

Institut Supérieur d'Informatique et de Mathématiques de Monastir

Université de Monastir



Systèmes d'exploitation 1

Fascicule de Travaux

Pratiques

Responsable du cours :

SEKMA MANEL

2022/2023

Niveau : Cycle préparatoire intégré 1 (PI1)

TP N°1 : Système d'exploitation MS-DOS

Quelques commandes de base

Objectif : Le but de ce premier TP est de manipuler quelques commandes de base de système d'exploitation MS-DOS.

Durée : 1 :30

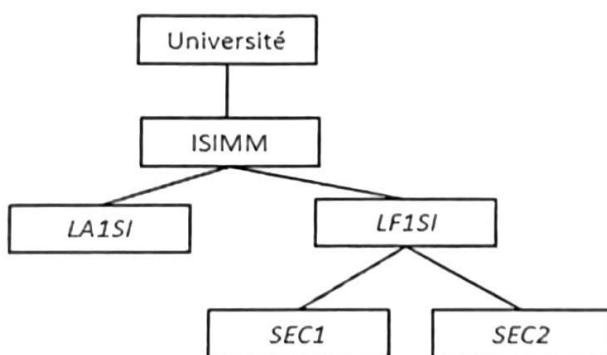
Exercice 1 :

Chercher la syntaxe et la description des commandes suivantes :

- ▶ help.....
 - ▶ cd.....
 - ▶ copy.....
 - ▶ rd.....
 - ▶ ren.....
 - ▶ find.....

Exercice 2 :

1. Ecrire la liste de commandes pour créer l'arborescence ci-dessous sous la racine C :



2. Renommer le répertoire **SEC1** par **SECTION1** et le répertoire **SEC2** par **SECTION2**:

3. Créer deux nouveaux répertoires sous **LA1SI** nommé **TP1** et **TP2** :

4. Affiche tous les fichiers et sous-répertoires présents dans le répertoire **Université** :

5. Supprimer le répertoire **TP1** :

6. Dessiner la nouvelle arborescence :

Exercice 3 :

1. Ecrire la commande permettant d'afficher tous les noms de fichiers dont l'extension est **.txt** dans le répertoire **TP1** :

2. Donner la commande permettant d'afficher tous les noms de fichiers qui commencent par "univ" suivi de 6 autres lettres dont l'extension est **.txt** dans le répertoire **LF1SI** :

3. Donner la commande permettant d'afficher tous les noms de fichiers y compris ceux des fichiers cachés d'extension **.exe** sous le répertoire Université.

4. Donner la commande permettant de supprimer toutes les fichiers dont les noms commencent par "ex" et d'extension **.txt** sous le répertoire **SECTION2** :

TP N°2: Système d'exploitation LINUX

Prise en main de l'environnement Linux

Objectif : Le but de ce TP est de découvrir l'interface ainsi le Shell de système Linux, ainsi de se familiariser avec le Help

Exercice 1 :

1. Connectez-vous à un poste.
2. Quels sont votre login et mot de passe ?
3. Ouvrir une fenêtre Shell, quel est le prompt dans cette fenêtre ?
4. Où trouvez-vous dans l'arborescence Linux ?
5. Que renvoie cette commande sur votre poste :

a. [user@poste10 user]\$pwd

6. Tapez la commande suivante :

[user@poste10 user]\$hostname

- a. Que renvoie-t-elle sur votre poste ?
- b. Vérifier le résultat de la commande précédente ?

7. Lancez la commande suivante :

[user@poste10 user]\$hostname –help

- a. Quel est le rôle de la commande précédente ?
- b. Notez quelques options qui vous paraissent intéressantes.

8. Lancez la commande :

[user@poste10 user]\$ls

- a. Quels fichiers trouvez-vous ?
- b. Etes-vous sûr d'avoir vu tous les fichiers ?

9. Lancez les commandes suivantes :

[user@poste10 user]\$ls -l

[user@poste10 user]\$ls -al

[user@poste10 user]\$ls -Ral

- a. Quel est le rôle de l'option **l** de la commande ls ?
- b. Quel est le rôle de l'option **a** de la commande ls ?
- c. Quel est le rôle de l'option **R** de la commande ls ?

