

Systeme, scripts et sécurité

Introduction du projet

Pour ce projet, il est attendu de faire une documentation détaillée tout au long du projet regroupant les différentes étapes appliquées dans chaque exercice, les lignes de commande utilisées avec une explication de leurs fonctionnements et les différentes recherches effectuées tout au long de cette semaine.

Création d'une VM Debian

Tout d'abord, j'ai téléchargé Debian 12 AMD64. Ensuite je l'ai installé sur Oracle VM Virtualbox manager qui est d'ailleurs mon Hyperviseur préféré. Je me suis assuré que ma VM soit connectée à Internet lors de sa création.

Commandes de recherche avancée

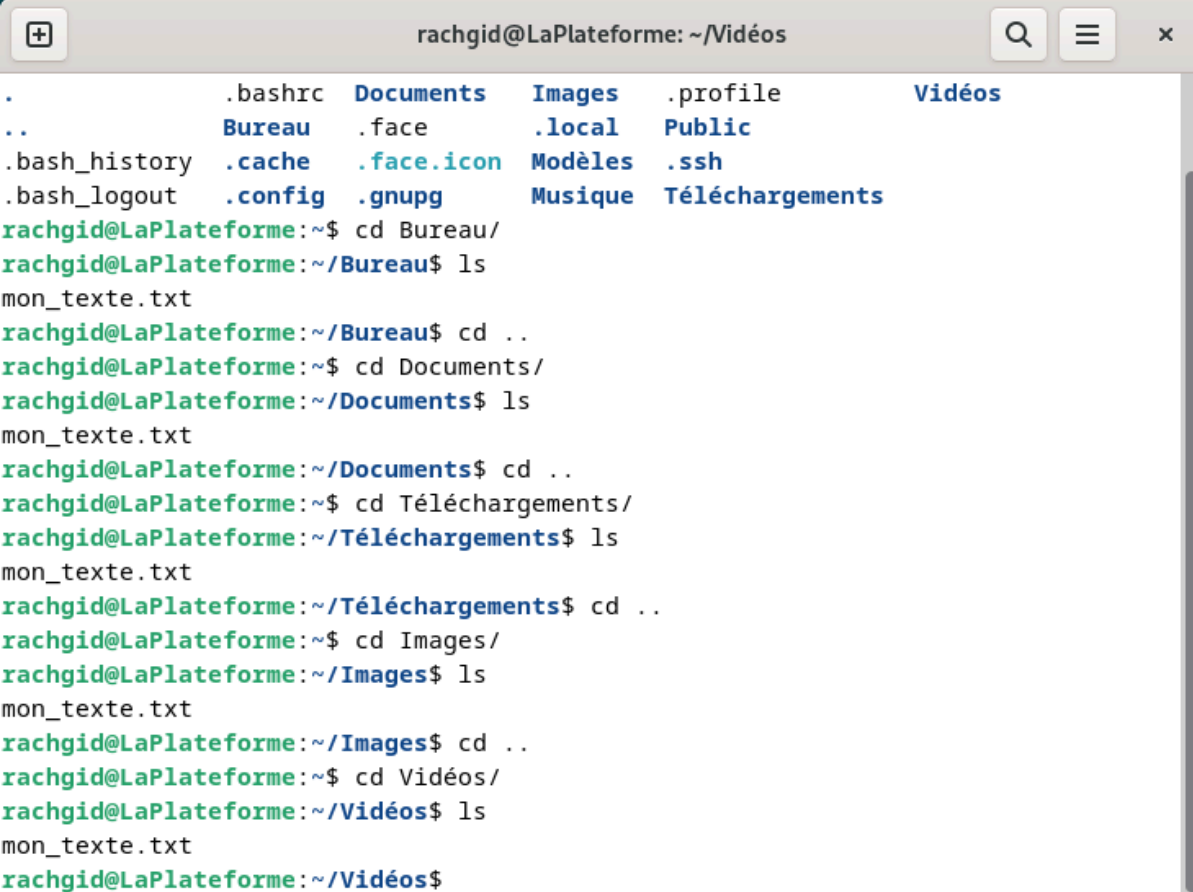
Les commandes à utiliser pour cet exercice sont :

Pour créer un fichier dans un répertoire donné, il faut exécuter la commande :

cd : Cette commande + le nom du dossier (s'il existe) permet de parcourir ce répertoire ou ce dossier. Dans le cas précis de cet exercice, pour parcourir les dossiers Bureau ou Téléchargement..., il faut taper :
cd Bureau/ ou **cd Téléchargement/**

nano mon_texte.txt : Cette commande permet de créer un fichier nommé “mon_texte.txt” et de pouvoir l’éditer pour écrire un texte.

cd .. : Cette commande permet de revenir au répertoire précédent.

A terminal window titled 'rachgid@LaPlateforme: ~/Vidéos' with search, menu, and close buttons. It displays a series of commands and their outputs, showing navigation through various directories. The terminal output is as follows:

```
.          .bashrc Documents Images .profile  Vidéos
..         Bureau  .face    .local  Public
.bash_history .cache  .face.icon Modèles .ssh
.bash_logout .config .gnupg   Musique Téléchargements
rachgid@LaPlateforme:~$ cd Bureau/
rachgid@LaPlateforme:~/Bureau$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Bureau$ cd ..
rachgid@LaPlateforme:~$ cd Documents/
rachgid@LaPlateforme:~/Documents$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Documents$ cd ..
rachgid@LaPlateforme:~$ cd Téléchargements/
rachgid@LaPlateforme:~/Téléchargements$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Téléchargements$ cd ..
rachgid@LaPlateforme:~$ cd Images/
rachgid@LaPlateforme:~/Images$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Images$ cd ..
rachgid@LaPlateforme:~$ cd Vidéos/
rachgid@LaPlateforme:~/Vidéos$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Vidéos$
```



```

rachgid@LaPlateforme: ~/Documents/Plateforme
GNU nano 7.2 mon_texte.txt
que la force soit avec toi.

[ Lecture de 1 ligne ]
^G Aide    ^O Écrire  ^W Chercher ^K Couper  ^T Exécuter ^C Emplacement
^X Quitter ^R Lire fich. ^\ Remplacer ^U Coller  ^J Justifier ^_ Aller ligne
```

