

---

## Script Shell : Correction TP1

---

### Exercice 1 : Introduction

1. Quel est le système installé? `uname -a`
2. Quelle est la distribution installée? `cat /etc/issue`
3. Quels sont les utilisateurs en cours du système? `who`
4. Ouvrir un nouveau terminal en utilisant un autre compte ? `su user`
5. Fermer le terminal ? `exit`
6. Tester la commande man pour obtenir de l'aide sur man. `man man`

### Exercice 2

#### Informations système

Décrire l'environnement dans lequel vous évoluez:

- nom de la machine

`hostname`

- détail des disques

`df`

- programmes en fonctionnement
  - les vôtres  
`ps`
  - ceux des autres utilisateurs  
`ps -A`

- configuration réseau

`ifconfig (ipconfig) ou /usr/sbin/ifconfig`

- place utilisée dans mon répertoire

`du -b -c $HOME`

- processus les plus gourmands en fonctionnement

`top`

### Exercice 3 : Contenu de répertoires et de fichiers ascii (texte)

1. Lister le contenu du répertoire /etc. `ls /etc`

2. Lister de manière détaillée le contenu du répertoire /etc. **ls -l /etc, ls -la /etc**  
Les fichiers nommés "passwd", "shadow", "group" et "gpasswd" sont-ils présents dans /etc? **Tous ces fichiers sont bien présents**
3. Lister le contenu du répertoire /dev. **ls /dev**
4. Lister le contenu du fichier /etc/passwd. **cat /etc/passwd** *Le fichier /etc/passwd contient toutes les informations relatives aux utilisateurs (login, mots de passe, ...). Seul le superutilisateur (root) doit pouvoir le modifier. Il faut donc modifier les droits de ce fichier de façon à ce qu'il soit en lecture seule pour les autres utilisateurs.*
5. Lister le contenu du fichier /etc/shadow. **cat /etc/shadow :fichier contenant les mots de passe** **Lecture refusée.** *Le fichier /etc/shadow est utilisé pour augmenter le niveau de sécurité des mots de passe.*
6. Afficher par ordre alphabétique les utilisateurs définis dans le fichier /etc/passwd. **cat /etc/passwd | sort**
7. Rechercher tous les fichiers du répertoire /etc contenant la chaîne de caractères "root". **grep root /etc/\***
8. Rechercher la localisation du fichier "stdio.h" dans le système de fichier de votre installation. **find / -name stdio.h**
9. A l'aide de la commande "od", illustrer la différence majeure entre les types de fichier ascii (texte) DOS, UNIX et Macintosh. Utiliser le fichier ascii Dos monfichier.dos et créer les fichiers ascii Unix "monfichier.unix" et ascii Macintosh "monfichier.mac" avec le même contenu au moyen d'un éditeur texte (Notepad par exemple) puis comparer (version Macintosh si l'éditeur le permet).

Quelle est la différence?

**od -x monfichier.dos**

**0000000 6e75 0a0d 6564 7875 0a0d 7274 696f 0d73**

**0000020 710a 6175 7274 0d65 630a 6e69 0d71 730a**

**0000040 7869 0a0d 6573 7470 0a0d 0a0d 0a0d 0a0d**

**0000060 0a0d 0a0d**

**0000064**

**od -c monfichier.dos**

**0000000 u n \r \n d e u x \r \n t r o i s \r**

**0000020 \n q u a t r e \r \n c i n q \r \n s**

**0000040 i x \r \n s e p t \r \n \r \n \r \n \r \n**

```
0000060 \r \n \r \n
```

```
0000064
```

```
od -x monfichier.unix
```

```
0000000 6e75 640a 7565 0a78 7274 696f 0a73 7571
```

```
0000020 7461 6572 630a 6e69 0a71 6973 0a78 6573
```

```
0000040 7470 0a0a 0a0a 0a0a
```

```
0000050
```

```
od -c monfichier.unix
```

```
0000000 u n \n d e u x \n t r o i s \n q u
```

```
0000020 a t r e \n c i n q \n s i x \n s e
```

```
0000040 p t \n \n \n \n \n \n \n
```

```
0000050
```

```
od -x monfichier.mac
```

```
0000000 6e75 640d 7565 0d78 7274 696f 0d73 7571
```

```
0000020 7461 6572 630d 6e69 0d71 6973 0d78 6573
```

```
0000040 7470 0d0d 0d0d 0d0d
```

```
0000050
```

```
od -c monfichier.unix
```

```
0000000 u n \r d e u x \r t r o i s \r q u
```

```
0000020 a t r e \r c i n q \r s i x \r s e
```

```
0000040 p t \r \r \r \r \r \r \r
```

```
0000050
```

Codage des fins de ligne sous Dos par les deux caracteres 0x0D et 0x0A.