10. Donner le rôle de chacune des commandes suivantes :

[user@poste10 user]\$clear

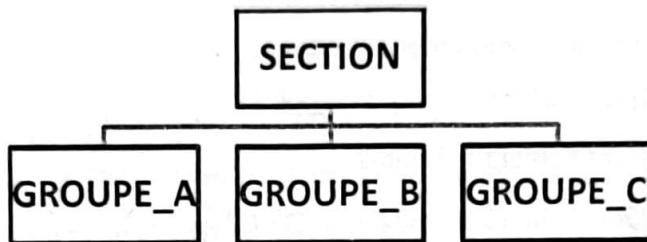
[user@poste10 user]\$history

[user@poste10 user]\$exit

11. Chercher la syntaxe et la description des commandes suivantes :

- mkdir.
- cd.
- cp.
- mv.
- rm.

12. Créez l'arborescence ci-dessous sous votre fichier répertoire personnel :



13. Tapez les commandes suivantes :

[user@poste10 user]\$touch institut

- a. Quel est le rôle de la commande précédente ?
- b. Déplacer le fichier précédent dans le répertoire **GROUPE_A** ?
- c. Créer une copie du fichier précédent dans le répertoire **GROUPE_B** ?

14. Détruisez tous les fichiers que vous avez créés.

15. Lancer la commande :

[user@poste10 user]\$exit

[user@poste10 user]\$reboot

16. Positionnez-vous dans la racine et tapez les commandes suivantes :

[user@poste10 user]\$pwd

[user@poste10 user]\$ls

- a. Notez ce qui s'affiche sur votre poste.
- b. Afficher sur l'écran l'arborescence à partir de la racine.

17. Positionnez-vous dans le répertoire /etc et listez quelques fichiers.

18. Positionnez-vous dans le répertoire /home et listez quelques fichiers.

19. Positionnez-vous dans le répertoire /dev et listez quelques fichiers.

20. Positionnez-vous dans le répertoire /usr et listez quelques fichiers.

Exercice 2 :

1. Créer un fichier nommé **fichier1** par un éditeur de texte et enregistrer sous votre répertoire personnel. Le contenu de fichier est le suivant :

La différence essentielle de GNU/Linux par rapport à d'autres systèmes d'exploitation concurrents - comme Mac OS, Microsoft Windows - est d'être un système d'exploitation libre, apportant quatre libertés aux utilisateurs, définies par la licence publique générale GNU (GPL), les rendant indépendants de tout éditeur et encourageant l'entraide et le partage.

2. Donner les résultats de chacune des commandes suivantes :

```
[user@poste10 user]$ls -l fichier1
```

```
[user@poste10 user]$cat fichier1 > fichier2
```

```
[user@poste10 user]$cat fichier1 fichier2 >> fichier 3
```

3. Donner le rôle des commandes suivantes :

- **compress**.
- **uncompress**.
- **gzip**.
- **gunzip**.

4. Donner la caractéristique de fichier **fichier1** avant et après cette commande, commenter.

```
[user@poste10 user]$compress -v fichier1
```

TP N°3: Commandes de base de LINUX

Exercice 1 :

1. Que font les commandes suivantes ? Proposez éventuellement pour chacune une meilleure façon de procéder (plus courte).

- a. `cd ..`.
- b. `ls .`
- c. `cp /etc/passwd .`
- d. `mkdir ./toto`
- e. `cp /etc/hosts ./toto/..`
- f. `mkdir vide ; mv ./toto ./vide/tata`
- g. `ls /home/..../usr/./bin/..../lib/linuxconf`

2. Le répertoire `/usr/include` contient les fichiers d'entête standards en langage C (`stdlib.h`, `stdio.h`, etc.).

a. Créer un répertoire nommé `inc` dans votre répertoire de connexion. En utilisant une seule commande, y copier les fichiers du répertoire `/usr/include` dont le nom commence par `std` et termine par `.h`.

b. Afficher la liste des fichiers de `/usr/include` dont le nom comporte 3 caractères suivis de `.h`.

c. La commande `echo` affiche simplement ses arguments, après éventuelle expansion des métas caractères et des variables par shell.