Pour localiser les cinq fichiers “mon_texte.txt” à partir du répertoire de ma session en utilisez le terminal et le mot “force, j’exécute la commande :

grep -rl “force” ~/

Voici le résultat :

```
rachgid@LaPlateforme: ~  
rachgid@LaPlateforme:~$ grep -rl "force" ~/  
/home/rachgid/.cache/tracker3/files/meta.db  
/home/rachgid/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontolog  
y%2Fv3%2Ftracker%23Documents.db  
/home/rachgid/.cache/gnome-software/appstream/components.xmlb  
/home/rachgid/.cache/gnome-software/odrs/ratings.json  
/home/rachgid/Images/mon_texte.txt  
/home/rachgid/.bashrc  
/home/rachgid/Vidéos/mon_texte.txt  
/home/rachgid/Bureau/mon_texte.txt  
/home/rachgid/Téléchargements/mon_texte.txt  
/home/rachgid/Documents/mon_texte.txt  
/home/rachgid/.bash_history  
rachgid@LaPlateforme:~$ grep -rl "force" ~/
```

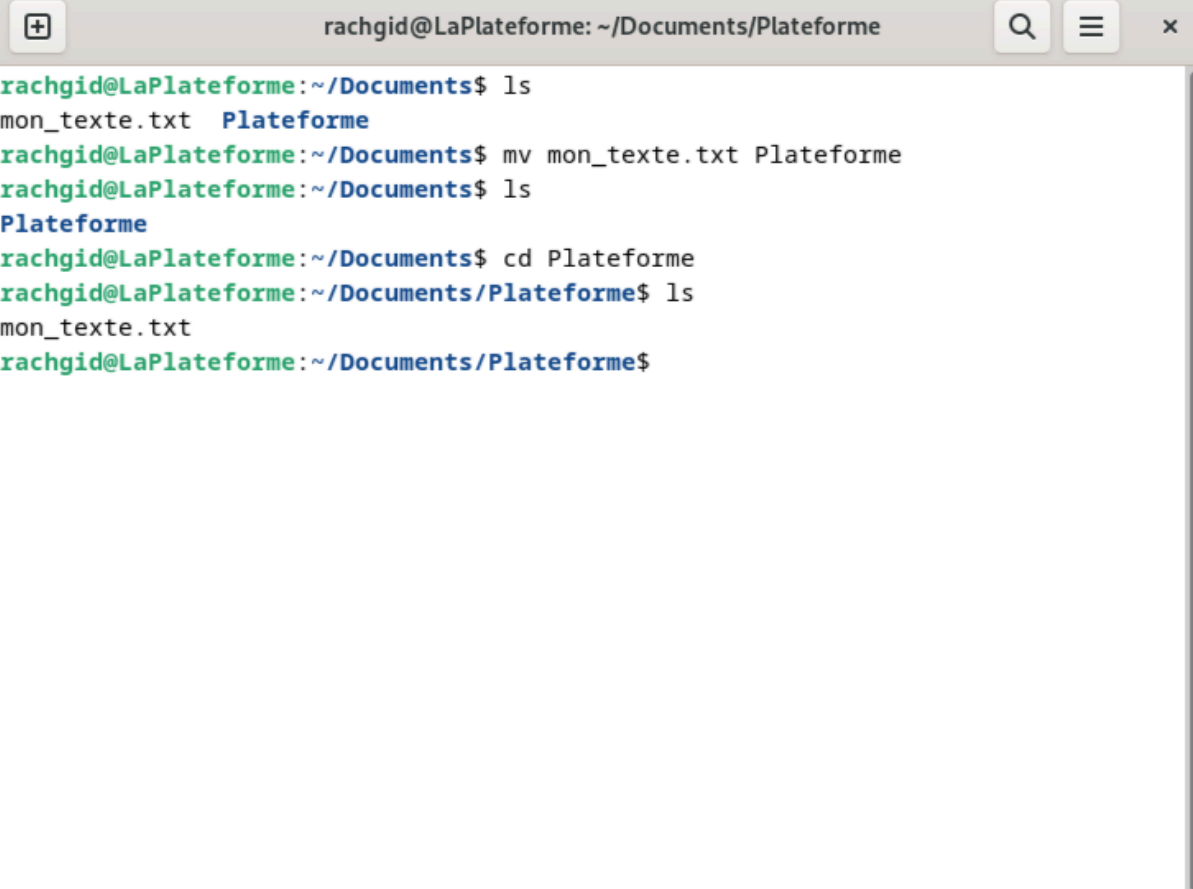
Compression et Décompression de fichiers

Pour créer un répertoire nommé Plateforme, il faut exécuter la commande suivante :

mkdir Plateforme

Pour ajouter le fichier `mon_texte.txt` au répertoire `Plateforme` qu'on vient de créer, on exécute la commande suivante :

`mv mon_texte.txt Plateforme`



```
rachgid@LaPlateforme: ~/Documents/Plateforme
rachgid@LaPlateforme:~/Documents$ ls
mon_texte.txt  Plateforme
rachgid@LaPlateforme:~/Documents$ mv mon_texte.txt Plateforme
rachgid@LaPlateforme:~/Documents$ ls
Plateforme
rachgid@LaPlateforme:~/Documents$ cd Plateforme
rachgid@LaPlateforme:~/Documents/Plateforme$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Documents/Plateforme$
```

Pour dupliquer ce fichier quatre fois dans le même répertoire, formant ainsi un total de cinq fichiers dans le répertoire "`Plateforme`", il faut exécuter la commande suivante :

`cp` (Qui est souvent utilisé pour copier). Par conséquent les lignes de commandes à exécuter sont :

