Gestion des fichiers 1

Partie1 :

1/ mkdir rep

2/ cd rep

3/ mkdir work play && touch fich1 fich2 fich3

4/ ls -a

5/ ls -l

6/ cp fich1 fich2 play/

7/ mv fich3 fichier3

8/ cp /etc/iproute2 ./

9/ mv iproute2 play/

10/ cd play && echo « Hello World » > fichier1

11/ ln -s fichier1 lienfichier1 (On aura un lien symbolique vers fichier 1 mais lorsque on suppriment fichier1 le lien ne sera pas fonctionnel)

12/ cat lienfichier1 (le contenu de fichier1 sera affiché)

13/ echo « Bonjour ISET » >> fichier1 (le contenu de echo est ajoutée a la fin du fichier1)

14/ mkdir other && cd other && touch something

15/ cd .. && rmdir other (le dossier n’est pas supprimé car il contient de fichiers)

16/ ls -a other

17/ rm -rf other

18/ cd play

Chemin Absolut : cd /home/kali/Desktop/LPIC/rep/work

Chemin relative: cd ../work

19/ rm -rf ../../rep

Partie1 :

1/ ls -i ~

2/mkdir reper && ls -la reper

On a deux liens : Un lien pour le répertoire lui-même et Un autre lien pour référence à lui même

3/ mkdir reper/sous && ls -la reper

On 3 liens : on a un autre lien pour le dossier sous

4/ touch exercice && ls -i (2633009 exercice)

5/ mv exercice fich (le numéro d’inode ne change pas)

6/ ln fich liendur && ls -i

7/ ls -s fich liensymb (le lien symbolique a son propre i-onde)

8/ mv fich ../ (le lien physique reste fonctionnel mais le lien symbolique n’est plus fonctionnel)

9/ rm ../fich (le lien physique conserve une copie du fichier mais le lien symbolique n’est plus fonctionnel)

Gestion des fichiers 2

Partie1 :

1/ mkdir divers

2/ touch divers/code.c divers/alpha1 divers/alphabet divers/CV divers/Topologie divers/tp12.c divers/d2b divers/header.h divers/Me\*My\*I divers/spell[14].tab divers/réseau divers/etoile divers/bus divers/anneau

3/ cd divers

4/

1. ls | grep ‘.c$’
2. ls | grep ‘^……$’
3. ls | grep ‘^.\{1,5\}$’
4. ls | grep ‘^[A-Z]’
5. ls | grep ‘^..$’
6. ls | grep -v ‘^[at]’
7. ls | grep ‘^[at]’
8. ls | grep ‘\.’
9. ls | grep ‘[0-9]’
10. ls | grep '.c$' && ls | grep '.h$' ou ls | grep '.c$\|.h$'
11. ls | grep ‘\\*’
12. ls | grep '\[[0-9][0-9]\]' ou ls | grep '\[[0-9]\{2\}\]'
13. ls | grep ‘!’
14. ls | grep -v ‘!’
15. ls | grep ‘[ !\]\[\\*]’

5/ mkdir sauvegarde && cp \*.c sauvegarde

6/ ls -ltR ~

7/ sudo find / -type f -name 'std.\*h'

8/ cp /usr/include/std.\*h ~/divers/

Les Filtres

Exercice1 :

1/ cat carnet\_adresse

sort -t : -k1,1n carnet\_adresse

2/ sed -i 's/:/#/g' carnet-adresse

3/ sed -i 'y/abcdef/ABCDEF/' carnet-adresse

4/ tail -n 2 carnet-adresse

5/ tail -c 10 carnet-adresse

6/ head -c 10 carnet-adresse

7/ tac carnet-adresse | tail -n +3

8/ head -n 2 carnet\_adresse

9/ cut -c 1-10 carnet-adresse

10/ cut -c 2-5 carnet-adresse

11/ cut -d : -f1,4 carnet\_adresse

12/ cut -d : -f 3- carnet\_adresse

Exercice1 :

1/ sort -t : -k1,1n carnet\_adresse

2/ sed -i 's,/:/g' client

3/ cut -c 1-10 client

4/ cut -c 2-5 client

5/ cut -d : -f1,4 client

6/ cut -d : -f 3- client

7/ grep "Tunis" client

8/ grep "/03/" client

9/ grep "78" client

Exercice3 :

1/ grep '^.t' dictionnaire

2/ grep '^[pP]' dictionnaire

3/ grep -v '^[pP]' dictionnaire

4/ grep 'r.c' dictionnaire

5/ grep -c 'e$' dictionnaire ou grep 'e$' dictionnaire | wc -l

6/ grep '[ens]$' dictionnaire

Exercice4 :

1/ ls | grep ‘^T’

2/ ls | grep ‘e$’

3/ ls | grep -v ‘ro’

4/ ls | grep ‘^[APS]’

5/ ls | grep ‘e$’ | wc -l

6/ ls | grep ‘^T’ | wc -l

7/ ls | grep ‘ro’ | wc -l

8/ ls | grep ‘[0-9]’

