

1 Quelques questions de cours

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

Un système Unix repose sur quatre concepts : celui de fichier, de processus, d'IPC et de contrôle d'accès.

- Tout est fichier: tout est représenté sous forme de fichier (documents, matériels, ...)

- Processus: Un processus est l'image en mémoire d'un programme lors de son exécution. Chaque processus dispose de son espace mémoire, et peut éventuellement contenir un ou plusieurs threads (processus léger).

- Communication interprocessus (IPC): La communication interprocessus sert, d'une part à échanger des données entre les processus, et d'autre part à synchroniser les processus.

- Contrôle d'accès: les droits d'accès (lecture, écriture et exécution) sont appliqués à trois entités :
· Le propriétaire du fichier · Le groupe auquel appartient le fichier · Les autres utilisateurs

2. Quelle est la commande qui permet d'obtenir l'aide en ligne de (presque) toutes les commandes Unix? Donnez un exemple.

La commande est 'man'

Exemple 'man ls' l'aide de la commande 'ls' ou 'man sex' pour de l'aide sur...

3. Un utilisateur est identifié par deux identifiants numériques. Quels sont-ils?

UID (identifiant de l'utilisateur) et le GID (identifiant de groupe)

" Un utilisateur est décrit par son UID (Unique Identifier) et son GID (Group Identifier).

L'UID lui est propre, alors qu'il partage son GID avec les autres membres du groupe auquel il appartient."

4. Qu'est-ce que le répertoire d'accueil ou "home" ?

Le répertoire HOME est le répertoire personnel de l'utilisateur.

2 Manipulation des fichiers et répertoire

1. Créez un répertoire `rep1` contenant deux fichiers `fich11` et `fich12` et deux répertoires `rep2` et `rep3`, contenant chacun deux fichiers, respectivement `fich21` et `fich22`, `fich31` et `fich32`, sans utiliser la commande `cd`. On suppose que le répertoire courant est votre répertoire d'accueil.

3 Redirections, pipes et filtres

```
mkdir rep1
touch rep1/fich11
touch rep1/fich12

mkdir rep1/rep2
touch rep1/rep2/fich21
touch rep1/rep2/fich22

mkdir rep1/rep3
touch rep1/rep3/fich31
touch rep1/rep3/fich32
```

2. Comment supprimer le répertoire `rep1` avec une seule commande ? Indiquez cette commande et ses paramètres.

```
on fait 'rm -rf rep1'
rm pour supprimer
-r pour récursif / -f pour forcer a le faire (car le repertoire n'est pas vide)
```

3. Comment créer un fichier nommé `fich` avec la commande `cat`? Une fois ce fichier créé, rendez-le lisible et modifiable uniquement par vous-même, avec la méthode symbolique, et non octale. Indiquez les commandes utilisées.

```
cat > fich //pour creer le fichier
chmod rw -- fich //changer le droit d'accès User / Group / Other
```

4. Même question que précédemment mais le groupe auquel vous appartenez a maintenant le droit de lire le fichier `fich`.

```
chmod u=rw, g=r fich
```

5. Quelle commande vous permettra de lister en récursif tous les fichiers de votre `home` y compris les fichiers cachés ?

```
ls -aR
// -a pour ALL
// -R pour recursif
```

3 Redirections, pipes et filtres

1. Vous souhaitez afficher l'ensemble des lignes du fichier `fich1` contenant la lettre 'p'. Comment allez-vous réaliser ce filtre en utilisant les commandes `cat` et `grep` ?

```
cat fich1 | grep 'p'  
//cat fich1 l'affiche  
// grep 'p' permet de choisir quelle ligne afficher
```

2. Proposez une amélioration de votre réponse à la question 1 qui permet en plus de compter le nombre de ligne contenant la lettre 'p'.

```
cat fich1 | grep 'p' | wc -l  
// wc permet de compter  
// -l compte les lignes
```

3. Vous souhaitez que les erreurs de la commande traitement soient redirigées vers un fichier texte. Quel opérateur allez-vous employer ? Indiquez la ligne de commande redirigeant les messages d'erreur vers le fichier traitement.log.

```
traitement 2> traitement.log  
// 1> permet de stocker la sortie normale  
// 2> permet de stocker les erreurs
```

4. Finalement, vous souhaitez ajouter le mot TERMINE à la fin de votre fichier traitement.log. Comment allez-vous procéder ?

```
echo 'TERMINE' >> traitement.log  
// > stock dans un fichier en l'effaçant avant  
// >> stock dans un fichier en le mettant a la fin
```

4 Variable d'environnement, les alias et le prompt

1. Que contiennent les variables \$HOME et \$PATH?

```
$HOME Retourne le chemin d'accès vers le répertoire de l'utilisateur courant.  
$PATH Retourne une liste de chemins d'accès vers des répertoires contenant les  
aexécutables, séparés par des points-virgules.
```

2. Donnez un exemple d'application de la commande which. *localise une commande*

```
La commande which permet de savoir quel est le fichier exécuté pour une commande donnée.  
vinzelle@pommard:~$ /usr/bin/who  
vinzelle pts/0 2008-05-17 13:31 (choam.etudiants.efrei.fr)  
vinzelle@pommard:~$ which who  
/usr/bin/who
```

3. Affecter une chaîne de caractère vide à la variable \$PATH. Lancer la commande ls. Que se passe-t-il ? Pourquoi ?

```
PATH=""  
ls ne marche pas car le systeme ne trouve pas le programme  
en effet le programme cherche 'ls' dans le dossier '' qui n'existe pas
```

4. Sur un terminal, quelles commandes faut-il taper pour affecter la valeur 3 à la variable test puis afficher son contenu ?

```
test=3  
echo $test
```

