

TP LINUX

II) Découverte du système de fichiers Linux via KDE

1/

Click gauche :

- Bureau : affiche le contenu du bureau
- Dossier : affiche le contenu du dossier
- Fichier : ouvre le fichier

Click centre :

- Bureau : affiche le bureau dans un autre onglet
- Dossier : affiche le dossier dans un autre onglet
- Fichier : rien

Click droit :

- Bureau : affiche les paramètres du bureau
- Dossier : affiche les paramètres du dossier
- Fichier : affiche les paramètres du fichier

2/ Le dossier de mon répertoire principal est : /filer/etudiants/g2/

3/ `ls -a` Fait :

```
.          .gnome2_private          PRIV
..         .gnupg              PUB
.bash_history  .ICEauthority              Public
.bash_profile  Images                    public_html
.bashrc        IMPORTANT_A_LIRE_mon_compte_a_luminy.html
.recently-used
Bureau        Library                    signature.txt
.cache        .local                      .snapshot
.config       Modèles                     .ssh
.dbus         .mozilla                    Téléchargements
.dmrc         .mozilla-thunderbird       .thunderbird
Documents     MozillaWindows2000         Vidéos
DOSSIERS_WINDOWS Musique                    .Xauthority
.gconf        .nfs0000000002bab8ea00000001 .xsession-
errors
.gnome2       .oracle_jre_usage
```

III) Observation des processus via KDE

1/ ouverture de l'application Table des processus qui affiche les processus actifs et surveille le système.

2/

- La colonne Utilisateur décrit à quel utilisateur appartient le processus.
- Les colonnes Titre de la fenêtre et Nom permettent de savoir à quelle application correspond un processus.
- La colonne PID permet de connaître l'identifiant PID de chaque processus (Clic droit sur l'en-tête des colonnes du tableau et on choisit « Afficher la colonne PID »).

3/ Pour afficher :

- L'arborescence de tous les processus, on sélectionne : Tous les processus, en arbre.
- Uniquement mes processus, on sélectionne : Processus utilisateurs.

4/ Fait

5/

SIGTERM : arrête un processus et le ferme (ne peut pas fermer un terminal).

SIGKILL : tue un processus, arrêt brutal et fermeture (peut fermer un terminal).

IV) Premier contact avec l'interpréteur de commande

1/ Dans le menu, recherche de Terminal.

2/ la commande `pwd` montre que le terminal s'ouvre sur le repertoire principal de l'utilisateur : `/filer/etudiants/g2/b14015170` pour l'utilisateur `b14015170`.

3/

`whoami` : affiche le nom de l'utilisateur

`ps -f -U user` : liste des processus que l'utilisateur (spécifié après `-U`) utilise, sous forme de liste (avec l'option `-f`).

4/ Fait.

5/ `ls` : liste le contenu d'un repertoire courant.

6/

L'option `-a` affiche tous les fichiers et dossiers, même s'ils sont cachés.

L'option `-al` regroupe les effets de l'option `-a` et de l'option `-l`.

L'option `--color=auto` colore les fichiers selon leurs types.

7/ pour créer un alias il faut utiliser la commande `alias`.

8/ `alias llc='ls -la --color=auto'`

9/ Fait

10/ `exit` : ferme le shell.

11/ `llc` n'existe plus car l'alias n'est plus en mémoire après la fermeture du shell.

V) Se déplacer dans une arborescence

1/ avec `mkdir`

3/ `cd Initiation_Informatique`

4/ `cd ..`

5/ `cd Initiation_Informatique/Linux/TP2`

VI) Création de fichiers

1/ touch bonjour

2/ cat bonjour

3/ cat > bonjour
Hello World
Ctrl+D

4/ cat bonjour

5/ cat>>bonjour
How do you do?

6/ cat bonjour

7/ cp bonjour salut
diff bonjour salut

8/

> : ce symbole créé ou écrase le contenu existant.

>> : ajoute le texte à la fin du contenu existant.