Codage des fins de ligne sous Unix par le seul caractere 0x0A.

Codage des fins de ligne sous Macintosh par le seul caractere 0x0D.

10. Utiliser les commandes de comparaison de fichiers de Linux pour tester le contenu d'un même fichier texte en version Unix et Dos.

```
cmp monfichier.unix monfichier.dos
```

```
diff monfichier.unix monfichier.mac
```

11. Combien de lignes, de mots et de caractères comportent les fichiers "montexte.unix", "montexte.dos" et "montexte.mac"?

```
wc monfichier.unix
```

```
wc monfichier.dos
```

```
wc monfichier.mac
```

## Exercice 4 : Commandes

Déterminer les commandes permettant de réaliser les actions suivantes:

1. Déterminer le répertoire par défaut dans la hiérarchie des répertoires? , **pwd**
2. Y a t-il des fichiers, des répertoires dans ce répertoire?, **ls -la**  
**Oui, mais ce sont les entrées vers la racine**  
**du repertoire et la racine du repertoire père**
3. Entrer du texte dans un fichier nommé "Mon\_fichier".  
**echo aaaaaaaaaaaaaaaaaaaa >Mon\_fichier**
4. Lister le contenu de "Mon\_fichier"  
**cat Mon\_fichier**
5. Lister le répertoire courant. **ls**  
**Mon\_fichier**  
**ls -la**
6. Lister les répertoires /bin et /dev.  
**ls /bin**  
**ls /dev**
7. Créer sous votre répertoire deux sous-répertoires : "Source" et "Data". **mkdir Source**  
**Data**
8. Se positionner sous "Source". **cd source**
9. Listez le répertoire courant. **ls -la**
10. Revenir sous le répertoire de départ et détruire "Source".  
**cd ..**  
**rmdir Source**
11. Créer un deuxième fichier nommé "Mon\_fichier\_2". **touch Mon\_fichier\_2**
12. Copier chaque fichier en nom\_de\_fichier.old. (fichier .OLD est tout simplement une sauvegarde d'un fichier originale)  
**cp Mon\_fichier Mon\_fichier.old**  
**cp Mon\_fichier\_2 Mon\_fichier\_2.old**  
**ls -la**
13. Créer un répertoire "Old". **mkdir old**
14. Déplacer les fichiers avec l'extension old vers le répertoire "Old".  
**mv \*.old Old**  
**ls -la Old**

**ls -la**

15. Copiez les fichiers sans extension dans le repertoire "Data".

**cp \* Data ,,, ls -la**

16. Sous votre repertoire de depart, créez un lien materiel "Mon\_lien" equivalent à "Mon\_fichier\_2".

**ln Mon\_fichier\_2 Mon\_lien**

**ls -la**

17. Lister les deux fichiers "Mon\_lien" et "Mon\_fichier\_2" en affichant leur numero d'inode. Que remarquez vous?

**ls -lai**

Leurs numeros d'inode sont identiques donc ces deux fichiers n'en sont physiquement qu'un seul.

18. Supprimer "Mon\_lien".

**rm Mon\_lien**

**ls -lai**

"Mon\_fichier\_2" a-t-il disparu? "Mon\_fichier\_2" existe toujours.

19. Sous votre repertoire de depart, créez un lien symbolique "Mon\_nouveau\_lien" sur "Mon\_fichier\_2".

**ln -s Mon\_fichier\_2 Mon\_nouveau\_lien**

20. Lister les deux fichiers "Mon\_nouveau\_lien" et "Mon\_fichier\_2". **ls -la**

Que remarquez vous?

Le lien est clairement mis en evidence par le caractere l et par l'affichage du nom du fichier lie.

La taille n'est pas bonne.

21. Supprimer "Mon\_fichier\_2".

**rm Mon\_fichier\_2**

**ls -la**

"Mon\_nouveau\_lien" a-t-il disparu?

Le lien existe toujours mais est brise.

Il est caracterisé par un code couleur spécifique dans le shell utilisé.