- Que fait `echo PATH` ?
- Que fait `echo $PATH` ?
- Comment afficher la liste des noms de fichiers du répertoire courant avec `echo` ?

Exercice 2 :

1. Positionnez-vous sur votre répertoire personnel.

2. Créez un répertoire nommé `TP2` dans le répertoire courant.

3. Affichez le contenu du répertoire courant.

4. Déplacez-vous dans le dernier répertoire créé.

5. Affichez le contenu du répertoire courant.

6. Copiez le fichier `/etc/passwd` dans le répertoire courant avec le nom `passwd1`.

7. Affichez le contenu du fichier `passwd1`.

8. Affichez les dix dernières lignes du fichier **passwd1**.
9. Affichez les dix premières lignes du fichier **passwd1**.
10. Créez un répertoire nommé **SUB** dans le répertoire courant.
11. Affichez le contenu du répertoire courant.
12. Créez un répertoire nommé **SUB2** dans le répertoire courant.
13. Affichez le contenu du répertoire courant.
14. Supprimez le répertoire **SUB2**.
15. Affichez le contenu du répertoire courant.
16. Dupliquez le fichier **passwd1** dans le répertoire courant avec comme nouveau nom de fichier **passwd2**.
17. Regardez les numéros d'inode de ces deux fichiers avec l'option **-i** de la commande **ls**.
18. Renommez ce dernier fichier en **dup**.
19. Regardez les numéros d'inode de ces deux fichiers avec l'option **-i** de la commande **ls**.
20. Créez, dans le répertoire **SUB** mais sans vous déplacer dans ce dernier, le lien physique **lienphy.txt** sur le fichier **passwd1**.
21. Affichez le numéro d'inode des fichiers **passwd1** et **lienphy.txt**.
22. Créez, dans le répertoire **SUB** mais sans vous déplacer dans ce dernier, le lien symbolique **liensym.txt** sur le fichier **passwd1**.
23. Affichez le numéro d'inode des fichiers **passwd1** et **liensym.txt**.
24. Utilisez la commande **cat** pour afficher le contenu des fichiers **lienphy.txt** et **liensym.txt**.
25. Déplacez le fichier **passwd1** dans le répertoire **SUB**.
26. Utilisez la commande **cat** pour afficher le contenu des fichiers **lienphy.txt** et **liensym.txt**.
27. Déplacez le fichier **passwd1** dans le répertoire courant en le renommant en **passwd3**.
28. Revenez sur votre répertoire personnel.
29. Affichez toutes les éléments non cachés de l'arborescence à partir du répertoire personnel en utilisant l'option d'affichage récursif de la commande **ls**.
30. Effacez les éléments créés lors de cette balade par la commande **rm -rf TP1**.

Attention : Cette dernière commande est à utiliser avec précaution. Regardez la page manuelle de la commande **rm** pour vérifier ce que font les options **-r** et **-f** et vérifiez bien que vous êtes dans votre répertoire personnel pour lancer la commande.

Exercice 3 :

1. Dans votre répertoire courant, créez en une commande les fichiers suivants : **annee1, Annee2, annee4, annee45, annee41, annee510, annee_saucisse, banane, mois.**
2. Créer le répertoire **Year** dans votre répertoire courant, en une commande, déplacez les fichiers précédemment créés dans le répertoire **Year**.
3. Lister tous les fichiers, du répertoire **Year**, dont les noms :
 - a. Se terminant par **5**.
 - b. Commençant par **annee4**.
 - c. Commençant par **annee4** et de 7 lettres maximum.
 - d. Commençant par **annee** avec aucun chiffre numérique
 - e. Contenant la chaîne **ana**.
 - f. Commençant par **a** ou **A**.