5. Comment afficher l'ensemble des alias définis sur votre machine ?

```
alias
```

6. Définissez un alias dir qui réalise la même chose que la commande ls -ltr.

```
alias dir='ls -ltr'
```

5 L'éditeur de texte vi

1. Combien de modes existe-t-il dans vi ? Décrire rapidement chacun d'eux

- mode insertion: écrire du texte
- mode remplacement
- mode commande: modifier les commandes de VI

2. Comment entrer dans le mode de remplacement depuis le mode de commande ? Comment en sortir ?

```
entrer mode remplacement : touche R  
sortir du mode : touche ECHAP
```

3. Quel est l'intérêt d'utiliser vi par rapport à Wordpad pour vos fichiers script (imaginez que vous programmez un logiciel C++ qui nécessite la manipulation de fichiers) ?

```
*joker* je sais pas ^^          la coloration syntaxique c'est beau :p
```

6 Gestion des processus

1. Vous souhaitez arrêter le processus `deadproc` qui est totalement planté. Vous utiliserez pour cela la commande `kill`. Indiquez la commande nécessaire.

```
kill -9 deadproc  
// le -9 permet d'envoyer un SIGKILL qui tue le prog
```

2. Vous souhaitez savoir quel processus utilise en ce moment un fichier de votre système. Quelle est la commande à utiliser ?

```
jobs
```

3. Est-il possible de lancer une commande en tâche de fond ? Si oui, comment ?

```
il est possible de l'exécuter en arrière plan en ajoutant le caractère spécial &  
à la fin de la commande.
```

4. Qu'est ce qui identifie un processus de manière unique ? Comment afficher cette information ?

```
le numéro du processus (PID)  
on utilise la commande 'ps'
```

7 Commandes relatives au matériel

1. La commande `dmesg` permet d'afficher tous les messages du kernel d'un système Linux. Vous souhaitez lire le contenu d'une clef USB sur votre machine Linux. Comment procéder pour afficher tous les messages de `dmesg` concernant le bus USB ?

```
dmesg | grep USB
```

2. Vous venez de connecter votre clef USB. Comment afficher l'ensemble des périphériques USB connectés à votre système ?

```
lsusb
```

3. Vous voulez maintenant lister les systèmes de fichiers montés pour savoir si votre clef est bien présente et accessible sur votre système. Quelle est la commande à employer ?

mount

8 Compresser, décompresser et archiver

1. Quelle est la différence entre compression et archivage ?

la compression: l'idée c'est de réduire la taille d'un ou de plusieurs fichiers
l'archivage: la tu prends plusieurs fichiers et t'en fais un unique

2. Quel logiciel est utilisé sous Unix pour la compression ? Pour l'archivage ?

compression: gzip
archivage: tar

3. Expliquer en détail le fonctionnement de la commande: `tar czvf test.tar.gz test`

la commande archive et compresse le fichier 'test' en 'test.tar.gz'

4. Cette commande est-elle correcte ? Si oui que fait-elle ?
`tar xvf test2.tar.gz -C test2`

incorrecte

5. Quelle est la différence entre ces deux commandes :

`gzip fichier_a_compresser`
`gzip -9 fichier_a_compresser`

Compression optimisée d'un fichier (pour fichiers > 100 Ko) bzip2 -9 fichier

9 SSH

1. Expliquez le concept de clefs publiques / privées. Quelle est l'utilité des clefs ? Quelles différences entre les clefs publiques et privées ?

Le client établit donc une connexion avec le serveur. La clef publique est envoyée au client, qui doit générer une clef de session de 256 bits chiffrée et la renvoyer au serveur.

Le serveur déchiffre celle-ci grâce à la clef privée et renvoie une confirmation au client, établissant alors une communication chiffrée avec une clef de session commune.

2. Quelle est la syntaxe de base d'une connexion ssh ?

```
ssh utilisateur@adresse_serveur
```

3. Quelle commande utiliser pour un transfert de fichier client -> serveur ?

```
scp fichier user@adresseip:destination
```

Exemple :

```
vinzelle@choam:~$ scp temp.list vinzelles@pommard.etudiants.efrei.fr:
```

4. Même question pour un transfert de fichier serveur -> client.

```
scp user@adresseip:adressefichier destination  
Exemple:  
vinzelle@choam:~$ scp vinzelles@pommard.etudiants.efrei.fr:temp.list
```

10 Programmation Shell

1. Ecrivez un script qui permet de calculer la factorielle d'un nombre passé en paramètre.

Exemple d'utilisation

```
$ ./factorielle.sh 5  
!5 = 120
```

```
#!/bin/bash  
  
$b = 1  
$c = $1  
  
for i in $c  
do  
    $b = $b*$i  
done  
  
echo "$b"
```

2. Ecrivez un script `isalive` qui détecte si une machine répond à un ping.

Exemple d'utilisation

```
$ ./isalive d
d ne répond pas
$ ./isalive localhost
localhost répond
```

```
#!/bin/bash

nom $1

if [ ping $nom ]
then
    echo "$nom repond"
else
    echo "$nom ne repond pas"
fi
```

3. Ecrivez un script `taillesrep` qui affiche la taille en Ko de tous les sous-répertoires du répertoire passé en argument.

Exemple d'utilisation

```
$ ./taillesrep /usr/src/
/usr/src/ : 52 ko
/usr/src/toto : 48 ko
```

un truc avec la commande 'du'

4. Proposez une commande qui permette de trier la sortie de `taillesrep` selon la taille des répertoires dans l'ordre décroissant.

```
taillesrep | sort -n  
// sort = trie et -n numériquement
```