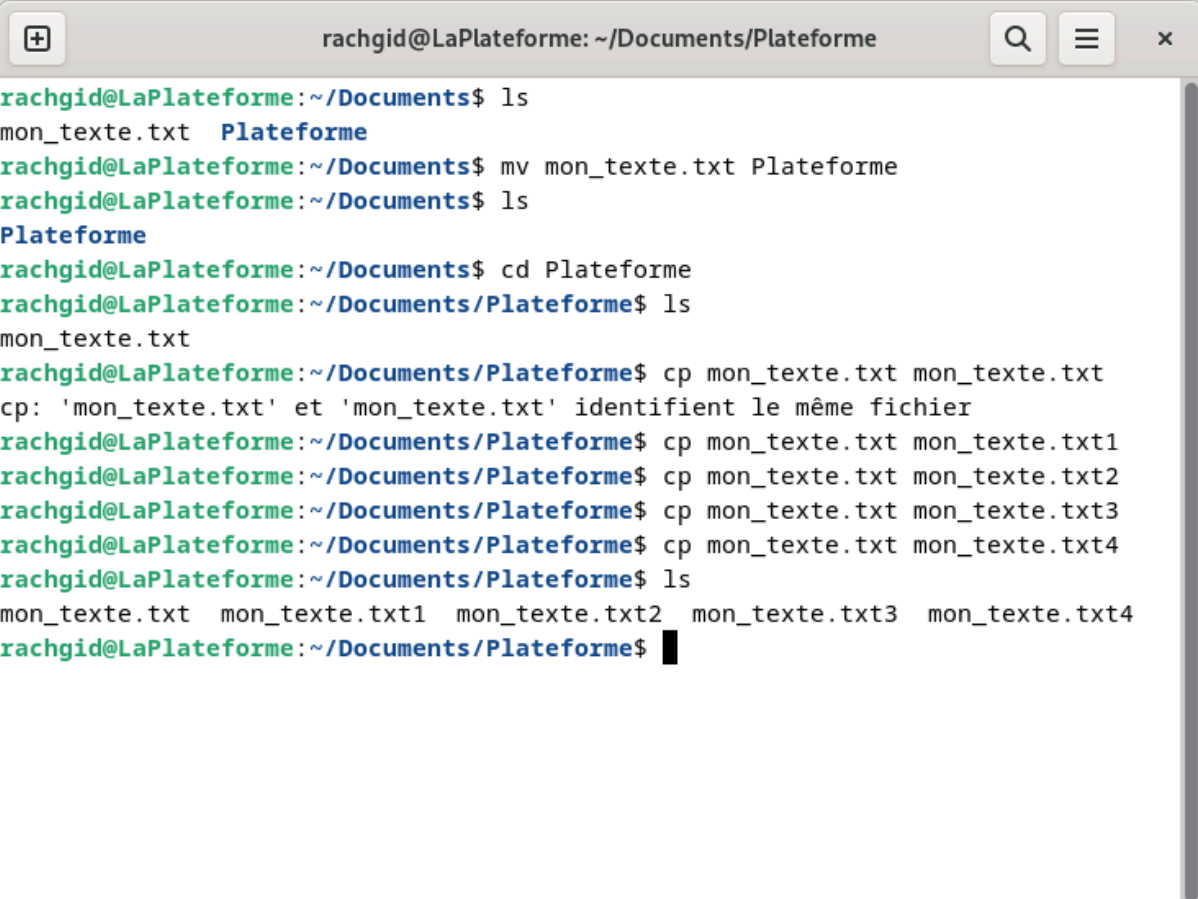
`cp mon_texte.txt mon_texte.txt1`

`cp mon_texte.txt mon_texte.txt2`

`cp mon_texte.txt mon_texte.txt3`

`cp mon_texte.txt mon_texte.txt4`

Ces commandes permettent de dupliquer 4 fois intégralement le même fichier, seulement que leurs noms doivent être différents parce que c'est impossible de dupliquer un fichier en plusieurs portant le même nom dans un même répertoire.



```

rachgid@LaPlateforme: ~/Documents/Plateforme
rachgid@LaPlateforme:~/Documents$ ls
mon_texte.txt  Plateforme
rachgid@LaPlateforme:~/Documents$ mv mon_texte.txt Plateforme
rachgid@LaPlateforme:~/Documents$ ls
Plateforme
rachgid@LaPlateforme:~/Documents$ cd Plateforme
rachgid@LaPlateforme:~/Documents/Plateforme$ ls
mon_texte.txt
rachgid@LaPlateforme:~/Documents/Plateforme$ cp mon_texte.txt mon_texte.txt
cp: 'mon_texte.txt' et 'mon_texte.txt' identifient le même fichier
rachgid@LaPlateforme:~/Documents/Plateforme$ cp mon_texte.txt mon_texte.txt1
rachgid@LaPlateforme:~/Documents/Plateforme$ cp mon_texte.txt mon_texte.txt2
rachgid@LaPlateforme:~/Documents/Plateforme$ cp mon_texte.txt mon_texte.txt3
rachgid@LaPlateforme:~/Documents/Plateforme$ cp mon_texte.txt mon_texte.txt4
rachgid@LaPlateforme:~/Documents/Plateforme$ ls
mon_texte.txt  mon_texte.txt1  mon_texte.txt2  mon_texte.txt3  mon_texte.txt4
rachgid@LaPlateforme:~/Documents/Plateforme$
```

Pour archiver le répertoire "Plateforme" en utilisant les commandes "tar" et "gzip":

tar -czvf Pateforme.tar.gz chemin_vers_le_repertoire (Archivage avec l'option de compression)

tar : l'Outil pour archiver

c : Créer une nouvelle archive

v : Afficher les fichiers pendant qu'ils sont ajoutés à l'archive

(optionnelle, mais utile pour suivre le progrès)