4. Copier les fichiers dont l'avant dernier caractère est un **4** ou **1** dans le répertoire **/tmp** en une seule

Exercice 4 :

1. Créez le répertoire **tmp** sous votre répertoire de connexion et positionnez les droits d'accès comme suit : **rwxr-x---**.
2. Créez un fichier vide de nom **motjour** sous **tmp** avec la commande **touch**, positionnez les droits d'accès **rwxr-x---** et lisez son contenu.
3. Retirez pour le propriétaire le droit en lecture de **motjour** et essayez de le relire.
4. Retirez pour le propriétaire le droit en écriture de **tmp** et essayez de détruire **motjour**.
5. Retirez pour le propriétaire le droit en lecture de **tmp** et essayez de lister son contenu.
6. Retirez le droit **x** pour **tmp** et essayez de vous positionner sur ce répertoire.
7. Positionnez pour le propriétaire les droits d'accès **rwx** du répertoire **tmp**.
8. Essayez de vous donner le droit en écriture sur le répertoire de connexion d'un membre de votre groupe.
9. Effacez tout le contenu du répertoire **tmp** et le répertoire lui-même.

TP N°4: Les commandes LINUX sur les fichiers

Exercice 1 :

1. Donner la définition du **umask**.
2. Donner la commande pour connaître **umask** de votre système.
3. Donner la commande pour changer **umask** à 022.
4. Taper la commande **umask**, de manière à ce que les fichiers lors de leur création aient par défaut les droits **640** et les répertoire **750**.
5. Essayer de supprimer ou modifier le fichier **/etc/passwd**. Que se passe-t-il ? Expliquer la situation à l'aide de la commande.
6. A l'aide de la commande **id**, vérifier votre identité et le(s) groupe(s) auquel vous appartenez.
7. Créer un fichier vide nommé **droit** sous votre répertoire personnel, vérifier les droits d'accès de ce fichier. Expliquer ?
8. Donner la commande pour ajouter le droit d'exécution pour le propriétaire du fichier **droit** ?
9. Donner la commande pour retirer le droit d'écriture pour tous les utilisateurs du fichier **droit** ?
10. Créer un répertoire nommé **secret** sous votre répertoire personnel.
11. Quels sont les droits d'accès du répertoire **secret** ? Pourquoi ?
12. Changer les droits d'accès du répertoire **secret** pour qu'il soit lisible uniquement par vous-même. Les fichiers placés dans ce répertoire sont-ils lisibles par d'autres membres de votre groupe ?
13. Créer un répertoire nommé **connasseurs** tel que les autres utilisateurs ne puissent pas accéder ou lister son contenu mais puissent lire les fichiers qui y sont placés.
14. Donner la syntaxe et la description de la commande **chown** avec les options **-f**, **-R**.
15. Créer un fichier nommé **expchown** sous votre répertoire personnel.
16. Si vous êtes le super-utilisateur, donner la commande pour changer le propriétaire du fichier **expchown** (*Exemple*: si le propriétaire du fichier est l'utilisateur **user1**, changer le propriétaire du fichier à l'utilisateur **user2**).
17. Donner le résultat de l'affichage long du fichier **expchown**.
18. Donner la syntaxe et la description de la commande **chgrp**.
19. Donner les descriptions des commandes suivantes :
 - ▶ [s2@localhost s2]\$**chgrp s1 fich1**
 - ▶ [s3@localhost s3]\$**chgrp s2 *.doc**
 - ▶ [s4@localhost s4]\$**chgrp s3 exemple.c**

Exercice 2 :

Le répertoire **/usr/include** contient les fichiers d'entête standards en langage C (**stdlib.h**, **stdio.h**, etc).

1. En utilisant une seule commande, copier les fichiers du répertoire **/usr/include** dont le nom commence par **std** dans le répertoire **inc** sous votre répertoire personnel.
2. Afficher la liste des fichiers de **/usr/include** dont le nom commence par **a**, **b**, ou **c**.
3. Au lieu d'afficher le résultat, le placer directement dans un fichier nommé **abc.list** de votre répertoire de connexion.
4. Afficher le contenu du fichier **abc.list** en utilisant la commande **cat**.
5. Copier avec **cat** le contenu du fichier **abc.list** dans un nouveau fichier nommé **copie**.
6. Toujours avec **cat**, créer un nouveau fichier nommé **cat.list** formé par la mise bout à bout (concaténation) des fichiers **abc.list** et **copie**. Vérifier que le nombre de lignes a bien doublé à l'aide de la commande **wc**.
7. On peut aussi utiliser **cat** pour créer un petit fichier sans passer par un éditeur de texte. Créer un fichier nommé **temps** contenant les lignes de texte suivantes :

Debut du fichier:

Ce fichier est cree sans inviter l'editeur de texte vi.

