateforme_backup existe."
fi

# Crée une archive du dossier Plateforme avec la date actuelle ajoutée au nom
tar -czf Plateforme_backup/Plateforme_copie_$current_date.tar.gz Plateforme && echo "L'archive de Plateforme a été créée et déplacée vers Plateforme_backup."
```

Puis j'ajoute l'exécution de mon script a crontab

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
*/2 * * * * /home/La_Plateforme/script_avance.sh
```

## 9. Automatisation des mises à jour logicielles

Voilà le script pour répondre à l'exercice

La variable spéciale `$?` est une variable intégrée dans les shells Unix et Linux, tels que bash, qui contient le code de retour de la dernière commande exécutée, dans mon cas apt update.

```
#!/bin/bash

# Exécute apt update et liste les paquets qui peuvent être mis à jour !
sudo apt update && apt list -- upgradable

# Vérifie le code de retour de apt update
if [ $? -eq 0 ]; then
    # Si la mise à jour est réussie, exécute apt upgrade
    sudo apt upgrade

    # Vérifie le code de retour de apt upgrade
    if [ $? -eq 0 ]; then
        echo "Mise à jour effectuée."
    else
        echo "Aucune mise à jour disponible."
    fi
fi
```

Et voilà la version spéciale TytySensei

```
#!/bin/bash

# Met à jour la liste des paquets et récupère le nombre de paquets pouvant être mis à jour
packages=$(sudo apt update 2>/dev/null | grep 'à jour' | cut -d ' ' -f 1)
# 2>/dev/null masque la sortie console de apt update
# grep packages permet de filtrer les lignes contenant le terme packages
# cut -d ' ' ajoute un séparateur en fonction des espaces
# -f 1 récupère uniquement le premier mot de la ligne

# Si la variable $packages est égale à "Tous", affiche "Tous les paquets sont à jour"
if [ "$packages" = "Tous" ]; then
    echo "Tous les paquets sont à jour"
    exit 0
else
    # Convertit la chaîne en un nombre entier pour effectuer la comparaison
    packages=$(echo "$packages" | tr -d '[:space:]')
    # Affiche le nombre de paquets à mettre à jour
    echo "$packages paquet(s) à mettre à jour"
fi

# Vérifie s'il y a des paquets à mettre à jour
if [ "$packages" -ge 1 ]; then
    # Si des paquets doivent être mis à jour, exécute apt upgrade et affiche le nombre de paquets mis à jour
    sudo apt upgrade -y && echo "$packages paquet(s) mis à jour"
fi
```