

Rapport du TP 01

Système d'exploitation

Binôme:

- MADI Sara - G1
- FERCHICHI Manel - G2

Exo1:

D'abord il y a deux façons d'exécuter les commandes sous linux, **soit en utilisant l'interface graphique** et ouvrir le terminal, **soit en utilisant le mode console**.

-Pour accéder à la console : **CTRL + ALT +Fn** (n selon le num de la console >1)

-Pour quitter la console: **CTRL+ALT+F1**

1- On ouvre le terminal et tap les commandes de l'exercice:

Commande	Elle sert à quoi ?
echo bonjour tout le monde	echo, pour afficher un message
i=18	l'affectation de la val 18 à i
echo valeur affectée a i est \$i	echo, pour afficher, le "\$" pour récupérer la valeur stockée dans la variable i.
date	afficher la date actuelle et l'heure.
uname -a	donner des info sur le système linux utilisé.
whoami	afficher l'utilisateur courant
clear	replacer la ligne de commande en haut de l'écran
history	afficher l'historique des commandes
man man	obtenir le manuel de la commande man
exit	pour terminer la session

Exo2:

Commande cd : change directory	
cd	permet de revenir au répertoire /home/utilisateur
cd-	permet de revenir au répertoire précédent
cd ..	permet de remonter au répertoire parent
cd /	permet de remonter à la racine de l'ensemble du système de fichiers
Commande pwd : print working directory Affiche le répertoire de travail	
Commande ls : list lister un répertoire	
ls -l	Permet un affichage détaillé du répertoire (permissions d'accès, le nombre de liens physiques, le nom du propriétaire et du groupe, la taille en octets, et l'horodatage).

Exemple a partir de ls -l et explication:

drwxr-xr-x	6	user	user	192	Mar 22 13:53	StudioProjects
↓	↓	↓	↓	↓	↓	↓
les droits d'accès	nombre de noms attribués au fichier	nom utilisateur	Groupe d'utilisateurs	Taille en oct	Date et heure	nom fichier

Exo3:

Pour commencer à programmer il nous faut sûrement un text editor ; sur la machine virtuelle Linux on ouvre ce text editor avec deux méthode :

1. Aller à UBUNTU trouver le text editor
2. Sinon on utilisant la commande ~ **gedit** ~ sur le terminal
techniquement
(père) Terminal ⇒ Gedit (fils)

Problème :mais puisque Terminal c' est le père si on le ferme le gedit se ferme automatiquement aussi ce qui n' est pas efficace puisque on a besoin d' exécution du programme écrit .

Solution: à l'aide de la commande $\Rightarrow \sim \text{nohup gedit \& } \sim$ ou $\sim \text{gedit \& } \sim$
Pour arrêter la
fermeture text editor lorsque
on ferme terminal

PROGRAMMATION SHELL ET PROGRAMMATION EN C :

- *Programmation shell:*

Après avoir ouvert le text editor on le sauvegarde
save \Rightarrow name file **nom.sh** \Rightarrow click save

Ecrire le texte donné dans l'exercice :

```
# !bin/bash
echo "hello world "
```

Executer ce programme à l'aide de la commande
 $\sim \text{bash hello.c } \sim \Rightarrow \sim \text{bash nomscript.type } \sim$

Remarque :

On peut effacer la première ligne le type de script à exécuter is set par défaut bash

On peut donner le droit et rendre le fichier exécutable sans l'écriture de la commande bash

Details :

ls -l hello.sh \Rightarrow Affiche le nom du fichier + **r w** (écriture lecture) sa couleur est blanche
Il faut ajouter le type exécutable (x) (il deviendra vert)
La commande : $\sim \text{chmod 746 hello.sh } \sim$ chmod=changer le mode de file, 7=x, 4=r, 6=w
maintenant on peut exécuter le script sans bash juste en écrivant :
 $\sim \text{./nomfile.type } \sim$ (S'il existe sûre le desktop sinon $\sim \text{./fichier/nom file.type } \sim$)

Quelques lignes de programmation shell :
condition :

```
if [ condition ] then
else if [ condition ] then
else
fi
```

Opérateurs de comparaisons (conditions numérique):

Equal to	if [“ \$a” -eq “b”]	a et b des entiers et les espaces sont importants
Not equal	if [“ \$a” -ne “b”]	a et b des entiers et les espaces sont importants
Superior	if [“ \$a” -gt “b”]	a et b des entiers et les espaces sont importants
Superior or equal	if [“ \$a” -ge “b”]	a et b des entiers et les espaces sont importants
Inferior	if [“ \$a” -lt “b”]	a et b des entiers et les espaces sont importants
Inferior or equal	if [“ \$a” -le “b”]	a et b des entiers et les espaces sont importants

- *Programmation en C :*

Sauvegarder le fichier sous type .C

Il est nécessaire d' installer le gcc en cas de son absence sur la machine virtuelle .
pour l' installer on utilise ses deux commandes sur Terminal une est suffisante :

1. ~ `sudo apt install gcc` ~
2. ~ `sudo apt get install build -essential` ~

On compile le programme grâce à la commande :

~ `gcc nomsript.typedefile -nomscript` ~

Ou suivre les même méthode pour rendre le type du file exécutable grâce à la commande `chmod` et l'exécuter avec (`./nomscript.typefile`)

Dans la partie deux on a écrit le code 'hello world ' en C

```
#include <stdio.h>

int main()
{
    printf("Hello World");

    return 0;
}
```

exo4 :

les commande avec mman pour avoir plus de details :

Commande	Rôle et résultat	
logname	Affiche le nom d'utilisateur par défaut Ubuntu <div><div>logname --help :[get help or exit]</div><div>logname --version : [affiche la version]</div></div>	
id	Affiche id du groupe et l'utilisateur réel ou effective ça dépend <div><div><div>id -a [ignorer autres (un seul)]</div><div>id -z[affiche le contexte de sécurité] (--content)</div></div><div><div>id -g [affiche id grou] (-group)</div><div>id -G [affiche id des groups] (-groups)</div></div><div><div>id -n [nom à la place des num] or id --name</div><div></div></div><div><div>id -r [real ID] id --real</div><div>id -u [effective ID] id --user</div></div></div>	
tty	Teletypewriter Linux représente tout dans un système fichier même le terminal Udev aussi tty ⇒ affiche /dev/pts/o a device file -s --silent --quiet après le tty retourne rien just exit status	
cat hello.c	Affiche le contenu du file hello.c <div><div><div>cat -A --show-all équivalent a -vET</div><div>cat -b number nonempty (non vide)</div></div><div><div>cat -e équivalent à -vE</div><div>cat -E affiche \$ a la fin de chaque ligne</div></div><div><div>cat -n affiche que les nombre</div><div>cat -s supprime les lignes répétées</div></div></div> Il reste cat avec -t -T -u -v	
wc hello.c	Affiche une nouvelle ligne mot byte comptée pour le file entré ça dépend ce qui vient après wc hello.c -c:nombre de bytes -m caracteres -l lignes -w mots ...	
who	Montre tt les users who logged on who + <div><div><div>-a same as --login</div><div>-b dernier system boot</div><div>-d dernier processus en panne</div></div><div><div>-H le nombre de colonnes headings</div><div>-ips hostnames</div><div>-l login system processes</div></div></div>	
df	Report file system disk space usage ex: df hello.c affiche 1ko	
echo \$HOME	Affiche une ligne de texte	
env	Exécuter le programme dans différent environnement -i empty -0 line==null -u remove variable -s séparer en plusieurs arguments	