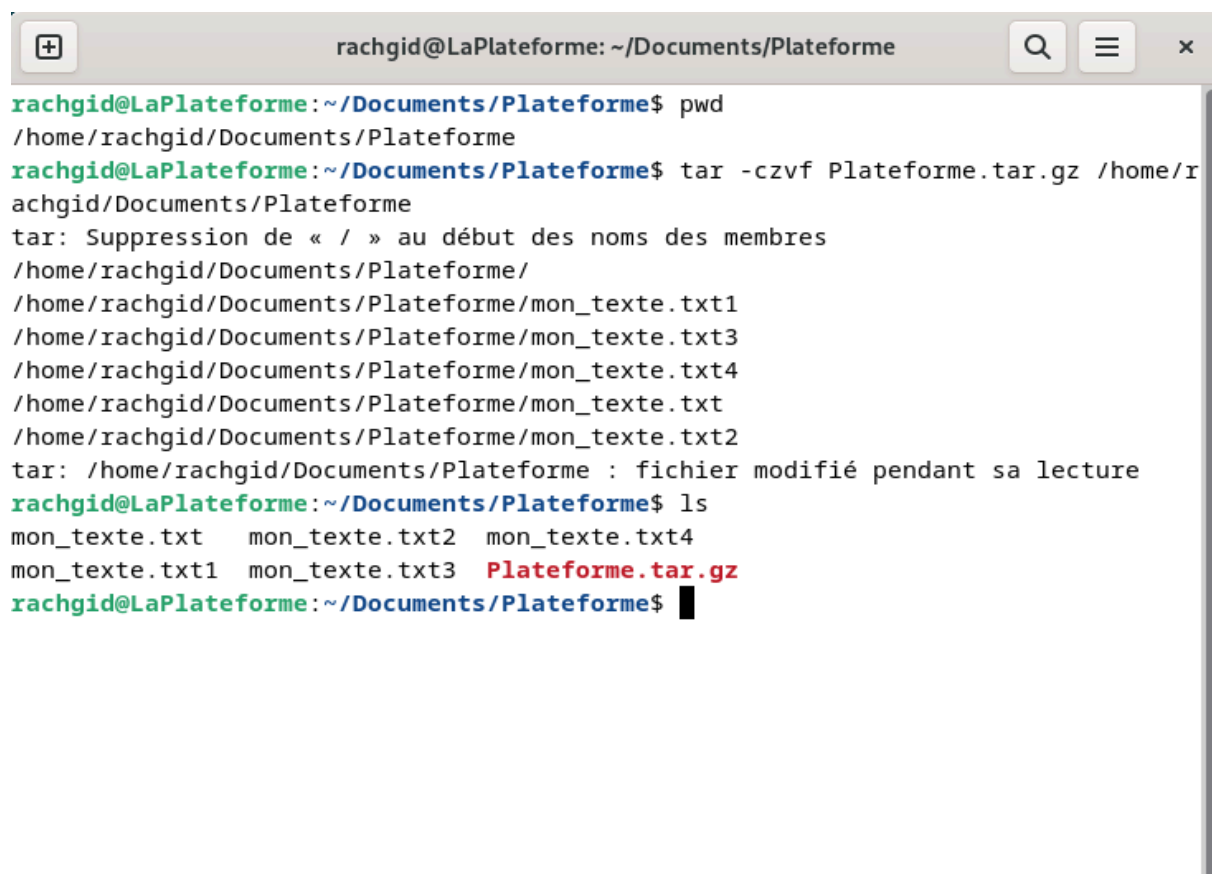
z : Utiliser gzip pour compresser l'archive

f : Spécifier le nom de l'archive

Plateforme.tar.gz : C'est juste le nom que je souhaite donner à mon archive compressée avec l'extension `.tar`

chemin_vers_le_repertoire : Spécification du chemin du répertoire selon chaque machine.

Le résultat final est dans la capture d'écran suivante :

A terminal window titled 'rachgid@LaPlateforme: ~/Documents/Plateforme' showing the execution of tar commands. The user runs 'pwd' to confirm the directory, then 'tar -czvf Plateforme.tar.gz /home/rachgid/Documents/Plateforme'. The terminal output shows the files being archived, including a warning about slashes in filenames. Finally, the user runs 'ls' to show the directory contents, where 'Plateforme.tar.gz' is highlighted in red.

```
rachgid@LaPlateforme: ~/Documents/Plateforme
rachgid@LaPlateforme:~/Documents/Plateforme$ pwd
/home/rachgid/Documents/Plateforme
rachgid@LaPlateforme:~/Documents/Plateforme$ tar -czvf Plateforme.tar.gz /home/rachgid/Documents/Plateforme
tar: Suppression de « / » au début des noms des membres
/home/rachgid/Documents/Plateforme/
/home/rachgid/Documents/Plateforme/mon_texte.txt1
/home/rachgid/Documents/Plateforme/mon_texte.txt3
/home/rachgid/Documents/Plateforme/mon_texte.txt4
/home/rachgid/Documents/Plateforme/mon_texte.txt
/home/rachgid/Documents/Plateforme/mon_texte.txt2
tar: /home/rachgid/Documents/Plateforme : fichier modifié pendant sa lecture
rachgid@LaPlateforme:~/Documents/Plateforme$ ls
mon_texte.txt  mon_texte.txt2  mon_texte.txt4
mon_texte.txt1  mon_texte.txt3  Plateforme.tar.gz
rachgid@LaPlateforme:~/Documents/Plateforme$
```

tar -cvf Plateforme.tar chemin_vers_le_repertoire (Archivage sans l'option de décompression)

Décompression des archives créées en utilisant les commandes appropriées.

Explorons diverses options de décompression.

