

## TM 2 : Environnement Informatique.

COSTA SANTOS Pedro Groupe 2

FEIKIH HASSEN Yassine Groupe 2

### Introduction :

L'objectif de ce TM 2 est de se familiariser avec l'interpréteur de commandes Linux (shell), avec des plusieurs type d'opérations. Celles-ci sont ici exposées par des commandes de manipulation de fichiers et de dossiers et de permissions, c'est-à-dire comment créer un dossier, le déplacer, le supprimer ou bien lui redéfinir ses permissions par exemple. Durant ce TM, le système d'exploitation utilisé a été Ubuntu 20.04 et on a utilisé ça documentation sur le site « [doc.ubuntu-fr.org](http://doc.ubuntu-fr.org) ». Ce TM a réalisé par COSTA SANTOS Pedro ainsi que FEIKIH HASSEN Yassine en L1 Informatique du groupe 2.

1)

```
pedro@pedro-VirtualBox:~$ pwd
/home/pedro
pedro@pedro-VirtualBox:~$
```

La commande pour connaitre dans quel répertoire est lancé le terminal est la commande est « `pwd` ». Elle signifie « *print working directory* ».

2)

```
pedro@pedro-VirtualBox:~/Bureau$ mkdir ./MonDossier
pedro@pedro-VirtualBox:~/Bureau$ ls
MonDossier
pedro@pedro-VirtualBox:~/Bureau$
```

La commande qui permet de créer un dossier est « `mkdir` » suivi de l'emplacement où l'on veut mettre le fichier et le nom du fichier que l'on veut créer. Puis afin de vérifier le résultat on utilise la commande « `ls` » qui permet de lister les fichiers et les dossiers du répertoire courant.

3)

```
pedro@pedro-VirtualBox:~/Bureau$ cd ./MonDossier/
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

La commande permettant de changer de répertoire est « `cd` » (cela signifie « *change direct* ») suivi du lieu de destination (en chemin absolue ou relatif).

4)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ touch ./monFichier1
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
monFichier1
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Pour créer un fichier vide, nous devons utiliser la commande « touch » suivi de l'emplacement où l'on veut mettre le fichier suivi de son nom, dans ce cas-ci, on va le mettre dans le répertoire courant « ./ » et avec le nom « monFichier1 ».

5) Pour éditer un fichier sous Windows, on peut utiliser « bloc note », « NotePad++ », « VScode », « sublim text » ou « atom », cela sont les plus populaires. Sous MacOSX, on peut utiliser « Xcode », « TextEdit » ou « Vscod ».

6)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
monFichier1
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ mv ./monFichier1 ./monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Pour renommer un fichier, nous avons utilisé la commande « mv » suivi de l'emplacement et l'ancien nom puis du même emplacement et du nouveau nom du fichier. Cette commande permet de copier le fichier « monFichier1 », et de le coller dans le même répertoire en utilisant le nom « monFichier2 ».

7)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cp ./monFichier2 ./monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
monFichier2 monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Pour copier un fichier, on utilise la commande « cp » (*copy*) suivi de l'emplacement du fichier et de son nom et puis on met l'emplacement où on veut le coller avec son nom.

8)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ mkdir ./MonDossier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2 monFichier2 monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Comme à la question 2, il faut utiliser la commande « mkdir ».

9)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2  monFichier2  monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ mv ./monFichier3 ./MonDossier2/monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2  monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls ./MonDossier2/
monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

La commande permettant de déplacer un fichier est « mv » suivi de l'emplacement du fichier actuel et puis suivi du nouveau emplacement.

10)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2  monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls ./MonDossier2/
monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -R
.:
MonDossier2  monFichier2

./MonDossier2:
monFichier3
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

On peut utiliser la commande « ls » pour afficher le nom des fichiers et des dossiers du répertoire courant. On peut aussi lui indiquer un chemin afin d'aller voir dans un dossier, Ou bien on peut utiliser le paramètre « -R » dans la commande « ls » afin d'afficher tous les sous dossier et leurs contenus.

11)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
-rw-rw-r-- 1 pedro pedro  0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Grâce à la commande « ls » suivi de l'argument « -l », on peut voir que le « user » et le « group » a des permissions « rw- », le « r » signifie qu'il a le droit de lire (« read »), le « w » signifie qu'il a le droit d'écrire (« write ») mais il n'a pas le « x » ce qui signifie qu'ils n'ont pas le droit d'exécuter. On peut voir que le groupe « other » n'a que les droits « r-- » donc ils n'ont que le droit de lire mais pas de modifier, ni d'exécuter.

12)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
-rw-rw-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ chmod ug-r ./monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
--w--w-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

La commande « chmod » permet de modifier les droits. Pour retirer les droits de lecture au « user » et au « group », il faut mettre l'argument « ug-r » et le nom du fichier en l'occurrence « ./monFichier2 », dans le premier argument le « u » signifie utilisateur et le « g » signifie groupe, puis le « - » permet de retirer des permissions et le « r » est la permission qui est retirée.

13)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
--w--w-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ chmod ug+r ./monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
-rw-rw-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Pour vérifier le résultat, on utilise la commande « ls -l », et pour rétablir les droits initiaux, on utilise la même commande qu'à la question précédente mais on remplace le « - » par un plus « + » car on ne veut plus retirer des droits mais en ajouter.

14)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 5 11:50 MonDossier2
```

