

gestion des fichiers et des répertoires

ghandri ibtihel

exercice 1:

whoami → Affiche le nom de l'utilisateur courant.

```
ghandri@ghandri-VirtualBox:~$ whoami
ghandri
ghandri@ghandri-VirtualBox:~$
```

id → Affiche l'identifiant de l'utilisateur (UID), l'identifiant du groupe (GID) et les groupes auxquels l'utilisateur appartient.

```
ghandri@ghandri-VirtualBox:~$ id
uid=1000(ghandri) gid=1000(ghandri) groups=1000(ghandri),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),133(lxd),134(sambashare)
ghandri@ghandri-VirtualBox:~$
```

groups → Affiche les groupes d'appartenance de l'utilisateur.

```
ghandri@ghandri-VirtualBox:~$ groups
ghandri adm cdrom sudo dip plugdev lpadmin lxd sambashare
ghandri@ghandri-VirtualBox:~$
```

hostname → Affiche le nom de la machine.

```
ghandri@ghandri-VirtualBox:~$ hostname
ghandri-VirtualBox
ghandri@ghandri-VirtualBox:~$
```

clear → Efface le contenu du terminal.

```
ghandri@ghandri-VirtualBox:~$ clear
```

+ entrée

—>

```
ghandri@ghandri-VirtualBox:~$
```

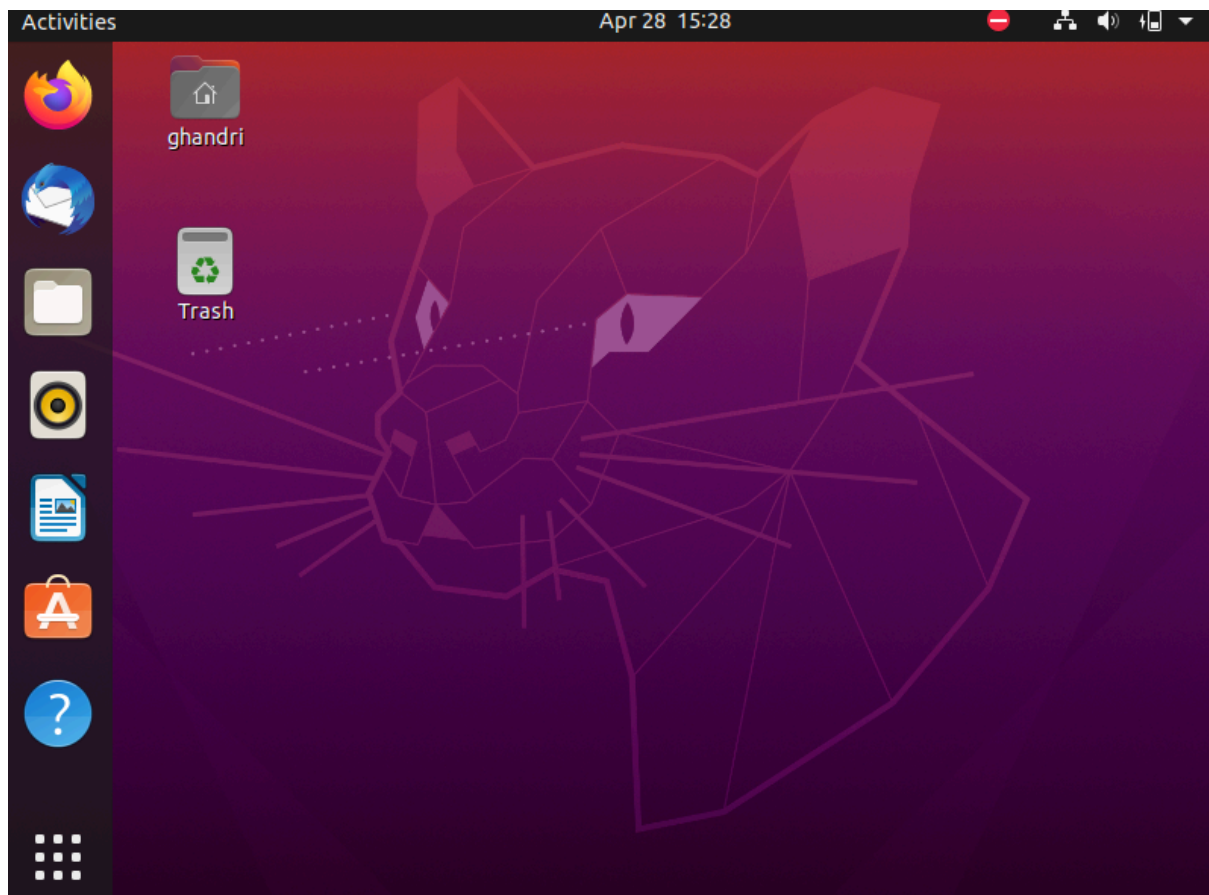
history → Affiche l'historique des commandes déjà tapées.

```
ghandri@ghandri-VirtualBox:~$ history
```

```
1  contab -e
2  sudo apt install contab
3  sudo apt install contab
4  contab -e
5  sudo apt install contab
6  sudo apt update
7  sudo apt install contab
8  sudo apt install cron
9  sudo apt update
10 sudo apt install cron
11 contab -e
12 crontab -e
13 (while true;do echo -n A/tmp/log;sleep 1;done)&
14 ls
15 cd
16 clear
17 whoami
18 id
19 groups
20 hostname
21 clear
22 history
```

```
ghandri@ghandri-VirtualBox:~$
```