VII) Notion de filtre de fichiers

1/ ls *[a,m][a,m]*

2/ Fait

3/

- dont le nom commence par a :

ls [a]*

Aucun fichier ou dossier de ce type

- dont le nom commence par une voyelle :

ls [a,e,i,o,u]*

echo ip openvt ulockmgr_server uname unicode_start

egrep open udevadm umount uncompress

- dont le nom contient sh :

ls *[s][h]*

bash bsd-csh csh dash rbash sh sh.distrib

- de 3 lettres qui se terminent par sh

ls ?[s][h]

csh

4/ ls ???[a,b,c,d]*

badblocks discover-pkginstall getcap iptables-multi on_ac_power
shadowconfig

blockdev e2label hdparm iptables-restore pccardctl sysctl

discover findfs ipmaddr iptables-save rtacct update

discover-modprobe fstab-decode iptables lspcmcia setcap

VIII) Nom de fichiers et caractères spéciaux

1/ touch a*b
 acb
 addb

ls a*b affiche tous les fichiers qui commence par 'a' suivi d'une chaîne de caractères et qui finissent par b

2/ le \ et les guillemets protègent permettent d'échapper le caractère * et donc d'empêcher son interprétation par le Shell.

3/ mkdir cou«\»cou
 mkdir cou\\cou
 mkdir «cou\cou»

4/ touch mon\ fichier

IX) Compression/Décompression de fichiers

1/ mkdir TP3

2/ cd TP3

3/gzip [-cdfLNvr] [-chiffre de vitesse] {element}
 gunzip [-cdfLNvr] {element}

4/ Fait.

5/ Poids de départ : 293ko.
 Poids compressé 284ko.
 Poids gzip : 284ko.

6/ Fait

7/ Archiver le répertoire TP3 au format TAR : tar -cf TP3.tar TP3
 Lister le contenu de l'archive TP3.tar : tar -tvf TP3.tar
 Restituer le répertoire TP3 : tar -xf TP3.tar

X) Les droits d'accès

1/
 mkdir -m 777 RepertoireTestDroit
 cd RepertoireTestDroit
 cat > monFichierPerso
 Mon Secret
 Ctrl + D
 chmod 666 monFichierPerso
 mkdir Sous\ Repertoire\ 1
 cat > Sous\ Repertoire\ 1/fic
 Hello World
 Ctrl + D

```

mkdir Sous\ Repertoire\ 2
cat > Sous\ Repertoire\ 2/fic
    Salut l'ami
    Ctrl + D
chmod 666 Sous\ Repertoire\ 1/fic
chmod 666 Sous\ Repertoire\ 1
chmod 666 Sous\ Repertoire\ 2/fic
chmod 111 Sous\ Repertoire\ 2

2/
ls ./Sous Repertoire 1 :                oui
ls -l ./Sous Repertoire 1 :                non
cd ./Sous Repertoire 1 :                non
cat ./Sous Repertoire 1/fic :                non
touch ./Sous Repertoire 1/fic2 :    non
rm ./Sous Repertoire 1/fic :                non
ls ./Sous repertoire 2 :                non
cd ./Sous repertoire 2 :                oui
cat ./Sous Repertoire 2/fic :    oui
touch ./Sous Repertoire 2/fic2 :    non
rm ./Sous Repertoire 2/fic :                non
echo "Comment vas tu" >> ./Sous Repertoire

```

2/fic : oui
 les permissions font semblent fonctionner correctement.

3/ Ce n'est pas la meilleure solution car personne, même lui, ne pourra y accéder. Il serait mieux de faire rwx --- ---.

XI) Les alias permanents

dans le fichier bashrc, on y écrit `alias llc='ls -al -color=auto'`.

XII) Gestion des processus

```

1/
PID  TTY  TIME      CMD
2703 pts/1 00:00:00  bash
2728 pts/1 00:00:00  emacs
2760 pts/1 00:00:00  ps

```

2/ La commande `ps -l` permet d'afficher les caractéristiques de chaque processus (dont le PID et le PPID) sous forme de liste.

Le processus père au processus emacs est le processus bash.

3/ La commande `ps -Af` affiche tous les processus exécuté sur l'ordinateur sous forme de liste.

On tue le processus correspondant au premier terminal avec `kill -9 2728`
 Le premier terminal et emacs s'arrêtent..