Grâce à la commande « ls -l », on peut voir que le « user » et le « group » ont des permissions « rwx », le « r » signifie qu'il a le droit de lire, le « w » signifie qu'il a le droit d'écrire et le « x » signifie qu'il a le droit d'exécuter. On peut voir que le groupe « other » n'a que les droits « r-x » donc il n'a le droit de lire et d'exécuter mais pas de modifier.

15) Pour un répertoire, la permission « x » définit si un utilisateur pourra ou non ouvrir le dossier.

16)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ chmod a-x ./MonDossier2/
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cd ./MonDossier2/
bash: cd: ./MonDossier2/: Permission non accordée
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drw-rw-r-- 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
-rw-rw-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

En effet lorsque l'on retire le droit « x » à toutes les utilisateurs grâce à la commande « `chmod a-x ./MonDossier2` » (l'argument « a » signifie à toutes les utilisateur, le « - » permet de retirer des droits et le « x » est le droit que l'on veut retirer). Après avoir retiré le droit « x » à toutes les utilisateurs, on peut voir qu'on n'a plus le droit d'ouvrir ce répertoire.

17)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drw-rw-r-- 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
-rw-rw-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ chmod a+x ./MonDossier2/
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls -l
total 4
drwxrwxr-x 2 pedro pedro 4096 oct. 11 17:39 MonDossier2
-rw-rw-r-- 1 pedro pedro 0 oct. 11 17:35 monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cd ./MonDossier2/
pedro@pedro-VirtualBox:~/Bureau/MonDossier/MonDossier2$
```

Pour rétablir les droits initiaux, on utilise la même commande qu'à la question précédente mais on remplace le « - » par un plus « + » car on ne veut plus retirer des droits mais en ajouter.

18) Pour un fichier ordinaire la permission « x » signifie l'ouverture de ce fichier.

19) On peut remarquer que sur Linux on peut définir des permissions avec plus de précisions (que ce soit avec les groupes ou bien en terme lire/écrire/exécuté) que sur Windows.

20)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2  monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ rm ./monFichier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Pour supprimer le fichier « monFichier2 », j'ai utilisé la commande « `rm` » (qui signifie « remove ») suivi de l'emplacement du fichier que je veux supprimer.

21)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
MonDossier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ rm -r ./MonDossier2
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Pour supprimer le dossier « MonDossier2 », j'ai utilisé la commande « rm »(qui signifie « remove ») suivi de l'argument « -r » (qui permet de supprimé un répertoire) et suivi de l'emplacement du dossier.

22)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ echo -e '#!/bin/bash\necho Bonjour'>monScript.sh
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ls
monScript.sh
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Grace à la commande donnée dans le sujet, j'ai créé un script.

23)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cat ./monScript.sh
#!/bin/bash
echo Bonjour
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

J'ai vérifié le script grâce à la commande « cat », comme on peut voir ci-dessus.

24)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ bash monScript.sh
Bonjour
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

J'ai donc exécuté le script avec la commande « bash monScript.sh ».

25)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ./monScript.sh
bash: ./monScript.sh: Permission non accordée
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Lorsque je tente d'exécuter le script, il ne se lance pas car je n'ai pas les permissions.

26)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ chmod a+x ./monScript.sh
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ ./monScript.sh
Bonjour
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

Grace a la commande « `chmod a+x ./monScript.sh` », j'ai donner les droits a tous les utilisateurs d'exécuter ce script puis avec la « `./monScript.sh` », j'ai lancé le script et il a afficher « Bonjour ».

27)