exit → Ferme le terminal.



exercice 2:

```
ghandri@ghandri-VirtualBox:~$ help echo
echo: echo [-neE] [arg ...]
    Write arguments to the standard output.

    Display the ARGs, separated by a single space character and followed by a
    newline, on the standard output.

    Options:
      -n      do not append a newline
      -e      enable interpretation of the following backslash escapes
      -E      explicitly suppress interpretation of backslash escapes

    'echo' interprets the following backslash-escaped characters:
      \a      alert (bell)
      \b      backspace
      \c      suppress further output
      \e      escape character
      \E      escape character
      \f      form feed
      \n      new line
      \r      carriage return
      \t      horizontal tab
      \v      vertical tab
      \\      backslash
      \0nnn   the character whose ASCII code is NNN (octal).  NNN can be
              0 to 3 octal digits
      \xHH    the eight-bit character whose value is HH (hexadecimal).  HH
```

```

\n      new line
\r      carriage return
\t      horizontal tab
\v      vertical tab
\\      backslash
\0nnn   the character whose ASCII code is NNN (octal).  NNN can be
         0 to 3 octal digits
\xHH     the eight-bit character whose value is HH (hexadecimal).  HH
         can be one or two hex digits
\UHHHH   the Unicode character whose value is the hexadecimal value HHHH
         HHHH can be one to four hex digits.
\UHHHHHHHH the Unicode character whose value is the hexadecimal value
         HHHHHHHH. HHHHHHHH can be one to eight hex digits.

Exit Status:
Returns success unless a write error occurs.
ghandri@ghandri-VirtualBox:~$

```

a) `echo $USERNAME` → Affiche le nom de l'utilisateur courant :

```

ghandri@ghandri-VirtualBox:~$ echo $USERNAME
ghandri
ghandri@ghandri-VirtualBox:~$

```

b) `echo $HOSTNAME` → Affiche le nom de la machine. :

```

ghandri@ghandri-VirtualBox:~$ echo $HOSTNAME
ghandri-VirtualBox
ghandri@ghandri-VirtualBox:~$

```

c) `echo $HOME` → Affiche le répertoire personnel de l'utilisateur

```

ghandri@ghandri-VirtualBox:~$ echo $HOME
/home/ghandri
ghandri@ghandri-VirtualBox:~$

```

d) `echo $SHELL` → Affiche le nom de l'interpréteur de commandes utilisé (par exemple : `bash`)

```

ghandri@ghandri-VirtualBox:~$ echo $SHELL
/bin/bash
ghandri@ghandri-VirtualBox:~$

```

e) `echo $UID` → Affiche l'identifiant (UID) de l'utilisateur en cours :

```

ghandri@ghandri-VirtualBox:~$ echo $UID
1000
ghandri@ghandri-VirtualBox:~$

```

exercice 3;

`mkdir` : créer un répertoire (`mkdir` répertoire)

cd : changer de répertoire (cd nom_de_l répertoire)

```
ghandri@ghandri-VirtualBox:~$ mkdir lisi
ghandri@ghandri-VirtualBox:~$ cd lisi
ghandri@ghandri-VirtualBox:~/lisi$
```

touch : créer un fichier vide (touch fichier)

```
ghandri@ghandri-VirtualBox:~/lisi$ touch ghandriibtihel
ghandri@ghandri-VirtualBox:~/lisi$
```

cp : copier un fichier ou un répertoire (cp source destination)

```
ghandri@ghandri-VirtualBox:~/lisi/groupe$ cp g c
ghandri@ghandri-VirtualBox:~/lisi/groupe$
```