Pour arreter le saisie de texte taper CTRL-d pour terminer l'entree.

8. Avec **cat**, ajouter une ligne "**Fin du fichier.**" à la fin du fichier **temps**.
9. En une seule ligne, faire afficher le nombre de fichiers de **/usr/include** dont le nom contient la lettre **t** (utiliser un tube " | " avec la commande **wc**).

Exercice 3 :

1. Afficher, à l'aide de la commande **find**, la liste des fichiers nommés **stdio.h** situés sous le répertoire **/usr/include**.
2. Chercher dans toutes l'arborescence les fichiers dont le nom se termine par **.c**, redirigez les erreurs vers le fichier poubelle **/dev/null**.
3. Chercher dans **/usr** les fichiers dont la taille dépasse **1 Mo** et dont les droits sont fixés à **755**.
4. Combien il y a de fichiers dans toutes l'arborescence vous appartenant et ayant les droits fixé à **666**.

Exercice 4 :

1. Afficher, à l'aide de la commande **grep**, les lignes du fichier **stdio.h** (sous **/usr/include**) qui contiennent la chaîne **define**.
2. En combinant les commandes **find** et **grep**, afficher toutes les lignes contenant le mot **#def** dans les fichiers dont le nom se termine **.h** situés sous **/usr/include**.
3. Trouver dans le fichier **/etc/passwd** :
 - a ► Lignes vides.
 - b ► Lignes contenant au minimum 3 lettres successives.
 - c ► Lignes contenant les mots terminés par **ool**.
 - d ► Lignes contenant les mots terminés par **guage** ou **gage**.
 - e ► Lignes contenant les chaînes contenant **nda**.
 - f ► Lignes terminées par **lse**.
 - g ► Lignes contenant les sous chaînes **arp** ou **nam**.
 - h ► Lignes ne commençant pas par un chiffre.
 - i ► Lignes commençant par une voyelle ou un chiffre.
 - j ► Lignes ne commençant pas par une voyelle.
 - k ► Lignes commençant ou se terminant par une voyelle.
 - l ► Lignes qui font moins de 10 caractères et au plus 35 caractères.

Exercice 5 :

1. Copier le fichier **/etc/passwd** dans votre home directory avec le nom **passwd**.
2. Afficher les cinq premières, puis les 5 dernières lignes du fichier **passwd**.
3. Afficher la septième ligne de ce fichier (et elle seule) en une seule ligne de commande (utiliser un tube).
4. Editer uniquement les champs contenant le login et le home directory du fichier **passwd**.
5. Triez **passwd** sur le nom.
6. Extraire les noms de login et **UID** puis triez suivant les **UID**, le tout en une seule commande, vous redirigerez le tout vers un fichier.
7. Dans le fichier de résultat précédent remplacer les : par des espaces.

TP N°5: Commandes avancées de LINUX

Exercice 1 :

Maintenant, vous êtes le seul utilisateur connecté sur le terminal virtuel 1 (**tty1**).

1. Visualisez les informations au sujet de nom de connexion spécifique. Exécutez et examinez le résultat des commandes suivantes :

```
[user1@post9 user1]$whoami
```

```
[user1@post9 user1]$groups
```

```
[user1@post9 user1]$id
```

2. Visualisez les informations de tous les noms de connexion existants sur la station de travail. A ce stade, il ne devrait y avoir qu'un seul utilisateur connecté au système.

```
[user1@post9 user1]$users
```

```
[user1@post9 user1]$who
```

```
[user1@post9 user1]$w
```

3. Passez au terminal virtuel 2 (**tty2**) et connectez-vous à votre station de travail sous le nom d'un autre utilisateur. Visualisez les informations au sujet de nom de connexion spécifique. Exécutez et examinez le résultat des commandes suivantes :

```
[user1@post9 user1]$whoami
```

```
[user1@post9 user1]$groups
```

```
[user1@post9 user1]$id
```

4. Visualisez les informations de tous les noms de connexion existants sur la station de travail.

```
[user1@post9 user1]$users
```

```
[user1@post9 user1]$who
```

```
[user1@post9 user1]$w
```

5. Sortir du compte de terminal virtuel 2 (**tty2**).

Exercice 2 :

1. Donnez la syntaxe et la description des commandes suivantes :

► **gtop**.