```
pedro@pedro-VirtualBox:~$ ls / | grep "^m"
media
mnt
pedro@pedro-VirtualBox:~$
```

Je suis allé dans « `/` » afin d'avoir un emplacement avec des fichiers (dont certain commencent pas « `m` »), puis j'ai exécuter en même temps deux commande en les séparant par « `|` », la première parti était la commande « `ls` » permettant de récupérer tous les noms des fichiers dans le répertoire courant, puis la deuxième commande « `grep "^m"` » permet de recherché la chaine de caractère qu'on lui donnera entre guillemet dans ce cas-ci on a mis cela « `^m` », le caractère « `^` » donne l'ordre que le caractère suivant devra être le première.

28) Les trois commandes permettent de retourner au répertoire de travail sont :

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cd
pedro@pedro-VirtualBox:~$ cd ./Bureau/MonDossier/
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cd ../../
pedro@pedro-VirtualBox:~$ cd ./Bureau/MonDossier/
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ cd /home/pedro
pedro@pedro-VirtualBox:~$
```

- « `cd` », car lorsque la commande « `cd` » ne prend aucun argument alors elle redirigera vers le répertoire de travail.
- « `cd ../../` », on peut utiliser la commande « `cd` » et lui donner en argument un chemin relatif
- « `cd /home/pedro` », on peut utiliser la commande « `cd` » et lui donner en argument chemin absolu comme par exemple « `/home/nom_d'utilisateur` »

29)

```
pedro@pedro-VirtualBox:~/Bureau/MonDossier$ echo $HOME
/home/pedro
pedro@pedro-VirtualBox:~/Bureau/MonDossier$
```

La commande qui donne le chemin absolu du répertoire de travail est « `echo $HOME` », « `echo` » est une commande permettant d'affiche des valeurs et « `$HOME` » est une variable contenant le chemin absolu du répertoire de travail.

30)

```
pedro@pedro-VirtualBox:~/Bureau$ umask -p
umask 0002
pedro@pedro-VirtualBox:~/Bureau$ touch ./avant
pedro@pedro-VirtualBox:~/Bureau$ ls -l
total 4
-rw-rw-r-- 1 pedro pedro    0 oct.  11 22:53 avant
drwxrwxr-x 3 pedro pedro 4096 oct.  11 17:39 MonDossier
pedro@pedro-VirtualBox:~/Bureau$ umask 052
pedro@pedro-VirtualBox:~/Bureau$ touch ./apres
pedro@pedro-VirtualBox:~/Bureau$ ls -l
total 4
-rw--w-r-- 1 pedro pedro    0 oct.  11 22:54 apres
-rw-rw-r-- 1 pedro pedro    0 oct.  11 22:53 avant
drwxrwxr-x 3 pedro pedro 4096 oct.  11 17:39 MonDossier
pedro@pedro-VirtualBox:~/Bureau$
```

On peut voir que sur cette machine le umask est à 002 par défaut comment on peut le voir avec la commande « umask -p » ci-dessus, on peut aussi vérifier cela en créer un nouveau fichier du nom de « avant » et en affichant les permissions, on peut voir que « user » et « group » ont le droit d'écrire et le lire mais pas le droit d'exécuter et « other » n'a le droit que de lire. En définissent le umask à 052 avec la commande « umask 052 » alors rien ne changera pour le « user »(utilisateurs) et pour le « other »(autres) mais en revanche pour le « group » cela change, car il perde le droit de lire et donc n'a plus que le droit d'écrire, cela peut se vérifier en créant un nouveau fichier (après la modification du umask) qui se nomme « apres » et en affichant les permissions afin de vérifier.

Conclusion :

Ce travail nous a permis, d'apprendre à utiliser les bases du terminal Linux. Nous avons appris à créer des dossiers et des fichiers, à les déplacer, à les supprimer, à changer leurs droits et d'autres commandes. Nous avons également vu que sous Linux, il existe trois types de groupes (« user », « group », « other ») et trois types d'autorisations : lecture, écriture et exécution. Tous ces types d'autorisations nous donnent une large maîtrise sur nos données et une plus grande liberté donc nous pouvons dire que Linux a un meilleur system de permissions que Windows.