mv : déplacer ou renommer un fichier ou un répertoire (mv source destination)

```
ghandri@ghandri-VirtualBox:~/lisi/groupe$ mv g ghandri
ghandri@ghandri-VirtualBox:~/lisi/groupe$ ls
c  ghandri
ghandri@ghandri-VirtualBox:~/lisi/groupe$
```

rm : supprimer un fichier (rm fichier) :

```
ghandri@ghandri-VirtualBox:~/lisi/groupe$ rm ghandri
ghandri@ghandri-VirtualBox:~/lisi/groupe$
```

rmdir : supprimer un répertoire vide (rmdir répertoire)

```
ghandri@ghandri-VirtualBox:~/lisi/groupe$ mkdir lisi11
ghandri@ghandri-VirtualBox:~/lisi/groupe$ rmdir lisi11
ghandri@ghandri-VirtualBox:~/lisi/groupe$
```

exercice4:

1. Répertoire personnel et création de TP3 a) Accéder à votre répertoire personnel b) Créer un répertoire nommé TP3 c) Sans changer de répertoire, créer un répertoire rep1 dans TP3 d) Sans changer de répertoire, afficher le contenu du répertoire TP3

```
ghandri@ghandri-VirtualBox: ~
ghandri@ghandri-VirtualBox:~/lisi/groupe$ cd ~
ghandri@ghandri-VirtualBox:~$ mkdir tp3
ghandri@ghandri-VirtualBox:~$ mkdir tp3/rep1
ghandri@ghandri-VirtualBox:~$ ls tp3
rep1
ghandri@ghandri-VirtualBox:~$
```

2/Travail dans le répertoire rep1

- a) Se placer dans le répertoire rep1
- b) Vérifier que vous êtes bien dans rep1
- c) Afficher tout le contenu du répertoire rep1
- d) Créer un fichier vide nommé f3
- e) Créer le répertoire rep4
- f) Vérifier le contenu du répertoire rep1

```
ghandri@ghandri-VirtualBox:~$ cd tp3/rep1
ghandri@ghandri-VirtualBox:~/tp3/rep1$ pwd
/home/ghandri/tp3/rep1
ghandri@ghandri-VirtualBox:~/tp3/rep1$ ls -a
.  ..
ghandri@ghandri-VirtualBox:~/tp3/rep1$ touch f3
ghandri@ghandri-VirtualBox:~/tp3/rep1$ mkdir rep4
ghandri@ghandri-VirtualBox:~/tp3/rep1$ ls
f3  rep4
ghandri@ghandri-VirtualBox:~/tp3/rep1$
```

3. Sans changer de répertoire

- a) Créer dans rep4 un fichier vide nommé f4
- b) Faire une copie de f4 nommée f5 dans rep4
- c) Vérifier le contenu du répertoire rep

```
ghandri@ghandri-VirtualBox:~/tp3/rep1$ touch rep4/f4
ghandri@ghandri-VirtualBox:~/tp3/rep1$ cp rep4/f4 rep4/f5
ghandri@ghandri-VirtualBox:~/tp3/rep1$ ls rep4
f4  f5
ghandri@ghandri-VirtualBox:~/tp3/rep1$
```

4. Travail depuis le répertoire rep4

- a) Se placer dans le répertoire tp3
- b) Créer les répertoires rep2 et rep3
- c) créer le fichier vide f1 dans rep2
- d) Faire une copie de f1 nommée f2 dans rep2
- e) Vérifier le contenu du répertoire rep2 depuis rep4

```
ghandri@ghandri-VirtualBox:~/tp3/rep1$ cd ..
ghandri@ghandri-VirtualBox:~/tp3$ mkdir rep2 rep3
ghandri@ghandri-VirtualBox:~/tp3$ touch rep2/f1
ghandri@ghandri-VirtualBox:~/tp3$ cp ../rep2/f1 ../rep2/f2
cp: cannot stat '../rep2/f1': No such file or directory
ghandri@ghandri-VirtualBox:~/tp3$ ls ../rep2
ls: cannot access '../rep2': No such file or directory
ghandri@ghandri-VirtualBox:~/tp3$ ls rep2
f1
ghandri@ghandri-VirtualBox:~/tp3$
```