Exercice5 :

1/ find . -type d

2/ find . -type f

3/ find . -type f -name 's\*'

4/ find . -type f -name '\*.c'

5/ find . -type f -name '\*e\*'

6/ find . -type d -name 'e\*'

7/ find . -type f -name 'info' -user isetuser

8/ find . -type f -user iset-user -perm 664

9/ find . -type f -name 'gest\*' -exec rm -i {} \;

10/

find . -type f -name '\*.c' 2>/dev/null

find . -type f -name 'a\*' -o -name 'm\*'

find . -type f -name '\*[0-9]\*'

11/ find . -type f '^[es][a-zA-Z0-9]\*[1-3]$'

12/

find . -type f -mtime +30

find . -type f -mtime 30

find . -type f -mtime -30

13/ find . -type f -name '\*.c' -newer ./info

14/ find . -type f -mtime -12 -size +30k

15/ find . -type f -name '\*~' -mtime +30 -exec rm {} \;

16/ find /usr -type f -size +2000 -perm 755

17/ find . -type f -user kali -perm 666 | w

Gestion des utilisateurs et groupes 1

Partie1 :

1. Oui, min utilisateur est defini dans /etc/passwd car ce fichier contient les informations de base sur tous les utilisateurs du système.
2. Le répertoire de connexion de l'utilisateur root est /root
3. Le shell de l'utilisateur root est /bash
4. Les utilisateurs qui appartiennent au même groupe ont le même group ID (GID)

Partie2 :

1. Ce fichier ne peux pas etre modifier car je ne suis pas root et je n’ai pas le droit de supprimer ou modifier les fichiers de configurations
2. echo ‘Hello ISET’ > fichier1.txt

chmod 444 fichier.txt

1. echo "Bonjour ISET" > secret.txt

chmod 400 secret.txt

ce fichier ne sera pas lisible par les mombres de mon group

1. mkdir TI23

chmod 773 TI23

Partie3 :

1. chmod o-rwx /home/kali
2. mkdir /tmp/groupe

chown :RSI21 /tmp/groupe

chmod 750 /tmp/groupe

touch /tmp/groupe/DepInfo

chown :RSI21 /tmp/groupe/DepInfo

chmod 760 /tmp/groupe/DepInfo

1. Oui, les membres du groupe peuvent supprimer le fichier DepInfo car il ont les droits de lecture et modification

Partie4 :

1. Fichier dans un repertoire :   
   -repertoire 7 et fichier 7

Copier fichier :

-repertoire 7 et fichier 4

Renomer un fichier :

-repertoire 3

Lister :

-fichier 4

Créer repertoire :

-repertoire parent 7

Supprimer une repertoire avec contenu :

-repertoire parent 7

1. umask 126

Le droit d’acces d’un repertoire sera 651

1. umask 022

Le droite d’acces d’un fichier sera 644

Partie5 :

1. mkdir rep1

chmod 700 rep1

1. cd rep1

touch fich1.txt

cat fich1.txt

1. chmod -w fich1.txt
2. mkdir ~/rep2
3. cp fich1.txt ~/rep2/

mv /home/kali/rep1/fich1.txt /home/kali/rep1/fich2.txt

1. cat ~/rep2/fich1.txt

ls -l ~/rep2/fich1.txt ./fich2.txt

1. rm -rf rep1 rep2 fich1.txt fich2.txt

En utilisant rm -rf les dossier sont forcement supprimer et c’est possible puisque on a les droits nécessaires mais pour la suppression des fichier on n’a pas le droit d’écriture donc on ne peux pas les supprimer

Partie6 :

1. mkdir TP1 TP2 && touch f1 f2 f3 f4 f5 f6 f7
2. mv f1 f2 f3 TP1/ && mv f4 f5 f6 TP2/
3. chmod u=rw,g=r,o= TP1

chmod 530 TP2

chmod 644 f1

chmod 640 f2

chmod 544 f3

chmod 740 f4

chmod 644 f5

chmod 430 f6

chmod 664 f7

1. On ne peux pas le droit de deplacer le fichier f7 vers TP2 car on a le droit 530 qui manque le droit d’ecriture pour l’utilisateur   
   pour deplacer on peux avoir 730

Gestion des processus

1/

1. who; pwd; ps
2. (who; pwd; ps) > temp
3. ls -i >> temp
4. ca 2> erreurs
5. ls -i > 1

whoami > 2

ls -d \*/ > 3

cat 1 2 3 > f1

1. ls -i >> f2

whoami >> f2

ls -d \*/ >> f2

2/

1. ps

La commande ps affiche les processus actifs dans le terminal actuel

1. ps -l

L'output de la commande ps -l affiche des informations détaillées sur les processus en cours d'exécution pour l'utilisateur actuel.

