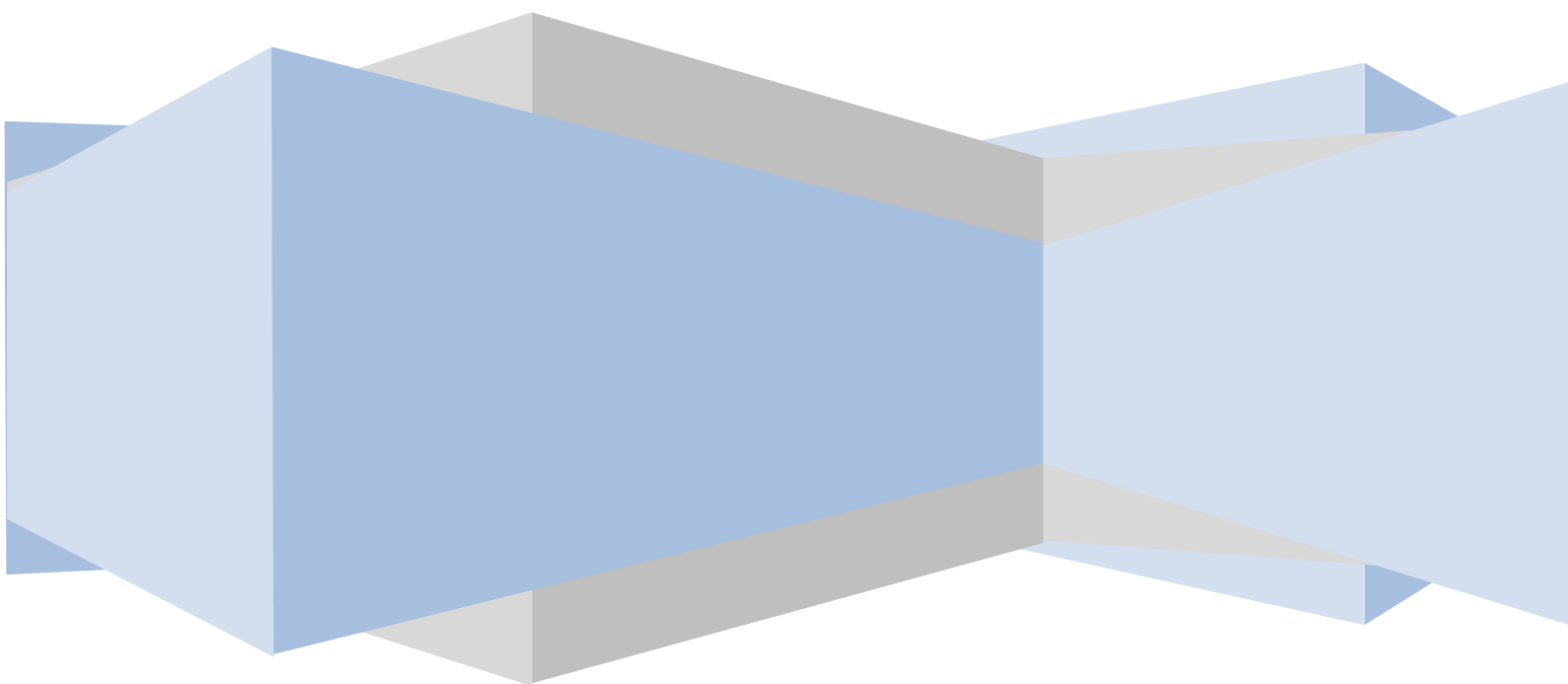


EFREI

DE Unix pour l'utilisateur L2

6 Janvier 2012



1 Fondamentaux & manipulation de fichiers

1. Quels sont les 4 concepts fondamentaux d'Unix ? Décrivez en une phrase chacun d'entre eux.

Tout est fichier

2

IPC

Processus

Contrôle d'accès

2. Qu'est-ce que l'UID et le GID ?

UID -> User ID

GID -> Group ID

1

3. Comment obtenir l'UID et le GID de l'utilisateur `bernardi` ?

`id bernardi`

1

4. Donnez la différence entre `ls /directory` et `ls directory` ?

`/directory` -> chemin absolu

`directory` -> chemin relatif

1

5. Dans la commande `ls -al`, à quoi correspondent les options `a` et `l` ?

`a` -> all

`l` -> long listing

1

6. En utilisant uniquement des chemins relatifs, créez l'arborescence suivante (pas de cd !). Les éléments appelés « dir » sont des répertoires et les éléments appelés « file » sont des fichiers.

```
.  
|-- dir1  
|   |-- file  
|   `-- file12  
|-- dir2  
|   |-- file21  
|   `-- file22  
`-- dir3  
    `-- rep4
```

```
mkdir -p dir1 dir2 dir3/dir4  
touch dir1/file dir1/file12 dir2/file21 dir2/file22 dir3/rep4
```

1

7. Comment copier le fichier /tmp/fichier dans votre home ?

```
cp /tmp/fichier ~
```

1

8. Quelle touche permet de compléter automatiquement le nom du fichier que vous êtes en train de saisir à la ligne de commande (autocomplétion)?

```
tab
```

1

2 Filtres et redirections

1. Qu'est-ce qu'un filtre ? Donnez un exemple.

```
cat /etc/passwd | grep root
```

1

2. Que fait la commande `cat /etc/passwd | cut -d ':' -f 1?`
(voir fichier en annexe)

Affiche uniquement les logins

1

3. Qu'est-ce qu'une redirection ? Donnez un exemple.

```
echo plop > fichier  
prend la sortie de la commande echo et la copie dans le fichier fichier
```