5. Retour dans TP3 et manipulations

- a) Revenir directement dans le répertoire TP3
- b) Faire une copie de f1 (dans rep2) nommée f6 dans TP3
- c) Vérifier le contenu du répertoire TP3*
- d) Depuis TP3, déplacer f6 dans rep4 sans changer son nom
- e) Depuis TP3, déplacer f4 dans TP3 en le renommant f8
- f) Renommer f8 en f9

```
ghandri@ghandri-VirtualBox:~/tp3$ cd ..  
ghandri@ghandri-VirtualBox:~$ cd tp3  
ghandri@ghandri-VirtualBox:~/tp3$ cp rep2/f1 f6  
ghandri@ghandri-VirtualBox:~/tp3$ ls  
f6  rep1  rep2  rep3  
ghandri@ghandri-VirtualBox:~/tp3$ mv f6 rep1/rep4/  
ghandri@ghandri-VirtualBox:~/tp3$ mv rep1/rep4/f4 f8  
ghandri@ghandri-VirtualBox:~/tp3$ mv f8 f9  
ghandri@ghandri-VirtualBox:~/tp3$
```

6. Dessiner la nouvelle arborescence du répertoire TP3 après toutes les opérations précédentes.

```
ghandri@ghandri-VirtualBox:~/tp3$ ls -R  
.:  
f9  rep1  rep2  rep3  
  
./rep1:  
f3  rep4  
  
./rep1/rep4:  
f5  f6  
  
./rep2:  
f1  
  
./rep3:  
ghandri@ghandri-VirtualBox:~/tp3$
```

exercice 5

1 mkdir user1

cd user1

mkdir progs docs

cd progs

mkdir c java

cd c

mkdir sources bin

cd c

```
cd progs
cd ..
cd docs
mkdir linux courSE
a)cd ~/user1/progs/java
touch programme1.java
b)touch user1/progs/c/sources programme1.c
C)touch user1/docs/linux introduction.odt
d)cp user1/progs/c/sources/programme1.c user1/docs/linux
e)mv user1/docs/linux/introduction.odt user1/docs/courSE
f)mv programme1.java programme2.java
g)rm -r user1/c/bin
h)ls
```

exercice 6:

```
1/mkdir -p docs
touch docs/notes.txt
2/echo "Ceci est une note." >> docs/notes.txt
3/cat docs/notes.txt
4/mkdir -p TP2
cp docs/notes.txt TP2/
```

exercice 7:

```
1/find ~ -name "notes_tp2.txt"
2/find TP2 -name "*.txt"
```

exercice 8:

```
1/ls TP2
2/ls TP2 > liste.txt
3/cat liste.txt
```

exercice 9:

```
1/tar -cvf tp2 archive.tar TP2/
2/déjà faite
3/mkdir TP2_extrait
tar -xvf tp2_archive.tar -C TP2_extrait/
```

exercice 10:

La suppression en Linux dépend du type d'élément (fichier ou répertoire) et de son contenu.

- **Suppression d'un fichier :**

La suppression d'un fichier se fait avec la commande `rm`. Cette opération est simple car un fichier ne contient pas d'autres éléments. Une fois supprimé, le fichier est définitivement effacé (sauf récupération spéciale).

- **Suppression d'un répertoire vide :**

Pour supprimer un répertoire vide, on utilise la commande `rmdir`. Cette commande fonctionne uniquement si le répertoire ne contient aucun fichier ni sous-répertoire.

- **Suppression d'un répertoire non vide :**

Un répertoire non vide ne peut pas être supprimé avec `rmdir`. Il faut utiliser la commande `rm -r` (récursive), qui supprime le répertoire ainsi que tout son contenu (fichiers et sous-répertoires).

Exercice 11

- **`?`** : désigne un seul caractère quelconque
- **`[c1-c2]` ou `[c1c2...]`** : désigne un caractère parmi ceux indiqués
 - **`[a-z]`** : une lettre minuscule
 - **`[AEIOUY]`** : une voyelle majuscule
- **`\` (*antislash*)** : permet d'annuler la signification spéciale du caractère qui suit

Exercice 12 :

1. Création des fichiers (en une seule commande)

`touch annee1 Annee2 annee4 annee45 annee41 annee510
annee_blanche`

2. Créer le répertoire Year

mkdir Year

3. Déplacer tous les fichiers dans Year (en une seule commande)

mv annee Annee2 Year/*

4. Lister les fichiers

(à exécuter dans le répertoire Year ou avec Year / devant)

a) Se terminant par 5

*ls *5*

b) Commenant par annee4

*ls annee4**

c) Commenant par annee4 et ayant 7 caractères

ls annee4?

d) Contenant la chaîne ana

*ls *ana**

e) Commenant par a ou A

*ls [aA]**

5. Copier les fichiers dont l'avant-dernier caractère est 4 ou 1 vers /tmp

*cp *[41]? /tmp/*