1. ps -e

Affiche tous les processus en cours sur le système, y compris les processus système.

ps -f

Affiche les processus avec plus de détails concernant l'utilisateur et le processus parent

ps -ef

Combine -e et -f , affichant tous les processus du système avec leurs details

ps -u  
d’afficher les processus d’un utilisateur

1. Avant-plan : Bloque le terminal jusqu'à la fin de la commande.

Arrière-plan : Permet de continuer à utiliser le terminal pendant que la commande s'exécute.

1. Lorsque on lance la commande sleep en avant plan le terminal reste bloque jusqu’au fil du temps puis se lance est le processus n’est pas afficher dans la liste des processus. Par contre en arrière-plan on peut lancer des commandes et le processus sleep existe dans la liste des processus.
2. Sleep 1000 &

ps -l (PPID 1472 ,PID 44200)

1. Lorsque on lance la commande kill -19 le processus sera suspendu avec l’état T (Stopped) du processus
2. Lorsque on lance kill -9 le processus sera totalement tuée
3. Lorsque on tue le terminal on ne peut plus exécuter de commandes

3/

1. Cette liste de commandes consiste à créer plusieurs processus en arrière-plan puis en utilisant fg %2 on rond le processus en avant-plan et ctrl+z bascule vers arrière-plan avec le même rôle que bg %2 et enfin kill pour tuer le processus
2. fg bascule le processus en avant-plan et bg bascule vers arrière-plan
3. job -l affiche uniquement les processus lancés à partir du terminal actuel et ps montre une liste plus complète des processus actifs dans le système ou par utilisateur.
4. **Arrière-plan (bg)**: Processus s'exécutant sans bloquer le terminal.

**Arrière-plan (fg)** : Processus s'exécutant sans bloquer le terminal.

**Suspendu (kill -19)** : Processus temporairement arrêté.

**Terminé (kill -9)** : Processus terminé.

4/

1. Pstree , top
2. pstree donne l’arboresance des processus et top affiche les processus actifs
3. time who calcule le temps d’exécution de la commande who
4. real :Temps total écoulé

User : Temps passé par le processeur à exécuter les instructions de la commande en mode utilisateur

Sys: Temps passé par le processeur à exécuter les opérations en mode noyau

5/

1. ps -e > ps.txt
2. ps -e | wc -l
3. ps -e | sort -k4

Gestion des utilisateurs et groupes 2

Exercice 1 :

sudo groupadd mi3a

sudo groupadd mi3b

sudo groupadd mi3c

sudo groupadd etudiant

Exercice 2 :

sudo useradd -g mi3a -G etudiant hamdi

sudo useradd -g mi3a -G etudiant moez

sudo useradd -g mi3b walid

sudo useradd -g mi3c -G etudiant sami

sudo useradd -g etudiant -d /home/mhamdi -u 1000 tarik

Exercice 3 :

1/ sudo usermod -aG etudiant walid

2/ sudo usermod -ag mi3c -aG etudiant -d /home/tarik tarik

Exercice 4 :

Partie1:

1/ sudo cp ~/linux.pdf /home/hamdi/

sudo chown hamdi:mi3a /home/hamdi/linux.pdf

2/ -sudo chown :mi3a /home/moez/bd.pdf

sudo chmod 640 /home/moez/bd.pdf

- chmod 600 /home/moez/prog.java

- sudo chown :etudiant /home/moez/docs

sudo chmod 750 /home/moez/docs

Partie2 :

-mkdir essai\_droi

-chmod 751 essai\_droit

-chmod 521 essai\_droit

-chmod 214 essai\_droit

-chmod 150 essai\_droit

Partie3 :

-unmask 026

Exercice 5 :

1/ sudo userdel tarik

2/ sudo userdel -r walid

Scripts Shell

Exercice1 :

ISET

SILIANA

./myfirstscript.sh

a1

5

a1 b1 c3 d4 e5

c3

4

ghi

abc def ghi

$ABCD

aaaBCD

aaa- -B++cccD

Aujourd'hui, nous sommes les Mon Nov 25 04:14:51 PM EST 2024

Exercice2 :

echo "voici les fichiers du répertoire courant : "

ls -la

if [ -f $1 ]; then

cat $1

else

echo "Fichier introuvable"

fi

Exercice3 :

#!/bin/bash

sum=0

product=1

count=0

average=0

if [ "$#" -ne 0 ]; then

for value in "$@"; do

sum=$((sum + value))

product=$((product \* value))

count=$((count + 1))

done

average=$((sum / count))

else

echo "no values"

fi

echo "Sum: $sum"

echo "Product: $product"

echo "Average: $average"

Exercice4 :

#!/bin/bash

nom=$1

n=$2

filename="${nom}\_${n}"

# Demander confirmation à l'utilisateur

echo "Voulez-vous vraiment créer le fichier '$filename' ? (y/n)"

read confirmation

# Si l'utilisateur confirme, créer le fichier

if [ "$confirmation" == "y" ] || [ "$confirmation" == "Y" ]; then

touch "$filename"

echo "Le fichier '$filename' a été créé."

else

echo "Création du fichier annulée."

fi