► **fork.**

► **exec.**

2. Que fait la commande suivante :

```
[user1@post9 s12]$pstree
```

3. Ecrire la ligne de commandes qui permet d'afficher toutes les lignes de la saisie de la commande **ps** avec l'option **-alx** contenant "lpd".

4. Afficher une liste de processus interactif et continuellement mise à jour à l'aide de **top** ou **gtop**.

5. Afficher une liste de tous les processus même ceux qui ne sont pas contrôlé par un terminal.

6. Quelle est l'option qui permet d'afficher plus d'informations sur les processus.

7. Que fait la commande suivante :

```
[user1@post9 s12]$kill -1
```

8. Envoyer un des signaux à l'un des processus identifié par son PID, exemple **TERM**.

9. Que fait le signal **TERM** et quelle démarche équivalente peut-on effectuer pour avoir le même résultat.

10. Si cette démarche échoue quel signal peut-on envoyer pour avoir le même résultat.

11. Exécuter commande suivante :

```
[user1@post9 s12]$jobs
```

12. Comment lancer un processus en arrière-plan et par quelle commande peut-on le ramener en premier plan (avant plan).

Exercice 3 :

Dans cette séquence, vous lancerez plusieurs processus puis vous utiliserez les fonctions de contrôle de travail du **shell bash** pour les gérer et les contrôler. Vous commuterez d'une console virtuelle à l'autre. Veillez à soigneusement noter quelle console est utilisée avec quelle commande.

1. Commencer par vous inscrire sur les terminaux virtuels 1 et 2 (**tty1**, **tty2**) en utilisant votre compte.

2. Passer à **tty1** et démarrer un processus à l'aide de la commande suivante :

```
[user1@post9 user1]${while true;do echo -n A >> log;sleep 1;done}
```

Noter que ce terminal est désormais utilisé par votre processus (exécuté en premier plan). Le processus que vous avez lancer ajoute la lettre 'A' de façon répétée à une seconde d'intervalle au fichier **log**. Passez à **tty2** et tapez la commande pour voir une séquence de caractère 'A' qui s'allonge.

3. Passer à **tty1** et interrompez le processus en cours d'exécution en appuyant sur **CTR+Z** sur cette console. Le Shell reporte que le processus a été arrêté et vous donne un numéro de travail **[1]** pour le travail. Passez à **tty2** et vérifier visuellement que le fichier n'augmente plus.

4. Passer à **tty1** et lancer à nouveau le processus, cette fois en arrière-plan. Utilisez **jobs** pour montrer que le travail **[1]** est en cours d'exécution. Enfin passez à **tty2** et vérifier visuellement que le fichier grossit à nouveau.

5. Passer à **tty1** et lancer deux autres processus à l'aide des commandes suivantes :

```
[user1@post9 user1]$ $(while true;do echo -n B >> log;sleep 1;done) &  
  
[user1@post9 user1]$ ^B^C
```

La première commande est presque identique à celle que vous avez exécuté dans la deuxième question (**2.**), il suffit de changer 'A' en 'B'. La deuxième commande exécute la première, remplaçant 'B' par 'C'.

6. Entrez **jobs** et vérifiez que tous les trois sont en cours. Passez à **tty2** et vérifiez que le fichier augmente de nouveau, ajoutant cette fois **A**, **B** et **C** chaque seconde.

7. Dans la question (**3.**), vous avez interrompu le processus de premier plan en appuyant sur **CTR+Z**. En réalité vous avez envoyé un message au processus. Utiliser **kill** pour afficher une liste de signaux avec leur nom et numéro respectif. Utilisez ensuite **kill** pour envoyer un **SIGSTOP** au travail **[1]** pour le suspendre. Faites ce travail sur **tty1**.

8. Enter **jobs** et vérifier que le travail **[1]** apparaît comme arrêté. Passez à **tty2** et vérifiez visuellement que le travail **[1]** est arrêté.