Les commandes sont les suivantes :

tar -xzvf Plateforme.tar.gz

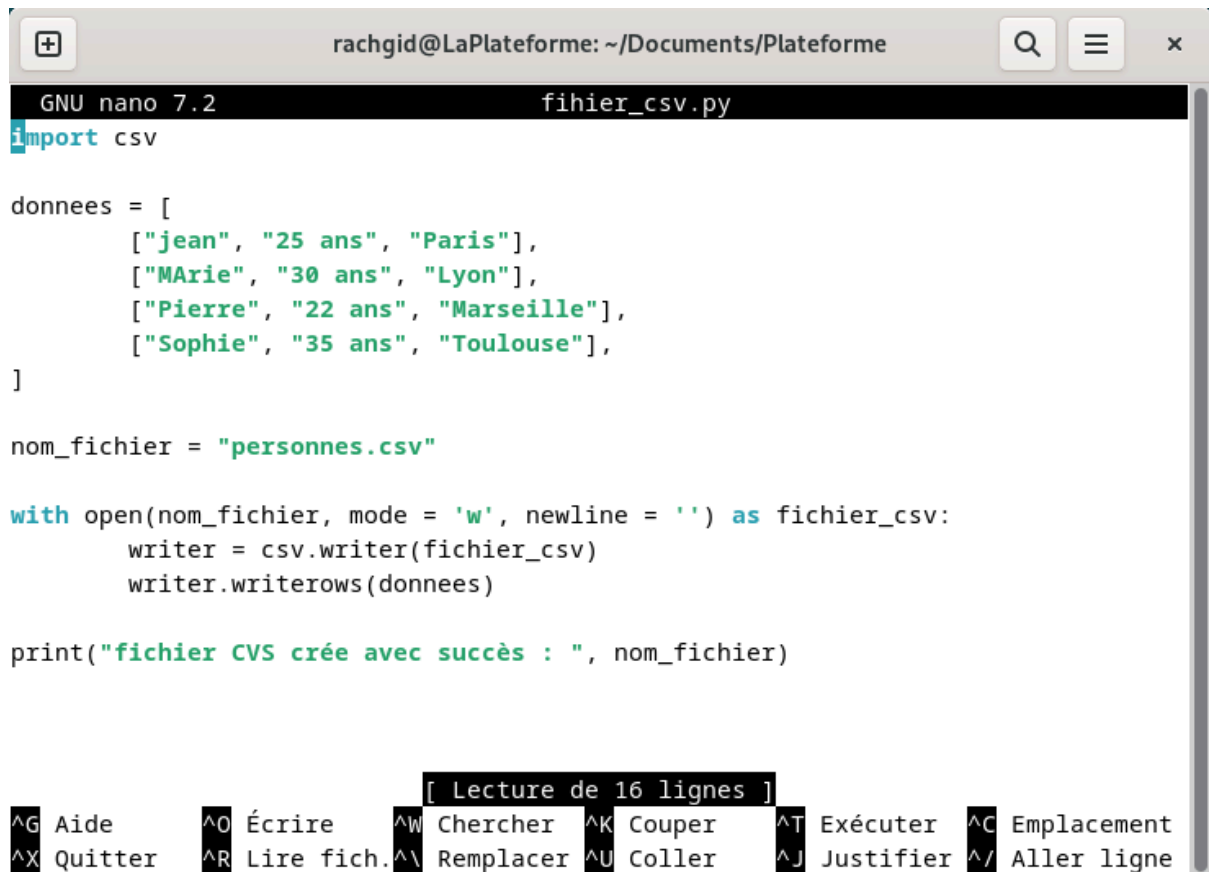
tar -xvf Plateforme.tar

```
rachgid@LaPlateforme:~/Documents/Plateforme$ tar -xzvf Plateforme.tar.gz
home/rachgid/Documents/Plateforme/
home/rachgid/Documents/Plateforme/mon_texte.txt1
home/rachgid/Documents/Plateforme/mon_texte.txt3
home/rachgid/Documents/Plateforme/mon_texte.txt4
home/rachgid/Documents/Plateforme/mon_texte.txt
home/rachgid/Documents/Plateforme/mon_texte.txt2
rachgid@LaPlateforme:~/Documents/Plateforme$ ls
home          mon_texte.txt1  mon_texte.txt3  Plateforme.tar.gz
mon_texte.txt mon_texte.txt2  mon_texte.txt4
rachgid@LaPlateforme:~/Documents/Plateforme$
```

```
rachgid@LaPlateforme:~/Documents/Plateforme$ ls
home          mon_texte.txt1  mon_texte.txt3  Plateforme.tar
mon_texte.txt mon_texte.txt2  mon_texte.txt4  Plateforme.tar.gz
rachgid@LaPlateforme:~/Documents/Plateforme$
```

Manipulation de texte

Le script python pour créer le fichier csv:



The screenshot shows a terminal window with the title bar 'rachgid@LaPlateforme: ~/Documents/Plateforme'. The editor is GNU nano 7.2, editing a file named 'fichier_csv.py'. The code in the file is as follows:

```
import csv

donnees = [
    ["jean", "25 ans", "Paris"],
    ["MArie", "30 ans", "Lyon"],
    ["Pierre", "22 ans", "Marseille"],
    ["Sophie", "35 ans", "Toulouse"],
]

nom_fichier = "personnes.csv"

with open(nom_fichier, mode = 'w', newline = '') as fichier_csv:
    writer = csv.writer(fichier_csv)
    writer.writerows(donnees)

print("fichier CVS crée avec succès : ", nom_fichier)
```

At the bottom of the window, there is a status bar with the text '[Lecture de 16 lignes]' and a set of keyboard shortcuts:

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

Extraction des informations relatives aux villes de chaque personne en utilisant la commande “awk”. La commande à exécuter pour obtenir les informations en question est la suivante :

awk -F "," '{print \$3}' personnes.csv

- **awk** : est un outil utilisé pour extraire et manipuler des données dans les fichiers texte.
- **-F ","** : spécifie que les champs dans le fichier sont séparés par des virgules.
- **'{print \$3}'** : c'est le code awk lui-même, qui demande à awk d'imprimer le troisième champ de chaque ligne. D'où l'extraction des informations relatives aux villes de chaque personne.

```
rachgid@LaPlateforme:~/Documents/Plateforme$ awk -F"," '{print $3}' personnes.csv
Paris
Lyon
Marseille
Toulouse
rachgid@LaPlateforme:~/Documents/Plateforme$ awk -F"," '{print $3}' personnes.csv
Paris
Lyon
Marseille
Toulouse
rachgid@LaPlateforme:~/Documents/Plateforme$
```