22. Quelle est la taille totale des fichiers contenus dans votre repertoire?

**du -b**

La taille est approximative en raison de l'utilisation de blocs de 512 octets.

23. Effacer tous les fichiers créés.

```
rm -r *
```

```
ls -la
```

## Exercice 5 : Mise en place d'un espace de travail pour 4 utilisateurs

### Mise en place d'un espace de travail pour 4 utilisateurs

#### 1. Création des groupes et des utilisateurs

- Création de 2 groupes  
**groupadd group1**  
**groupadd group2**  
**cat /etc/group**  
...  
**group1:x:1001:**  
**group2:x:1002:**
- Création des 4 utilisateurs avec création de leurs répertoires home:  
**useradd -m u1**  
**useradd -m u2**  
**useradd -m u3**  
**useradd -m u4**  
**cat /etc/passwd**  
...  
**u1:x:1001:100::/home/u1:/bin/sh**  
**u2:x:1002:100::/home/u2:/bin/sh**  
**u3:x:1003:100::/home/u3:/bin/sh**  
**u4:x:1004:100::/home/u4:/bin/sh**  
**ls -l /home**  
**total 20**  
...  
**drwxr-xr-x 2 u1 users 4096 2007-02-01 12:12 u1**  
**drwxr-xr-x 2 u2 users 4096 2007-02-01 12:12 u2**  
**drwxr-xr-x 2 u3 users 4096 2007-02-01 12:12 u3**  
**drwxr-xr-x 2 u4 users 4096 2007-02-01 12:12 u4**
- Placement des utilisateurs dans leurs groupes  
**usermod -G group1 u1**  
**usermod -G group1,group2 u2**  
**usermod -G group2 u3**  
**usermod -G group1,group2 u4**  
**cat /etc/group**  
...  
**group1:x:1001:u1,u2,u4**  
**group2:x:1002:u2,u3,u4**
- Changement de propriétaire des répertoires  
**chown u1:group1 /home/u1**  
**chown u2:group1 /home/u2**

```
chown u3:group2 /home/u3
chown u4:group2 /home/u4
```

- Création des répertoires communs

```
mkdir /home/group1
mkdir /home/group2
```
- Mise en place des permissions pour permettre aux utilisateurs d'écrire dans le répertoire de leur groupe

```
chgrp group1 /home/group1
chgrp group2 /home/group2
```
- Mise en place de la permission pour protéger de l'effacement tout en autorisant l'écriture:  
A ce niveau du cours, on ne sait pas résoudre la dernière problématique.  
Soit on fait confiance aux utilisateurs,  
soit on passe par root pour ajouter les fichiers et on ne donne pas la permission «w».  
Cas le plus permissif:

```
chmod g=rwx /home/group1
```

r : lecture, w : écriture, x : exécution

Cas pratique

<pre>Su mkdir group 1 chmod g=rwx /home/group1</pre> <p>J'ai créé un répertoire nommé "group1" en tant que super utilisateur.</p>	<pre>Su u1 rmdir group 1</pre> <p>Impossible de supprimer le groupe ; aucune permission accordée. Seul le super utilisateur est autorisé à supprimer le répertoire du groupe 1, car il en est le propriétaire.</p>
---	--

- 
- Activation d'un utilisateur

```
passwd u1
Enter new UNIX password:
Retype new UNIX password:
passwd : le mot de passe a été mis à jour avec succès
cat/etc/shadow
...
u1:$1$kiUUra9s$AxchvKz0J9OBJPXO8qNf./:13545:0:99999:7:::
u2!:13545:0:99999:7:::
u3!:13545:0:99999:7:::
u4!:13545:0:99999:7:::
```

Créez un script shell (par exemple, addition.sh).  
À l'intérieur du script, demandez à l'utilisateur d'entrer deux nombres.  
Ajoutez ces deux nombres.  
Affichez le résultat.