1

4. Comment afficher uniquement les lignes du fichier `/etc/passwd` comportant le mot `root` ?

```
cat /etc/passwd | grep root
```

1

5. Comment compter le nombre de caractères renvoyés par la commande précédente ?

```
cat /etc/passwd | grep root | wc -c
```

1

6. Comment rediriger le flux d'erreur de la commande `find /etc -type d` dans un fichier nommé `erreurs` ?

```
Find /etc -type d 2> erreurs
```

1

7. Comment faire disparaître la sortie standard d'une commande de l'écran ?

commande > /dev/null

1

8. Quelle est la différence entre `ls >> fichier` et `ls > fichier` ?

Avec > crée un nouveau fichier, écrase le contenu si fichier existant
Avec >> ajoute les nouvelles données à la fin du fichier

1

3 Gestion des permissions

1. Comment sont décrites les permissions d'un fichier ? Donnez un exemple.

Read (r) write (w) execute (x)
User (u) group (g) other (o)

1

2. Que fait la commande `chmod g-w fichier` ?

Supprime droit d'écriture du groupe pour le fichier nommé fichier

1

3. Comment faire pour rendre le script `killQuake.sh` exécutable ?

`chmod u+x killQuake.sh`

1

4 Processus

1. Comment afficher la liste des processus en cours d'exécution ?

Ps ou top par exemple

1

2. Comment demander à un processus de se terminer proprement avec la commande `kill` ?

`Kill <PID du processus>`

1

3. Dans le cas où la commande précédente ne fonctionne pas, comment forcer l'arrêt du processus ?

kill -9 <pid du processus>

1

4. Que fait la commande `gedit &` ?

Lance la commande `gedit` en tâche de fond

1

5. Que fait la commande `lsdf` ?

Affiche la liste des fichiers (au sens large) ouverts par les processus du système

1

5 Système

1. Qu'affiche la commande `mount` ? Détaillez la réponse.

La liste des montages du système.