Gestion des processus

Commande pour recenser les processus :

ps : Cette commande permet d'avoir la liste des processus actifs sur le système

ps aux : Cette commande permet aussi de recenser tous les processus actifs sur format étendu.

```

rachgid@LaPlateforme: ~/Documents/Plateforme
rachgid@LaPlateforme:~/Documents/Plateforme$ ps
  PID TTY          TIME CMD
 11185 pts/0    00:00:00 bash
 12539 pts/0    00:00:00 ps
rachgid@LaPlateforme:~/Documents/Plateforme$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root            1  0.0  0.3 168112 12528 ?        Ss   mars19    0:12 /sbin/init
root            2  0.0  0.0      0     0 ?        S    mars19    0:00 [kthreadd]
root            3  0.0  0.0      0     0 ?        I<   mars19    0:00 [rcu_gp]
root            4  0.0  0.0      0     0 ?        I<   mars19    0:00 [rcu_par_gp]
root            5  0.0  0.0      0     0 ?        I<   mars19    0:00 [slub_flushw
root            6  0.0  0.0      0     0 ?        I<   mars19    0:00 [netns]
root           10  0.0  0.0      0     0 ?        I<   mars19    0:00 [mm_percpu_w
root           11  0.0  0.0      0     0 ?        I    mars19    0:00 [rcu_tasks_k
root           12  0.0  0.0      0     0 ?        I    mars19    0:00 [rcu_tasks_r
root           13  0.0  0.0      0     0 ?        I    mars19    0:00 [rcu_tasks_t
root           14  0.0  0.0      0     0 ?        S    mars19    0:02 [ksoftirqd/0
root           15  0.0  0.0      0     0 ?        I    mars19    0:06 [rcu_preempt
root           16  0.0  0.0      0     0 ?        S    mars19    0:00 [migration/0
root           18  0.0  0.0      0     0 ?        S    mars19    0:00 [cpuhp/0]
root           19  0.0  0.0      0     0 ?        S    mars19    0:00 [cpuhp/1]
root           20  0.0  0.0      0     0 ?        S    mars19    0:01 [migration/1
root           21  0.0  0.0      0     0 ?        S    mars19    0:01 [ksoftirqd/1
root           23  0.0  0.0      0     0 ?        I<   mars19    0:00 [kworker/1:0
```

```

rachgid@LaPlateforme: ~/Documents/Plateforme
+  Q  ≡  x

rachgid  10544  0.0  0.4  342136  17664  ?      Sl  mars19  0:00  /usr/libexec
rachgid  10554  0.0  0.5  621928  19124  ?      Ssl mars19  0:00  /usr/libexec
rachgid  10560  0.0  0.3  534376  11392  ?      Ssl mars19  0:00  /usr/libexec
root     10567  0.0  0.0    2480    980  ?      Ss  mars19  0:00  fusermount3
rachgid  10570  0.0  0.2  160804  9240  ?      Sl  mars19  0:03  /usr/libexec
rachgid  10576  0.0  0.8  377528  29872  ?      Ssl mars19  0:00  /usr/libexec
rachgid  10654  0.0  0.7  339988  28136  ?      Ssl mars19  0:00  /usr/libexec
rachgid  10993  0.0  1.6  895592  59168  ?      Sl  mars19  0:00  /usr/bin/gno
rachgid  11002  0.0  1.5  492796  53460  ?      Ssl mars19  0:19  /usr/libexec
rachgid  11035  0.0  0.2  311980  10628  ?      Sl  mars19  0:00  /usr/libexec
rachgid  11071  0.0  0.1  156444  5600  ?      Ssl mars19  0:00  /usr/libexec
rachgid  11185  0.0  0.1   10052   6372  pts/0  Ss  mars19  0:00  bash
rachgid  11336  0.0  0.2  160124   8616  ?      Ssl mars19  0:00  /usr/libexec
root     12157  0.0  0.0      0      0  ?      I   mars19  0:04  [kworker/1:0
root     12354  0.0  0.0      0      0  ?      I   mars19  0:02  [kworker/u4:
root     12380  0.0  0.0      0      0  ?      I   00:27   0:00  [kworker/1:2-
root     12385  0.0  1.2  470500  43180  ?      Ssl 00:27   0:02  /usr/libexec/
root     12519  0.1  0.0      0      0  ?      I   01:24   0:01  [kworker/0:0-
root     12525  0.0  0.0      0      0  ?      I   01:25   0:00  [kworker/u4:3
root     12534  0.0  0.0      0      0  ?      I   01:29   0:00  [kworker/0:2-
root     12536  0.0  0.0      0      0  ?      I   01:34   0:00  [kworker/0:1-
root     12538  2.1  0.0      0      0  ?      I   01:35   0:00  [kworker/u4:1
rachgid  12540  0.0  0.1   11344   4908  pts/0  R+  01:35   0:00  ps aux
rachgid@LaPlateforme:~/Documents/Plateforme$
```

Commande pour fermer les processus :

kill : C'est une commande qui est utilisée pour fermer ou tuer un processus.

Pour fermer un processus spécifique il faut connaître le PID du processus.

Imaginons le PID d'un processus est 143. Dans ce cas il faudra exécuter la commande suivante :

kill 143

```

rachgid@LaPlateforme:~/Documents/Plateforme$ kill 11336
rachgid@LaPlateforme:~/Documents/Plateforme$
```

killall nom_du_processus : Cette commande est exécutée pour tuer tous les processus actifs dans le système ayant comme nom : **nom_du_processus**.