<pre>#!/bin/bash  # Demander à l'utilisateur d'entrer le premier nombre echo "Entrez le premier nombre : " read nombre1  # Demander à l'utilisateur d'entrer le deuxième nombre echo "Entrez le deuxième nombre : " read nombre2  # Calculer la somme des deux nombres somme=\$((nombre1 + nombre2))  # Afficher le résultat echo "La somme de \$nombre1 et \$nombre2 est : \$somme"  chmod +x addition.sh ./addition.sh</pre>
<p>Créez un nouveau script shell (par exemple, parite.sh).  Demandez à l'utilisateur d'entrer un nombre.  Utilisez une déclaration if pour vérifier si le nombre est pair ou impair.  Affichez un message indiquant si le nombre est pair ou impair.</p>
<pre>#!/bin/bash  # Demander à l'utilisateur d'entrer un nombre echo "Entrez un nombre : " read nombre  # Vérifier la parité du nombre if [ \$((nombre % 2)) -eq 0 ]; then     echo "Le nombre \$nombre est pair." else     echo "Le nombre \$nombre est impair." fi</pre>
<p>Créez un script shell (par exemple, carres.sh).  Demandez à l'utilisateur d'entrer un nombre N.  Utilisez une boucle for pour afficher les carrés des nombres de 1 à N.</p>
<pre>#!/bin/bash  # Demander à l'utilisateur d'entrer un nombre echo "Entrez un nombre N : " read N  # Utiliser une boucle for pour afficher les carrés des nombres de 1 à N echo "Les carrés des nombres de 1 à \$N sont :" for ((i = 1; i &lt;= N; i++)); do     carre=\$((i * i))     echo "\$i^2 = \$carre" done</pre>



## **Annexe : Les principaux répertoires sont:**

**/bin** : contient les commandes de base (sous Unix, presque toutes les commandes sont "externes", et non intégrées au shell comme sous DOS).

**/boot** : contient les informations nécessaires au démarrage de la machine.

**/dev** : contient les fichiers spéciaux correspondant aux périphériques.

**/etc** : la plupart des fichiers de configuration.

**/home** : contient les répertoires personnels des utilisateurs. Par exemple, l'utilisateur toto a généralement pour repertoire /home/toto.

**/lib** : contient les principales bibliothèques partagées (équivalent des DLL de Windows).

**/lost+found** : quand un disque crashe pour une raison ou pour une autre, on utilise le programme fsck pour réparer (équivalent de scandisk); c'est là qu'il dépose les fragments de fichiers perdus.

**/mnt** : les répertoires utilisés pour monter temporairement un système de fichiers (disquette, CD-ROM...).

**/opt** : c'est là qu'on installe les logiciels commerciaux.

**/proc** : un répertoire factice, dont les fichiers contiennent des infos sur l'état du système et des processus en cours d'exécution.

**/root** : le répertoire de l'administrateur système. Il n'est pas sous /home pour des raisons que je détaillerai plus loin.

**/sbin** : les commandes de base nécessaires à l'administration système (vérification et réparation des disques, mise en place du réseau...).

**/tmp** : les fichiers temporaires.

**/usr** : les logiciels installés avec le système.

**/usr/X11R6** : X-Window.

**/usr/bin** : les exécutables.

**/usr/dict** : les dictionnaires (pour les correcteurs d'orthographe et les craqueurs de mots de passe).

**/usr/doc** : la doc.

**/usr/etc** : des fichiers de config.

**/usr/games** : les jeux.

**/usr/include** : les fichiers d'en-tête pour la programmation.

**/usr/info** : la doc au format GNU info.

**/usr/lib** : les DLL non vitales.

**/usr/local** : une sous-hiérarchie qui contient des logiciels compilés sur place à partir des sources. Organisation similaire à /usr.

**/usr/man** : le manuel en ligne. Les fichiers sont compressés.

**/usr/sbin** : principalement les serveurs réseau.

**/usr/share** : des fichiers de données.

**/usr/spool** : généralement un lien symbolique vers /var/spool.

**/usr/src** : les sources de certains logiciels, principalement le noyau de Linux.

**/var** : des données fréquemment réécrites.

**/var/catman** : les pages du manuel décompressées (ça se fait à la demande).

**/var/lib** : des bases de données, des fichiers de config...

**/var/local** : complète /usr/local de la même façon que /var complète /usr.

**/var/lock** : des fichiers qui servent à marquer l'utilisation de certaines ressources. Par exemple, quand un logiciel se sert du modem, il crée un fichier ici pour le signaler.

**/var/log** : le journal du système.

**/var/run** : principalement des infos sur les serveurs en fonctionnement.

**/var/spool :** les spools: tout ce qui est "de passage" en attendant d'être utilisé par un logiciel. Ca inclut entre autres le mail, les news, les files d'attente des imprimantes...

**/var/tmp :** des fichiers temporaires.