4/ Après la suppression du terminal, le processus emacs reste actif et est adopté par le PID 1 qui est le processus init. On en déduit que si un processus n'a plus de processus père alors le système lui attribue le PID 1.

XIII) Redirections

1/ `ls -l | cat > toto`

2/ `ps -A | cat >> toto`

3/ `ps -Af | cat > tata`

4/

`wc -w bonjour`

`wc -m bonjour`

`wc -l tata`

`sort -n tata`

`cat toto | head -n 10`

`cat toto | tail -n 5`

5/ `sort -d -b tata.txt | head -n 23 | tail -n +4 > titi.txt`

XIV) Recherche d'une chaîne de caractères dans des fichiers textes

1/

`-c` : donne seulement le nombre de lignes trouvées contenant le motif.

`-L` : donne seulement le nom des fichiers dans lesquels le motif a été trouvé.

`-v` : donne les lignes où le motif n'a pas été trouvé.

`-i` : ne tient pas compte de la casse (ne différencie pas les majuscules des minuscules).

`-w` : impose que le motif corresponde à un mot entier sur une ligne du fichier.

2/ les expressions dans le `grep` agissent sur le caractère qui précède et non pas sur leur propre emplacement, de plus on s'intéresse au nombre de fois où l'on rencontre la lettre précédente.

3/

`grep "^o" tata` : toutes les lignes qui ont un o en 2eme lettre

`grep -ni bash tata` : numéro de la ligne et ligne contenant bash en majuscule ou minuscule

`grep -ni "td$" tata` : numéro de la ligne et ligne finissant par td en majuscule ou minuscule

4/ `grep "^r.*[0-9]$" tata.txt`

5/ `ps -aux | grep "b140151"`

6/ `ps -aux | grep "b140151" | wc -l`

XV) Recherche de fichiers dans une arborescence

1/ locate cherche un fichier dans une base de donnée alors que find recherche directement dans l'arborescence. Find est donc moins rapide mais il a l'avantage d'être à jour en temps réel contrairement à locate.

2/ find simulé : `ls -R | grep "Algo"`

XVI) Les variables d'environnement

1/

HOME : indique le répertoire principale de l'utilisateur

USER : indique l'identifiant de l'utilisateur

DISPLAY : indique aux application où afficher l'interface graphique utilisateur

TERM : nom de l'émulateur du terminal utilisé par UNIX

HOSTNAME : nom de domaine du système

SHELL : le shell courant

PATH : list de répertoires utiles au système

2/ `touch fichier $USER` : créer un fichier « fichier » dans le répertoire de l'utilisateur.

`touch "fichier $USER"` : créer un fichier « « fichier » et un fichier « b14015170 » dans le répertoire actuel.

`*touch `fichier $USER`` : commande introuvable

« et ' n'ont pas d'effet sur les variables d'environnement.

3/

`TEMP2=«ma première variable»`

`TEMP3=$PATH`

`PATH= «salut»`

`PATH=TEMP3`

Is utilise la variable d'environnement PATH, il est donc déconseillé de la modifier.

XVIII) Les arguments des scripts

1/

```
\#!/bin/bash
```

```
echo "Bonjour $USER"
```

```
echo "les quatres premiers arguments : $1, , $2 , $3 , $4"
```

```
echo "nombre total d'arguments : $#"
```

2/

```
\#!/bin/bash
```

```
ls $1
```

3/

```
\#!/bin/bash
```

```
mkdir $1
```

```
touch $1/$2
```

```
chmod a-wx $1/$2
```

XIX) Les conditionnelles dans les scripts

2/

```
\#!/bin/bash
if [ -e $1 ]
then
    cat $1
else
    echo "le fichier specifié n'existe pas"
fi
```

XX) Les boucles dans les scripts

2/

```
\#!/bin/bash
for fichier in ./*
do
    if [ -d $fichier ]
    then
        echo " $fichier est un repertoire"
    else
        echo " $fichier est un fichier"
    fi
done
```

3/

```
ls -R avec ls echo et cd :
\#!/bin/bash
cd /
for fichier in ./*
do
    if [ -d $fichier ]
    then
        echo `ls`
        cd $fichier
    else
    fi
done
```