Pour terminer un processus de façon forcée, il faut exécuter la commande suivante :

kill -9 143 : Comme mentionné précédemment, **143** représente le PID du processus

Surveillances des ressources systèmes

Pour mettre en place une surveillance en temps réel de l'utilisation du CPU, de la mémoire et d'autres ressources système. Les commandes permettant d'afficher les informations dans le terminal sont les suivantes :

top

vmstat 1

vmstat 1 > vmstat.csv : Cette commande permet d'enregistrer les informations en temps réel sur un fichier csv.

```
rachgid@LaPlateforme:~/Documents/Plateforme$ top
```

```
top - 03:06:18 up 1 day, 1:19, 1 user, load average: 0,07, 0,04, 0,00
Tâches: 158 total, 1 en cours, 157 en veille, 0 arrêté, 0 zombie
%Cpu(s): 0,5 ut, 0,2 sy, 0,0 ni, 99,3 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3460,1 total, 1311,7 libr, 1038,5 util, 1304,3 tamp/cache
MiB Éch : 976,0 total, 976,0 libr, 0,0 util. 2421,6 dispo Mem
```

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
10191	rachgid	20	0	3822880	289484	131752	S	2,0	8,2	5:36.22	gnome-s+
11002	rachgid	20	0	492732	53180	38696	S	0,3	1,5	0:23.53	gnome-t+
12656	root	20	0	0	0	0	I	0,3	0,0	0:01.40	kworker+
1	root	20	0	168112	12528	9204	S	0,0	0,4	0:12.80	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.07	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+
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+
12	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
13	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tas+
14	root	20	0	0	0	0	S	0,0	0,0	0:02.35	ksoftir+

```

rachgid@LaPlateforme:~/Documents/Plateforme$ vmstat 1
procs -----mémoire----- -échange- -----io----- -système- -----cpu-----
r  b    swpd  libre tampon  cache    si    so    bi    bo    in    cs  us  sy  id  wa  st
3  0      0 1370812 96760 1238908    0    0    6    10   47   31  0  0  99  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  166  157  2  1  98  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  457  482  3  3  94  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  192  142  2  1  97  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  225  154  2  2  96  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  183  129  2  2  96  0
0
1  0      0 1370812 96760 1238908    0    0    0   12  175  103  2  1  97  1
0
1  0      0 1370812 96760 1238908    0    0    0    0  130  142  1  1  98  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  157  115  2  2  96  0
0
1  0      0 1370812 96760 1238908    0    0    0    0  168  160  1  1  98  0
0

```

Scripting avancé

```

rachgid@LaPlateforme: ~/Documents
GNU nano 7.2 plateforme_backup.sh
#!/bin/bash

# Le répertoire à sauvegarder
REPERTOIRE_DIR="/home/rachgid/Documents/Plateforme"

# Le répertoire de sauvegarde
BACKUP_DIR="/home/rachgid/Documents/Plateforme/Sauvegarde"

mkdir -p "$BACKUP_DIR"

# Création du fichier pour stocker l'historique
LOG_FILE="$BACKUP_DIR/sauvegarde.log"

# Fonction pour la sauvegarde
execute_backup() {
    # Créer un nom de fichier unique pour la sauvegarde
    fichier_sauvegarde="$BACKUP_DIR/Sauvegarde_$(date +%Y%m%d_%H%M%S').tar>

# La sauvegarde

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

```

```

# La sauvegarde
tar -cvzf "$fichier_sauvegarde" "$REPERTOIRE_DIR"

# Ajouter une entrée dans le journal de sauvegarde
echo "$(date +%Y-%m-%d %H:%M:%S) : Sauvegarde effectuée : $fichier_sa>
}

# Effectuation de la Sauvegarde
execute_backup

# Afficher le contenu du journal de la sauvegarde
echo "---Historique des Sauvegardes---"
cat "$LOG_FILE"

```

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

rachgid@LaPlateforme: ~/Documents/Plateforme/Sauvegarde

```

GNU nano 7.2      sauvegarde.log
2024-03-20 13:59:06 : Sauvegarde effectuée :
2024-03-20 14:09:10 : Sauvegarde effectuée : /home/rachgid/Documents/Plateforme>

```

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

-

-

-

-

-

-

-

-

Automatisation des mises à jour logicielles

Création d' un script automatisant la recherche de mise à jour des logiciels existants sur le système. Il doit offrir la possibilité à l'utilisateur de procéder à la mise à jour de ces logiciels.

```

rachgid@LaPlateforme: ~/Documents/Plateforme
rachgid@LaPlateforme:~/Documents/Plateforme$ chmod +x mise_a_jour.sh
rachgid@LaPlateforme:~/Documents/Plateforme$ ./mise_a_jour.sh
Recherche des mises à jour disponibles...
[sudo] Mot de passe de rachgid :
Réception de :1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Réception de :3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Réception de :4 http://security.debian.org/debian-security bookworm-security/main Sources [85,0 kB]
Réception de :5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [147 kB]
Réception de :6 http://security.debian.org/debian-security bookworm-security/main Translation-en [88,4 kB]
424 ko réceptionnés en 1s (496 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Aucune mise à jour disponible.
Voulez-vous installer les mises à jour disponibles ? (o/n) o
Lecture des listes de paquets... Fait
```



```
rachgid@LaPlateforme: ~/Documents/Plateforme
Tous les paquets sont à jour.
Aucune mise à jour disponible.
Voulez-vous installer les mises à jour disponibles ? (o/n) o
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Mises à jour installées avec succès.
rachgid@LaPlateforme:~/Documents/Plateforme$ ./mise_a_jour.sh
Recherche des mises à jour disponibles...
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Aucune mise à jour disponible.
Voulez-vous installer les mises à jour disponibles ? (o/n) n
Aucune mise à jour installée.
rachgid@LaPlateforme:~/Documents/Plateforme$ chmod +x mise_a_jour.sh
```