Les montages peuvent être des périphériques de masse locaux (ie disque dur, clés USB), des systèmes de fichier distant (ie NFS), des systèmes de fichiers virtuels (ie `/dev` ou `/proc`)

1

2. Qu'affiche la commande `uname` ?

Le type d'OS ie Linux

1

3. Comment connaitre depuis combien de temps votre système est sous tension ?

`uptime`

1

4. Que permet de faire la commande `ifconfig` ?

Afficher les différentes interfaces réseau du système

1

5. Comment afficher l'espace disque occupé par chacun des répertoires de votre « home directory » ?

```
du -h --max-depth=1
```

1

6 Divers & Utilitaires

1. Dans `vi`, que fait la commande `:%s/coin/pouet/g` ?

Remplace coin par pouet dans tous le fichier

1

2. Comment stocker la date du jour dans la variable MAINTENANT ?

```
MAINTENANT=$(date)
```

1

3. Qu'est-ce qu'une commande interactive ? Donnez un exemple.

Une commande qui réagit à des appuis de touche de la part de l'utilisateur après son lancement :
mutt, nano, top, vi...

1

4. A l'aide de `ssh`, comment déporter l'affichage sur votre bureau du Iceweasel de votre voisin de TP ? Sa machine se nomme `chouffe`.

```
ssh -X chouffe
```

1

5. Que permet de faire la commande
`scp login@machine1:monfichier.txt login@machine2:?`

Copier le fichier monfichier.txt de la machine1 vers la machine2 dans le home de l'utilisateur
login

1

6. A quoi sert la variable système `$PATH` ?

Elle contient l'ensemble des chemins contenant des exécutables que l'on souhaite pouvoir exécuter n'importe où sur le système

1

7. Comment définir l'alias d'une commande ?

```
alias <nom de l'alias>=<commande>
```

1

8. Comment réaliser une archive compressée du répertoire `a_archiver/` avec `tar` ?

```
tar cvfz archive.tgz a_archiver/
```

1

9. Comment décompresser l'archive réalisée à la question précédente ?

```
tar xvfz archive.tgz
```

1

10. Comment rechercher tous les fichiers se terminant par « `sh` » dans le répertoire courant et tous ses sous-répertoires ?

```
find ./ -name « *.sh »
```

1

7 Scripts

1. Ecrire un script shell qui décrit sur sa sortie standard les messages suivants :

```
Mon nom est xxx  
Je suis appelé avec yyy arguments  
qui sont 111 222 333 444
```

(`xxx` sera remplacé par le nom sous lequel ce script aura été invoqué, `yyy` par le nombre d'arguments et `111`, `222`, etc. par les arguments en question).

5

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

```
echo "Mon nom est $0"
```

```
echo "Je suis appelé avec $# arguments"
```

```
echo "qui sont: $*"
```

2. Réaliser un script qui permet de réaliser la multiplication d'un nombre quelconque d'entiers passés en paramètre. Exemple : `./multiply.sh 1 2 3` devra afficher à l'écran $1*2*3=6$, de même `./multiply.sh 3 10 5 8` devra afficher à l'écran $3*10*5*8=1200$

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

10

```
RESULTAT=1
```

```
for i in $*; do  
  RESULTAT=$((i*RESULTAT))  
done
```

```
for i in $*; do  
  if [ -e $AFFICHAGE ]; then  
    AFFICHAGE=$i  
  else  
    AFFICHAGE=$AFFICHAGE*$i  
  fi  
done
```

```
echo $AFFICHAGE=$RESULTAT
```

3. Réaliser un script qui affiche le nom de chaque fichier du répertoire dans lequel il se trouve, en le faisant précéder d'un numéro d'ordre, comme dans l'exemple suivant :

```
vinzelle@pommard:~/scripts$ bash list.sh  
1) list.sh  
2) break.sh  
3) carres.sh  
4) case.sh  
5) compteur.sh  
6) fonction.sh
```

```
#!/bin/bash  
  
numero=1  
  
for file in *  
do  
    echo "$numero) $file"  
    numero=$((numero+1))  
done
```

5