Gestion des dépendances de logicielles

```
Activités Terminal 21 mars 12:10
rachgid@LaPlateforme: ~/Documents/Plateforme
GNU nano 7.2 gestion_install_logiciels.sh
#!/bin/bash

# Fonction pour installer Apache ou Nginx
install_web_server() {
    echo "1. Installation du serveur web"
    read -p "Choisissez le serveur web à installer (apache/nginx) : " server_choice
    case "$server_choice" in
        apache)
            sudo apt update
            sudo apt install apache2
            ;;
        nginx)
            sudo apt update
            sudo apt install nginx
            ;;
        *)
            echo "Choix invalide. Installation du serveur web annulée."
            ;;
    esac
}

# Fonction pour installer phpMyAdmin
install_phpmyadmin() {
    echo "2. Installation de phpMyAdmin"
    sudo apt update
    sudo apt install phpmyadmin
}

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement M-U Annuler
^X Quitter   ^R Lire fich. ^_ Remplacer ^U Coller    ^J Justifier ^_ Aller ligne M-E Refaire
CTRL DROITE
```

```
Activités Terminal 21 mars 12:12
rachgid@LaPlateforme: ~/Documents/Plateforme
GNU nano 7.2 gestion_install_logiciels.sh

}

# Fonction pour installer MySQL ou MariaDB
install_database() {
    echo "3. Installation du système de gestion de base de données"
    read -p "Choisissez le système de gestion de base de données (mysql/mariadb) : " db_choice
    case "$db_choice" in
        mysql)
            sudo apt update
            sudo apt install mysql-server
            ;;
        mariadb)
            sudo apt update
            sudo apt install mariadb-server
            ;;
        *)
            echo "Choix invalide. Installation du système de gestion de base de données annulée."
            ;;
    esac
}

# Fonction pour installer Node.js avec npm
install_nodejs() {
    echo "4. Installation de Node.js avec npm"
    sudo apt update
    sudo apt install nodejs npm
}

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement M-U Annuler
^X Quitter   ^R Lire fich. ^_ Remplacer ^U Coller    ^J Justifier ^_ Aller ligne M-E Refaire

CTRL DROITE
```

```
}

# Fonction pour installer Git
install_git() {
    echo "Installation de Git..."
    sudo apt update
    sudo apt install git
}

# Fonction principale pour exécuter l'installation
main() {
    install_web_server
    install_phpmyadmin
    install_database
    install_nodejs
    install_git
}

# Exécution de la fonction principale
main

exit 0

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement M-U Annuler
^X Quitter   ^R Lire fich. ^_ Remplacer ^U Coller    ^J Justifier ^_ Aller ligne M-E Refaire

CTRL DROITE
```

Sécuriser ses scripts

```
GNU nano 7.2 create_csv.py
#!/usr/bin/env python3
import csv
import subprocess

filename = 'file.csv'
liste = [
{'Nom': 'Jean', 'Age': '25 ans', 'Ville': 'Paris'},
{'Nom': 'Marie', 'Age': '30 ans', 'Ville': 'Lyon'},
{'Nom': 'Pierre', 'Age': '22 ans', 'Ville': 'Marseille'},
{'Nom': 'Sophie', 'Age': '35 ans', 'Ville': 'Toulouse'},
]

with open(filename, 'w') as f:
    fields = ['Nom', 'Age', 'Ville']
    writer = csv.DictWriter(f, fieldnames=fields)
    writer.writeheader()
    writer.writerows(liste)

permissions = "774" #774 TOUS LES DROITS POUR LE PROPRIÉTAIRE ET LE GROUPE
                    #LECTURE SEUL POUR LES AUTRES

subprocess.run(["chmod", permissions, filename])

owner = "elias"
group = "sudo"

subprocess.run(["chown", f"{owner}:{group}", filename])

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

Utilisation

Pour mettre en place une sécurisation des scripts, dans l'exemple ci-dessus nous importons la librairie subprocess en tant que module , nous utilisons donc **chmod** et **chown** qui permet de spécifier quel utilisateur/groupe d'utilisateurs peut ouvrir, lire, modifier ou exécuter le script.

chmod : Le mot "chmod" est une contraction de "change mode". Cette commande permet aux utilisateurs de spécifier qui peut lire, écrire et exécuter un fichier.

chown: Le mot "chown" est une contraction de "change owner". Cette commande permet de modifier le propriétaire et le groupe propriétaire d'un fichier ou d'un répertoire.

Utilisation d'API Web dans un Script

Ci dessous se trouve des screen d'un script qui exploite de manière sécurisée une API Web en utilisant HTTPS et une authentification par token. Il intègre une gestion des erreurs pour anticiper les problèmes avec l'API et assure une journalisation des requêtes et réponses pour des raisons de cybersécurité.



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

# Variables
API_BASE_URL="https://api.github.com"
USERNAME="abdou-rachidou"
TOKEN="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"

# Fonction pour effectuer une requête GET à l'API
function make_request() {
    local endpoint="$1"
    local response=$(curl -s -H "Authorization: token $TOKEN" "$API_BASE_URL/$endpoint")

    # Vérifier si la requête a réussi
    if [[ $? -ne 0 ]]; then
        echo "Erreur: La requête vers $endpoint a échoué."
        exit 1
    fi

    # Vérifier si la réponse contient des données
    if [[ -z $response ]]; then
        echo "Erreur: Aucune réponse reçue de $endpoint."
        exit 1
    fi

    echo "$response"
}

# Fonction pour afficher les informations d'un utilisateur
```

[Lecture de 55 lignes]

^G Aide	^O Écrire	^W Chercher	^K Couper	^T Exécuter	^C Emplacement	M-U Annuler
^X Quitter	^R Lire fich.	^_ Remplacer	^U Coller	^J Justifier	^_ Aller ligne	M-E Refaire

```

+      rachgid@LaPlateforme: ~/Documents/Plateforme
GNU nano 7.2      exploitation_API.sh
function get_user_info() {
    local user="$1"
    make_request "users/$user"
}

# Fonction pour afficher les dépôts d'un utilisateur
function get_user_repos() {
    local user="$1"
    make_request "users/$user/repos"
}

# Exemple d'utilisation
echo "Informations sur l'utilisateur $USERNAME :"
user_info=$(get_user_info "$USERNAME")
if [[ $? -eq 0 ]]; then
    echo "$user_info" | jq .
else
    echo "Impossible de récupérer les informations sur l'utilisateur $USERNAME."
fi

echo "Dépôts de l'utilisateur $USERNAME :"
user_repos=$(get_user_repos "$USERNAME")
if [[ $? -eq 0 ]]; then
    echo "$user_repos" | jq .
else
    echo "Impossible de récupérer les dépôts de l'utilisateur $USERNAME."
fi

```

^G Aide
^X Quitter

^O Écrire
^R Lire fich.

^W Chercher
^_ Remplacer

^K Couper
^U Coller

^T Exécuter
^J Justifier

^C Emplacement
^_ Aller ligne

M-U Annuler
M-E Refaire