9. Relancer le travail **[1]** à l'aide de **kill** pour lancer un **SIGCONT** au processus. Utilisez la commande **jobs** et **tty2** pour vérifier que les travaux fonctionne à nouveau.

10. Tuez les deux processus. Si vous ne spécifiez pas de signal à envoyer, **kill** enverra par défaut **SIGTERM**. L'utilisation de **SIGTERM** terminera la plupart des programmes. Après avoir signalé les travaux **[2]** et **[3]**, utilisez **jobs** pour vérifier qu'ils sont terminés.

11. Terminez le processus final en le ramenant tout d'abord en avant plan puis en tapant **CTR+C**.

12. Enter **jobs** et vérifiez qu'aucun travail n'est affiché. Passez à **tty2** et vérifiez visuellement que l'activité s'est arrêtée. Arrêtez la commande **tail** en appuyant sur **CTR+C**, et déconnectez-vous de **tty2**.

13. Passez à **tty1** et nettoyer en effaçant le fichier **log**.

Exercice 4 :

En utilisant le langage de commande Shell et les commandes "**echo**", "**exec**" et "...", nous allons voir la gestion des processus, la création de processus fils et l'exécution d'une commande dans un processus.

1. Construire avec l'éditeur de texte **vi**, les trois fichiers de commandes **com1**, **com2** et **com3** tels qu'ils sont décrits ci-dessous :

```
[user1@post9 user1]$cat com1
echo "Je suis dans com1."
ps
com2
echo "Je suis de retour dans com1."
[user1@post9 user1]$cat com2
echo "Je suis dans com2."
ps
com3
echo "Je suis de retour dans com2."
[user1@post9 user1]$cat com3
echo "Je suis dans com3."
ps
```

2. Essayer de lancer la commande **com1**. Que se passe-t-il ? Que devez-vous faire pour que l'exécution puisse être lancée ? Une fois ce problème résolu, lancer la commande **com1** en rediriger les traces dans le fichier **trace1**.

3. Reprendre les fichiers correspondant aux commandes **com1**, **com2** et **com3** et remplacer les appels aux commandes "**com?**" Par ".**com?**". Vous pouvez créer de nouveaux fichiers **com1p**, **com2p** et **com3p** ainsi modifiés.
4. Lancer la nouvelle commande **com1p** en redirigeant les traces dans le fichier **trace2**.
5. Reprendre les fichiers correspondant aux commandes **com1**, **com2** et **com3** et remplacer les appels aux commandes "**com?**" par "**exec com?**". Vous pouvez créer de nouveaux fichiers **com1e**, **com2e** et **com3e** ainsi modifiés.
6. Lancer la nouvelle commande **com1e** en redirigeant les traces dans le fichier **trace3**.
7. A partir de l'analyse des trois fichiers de trace obtenus (**trace1**, **trace2**, **trace3**), expliquer les différences entre l'appel d'une commande **com**, l'utilisation de la commande "." (.**com**) et l'utilisation de la commande **exec (exec com)**.

TP N°6 : Programmation Shell

Exercice 1 :

Soit le script Shell suivant :

```
#!/bin/sh
#-----
# Ma première application
#-----
if [ $# -gt 0 ]
then
    echo "Trop de paramètres"
else
    echo "Le nombre d'utilisateurs connectés est: \n"
    who | wc -l
    echo "Les utilisateurs connectés sont : \n"
    who | cut -c1-9
fi
```

1. Créer un fichier nommé **script1** avec l'éditeur Vi, écrire dedans ce code puis exécuter .
2. Que fait ce script ?

Exercice 2 :

Ecrire un script Shell qui affiche le nombre d'éléments de chaque répertoire passé en argument (on peut passer plusieurs répertoires en argument).

Exercice 3 :

Ecrire un script Shell qui affiche la liste des sous répertoires du répertoire passé éventuellement en argument et qui indique une erreur si l'argument n'est pas un répertoire.

Exercice 4 :

Ecrire un script Shell permettant de déterminer le plus long mot d'un fichier donné en argument.

Exercice 5:

Ecrire un script qui boucle tant qu'une réponse "oui" ou "non" n'est pas saisie.

Exercice 6:

Ecrire un script qui calcule le factoriel d'un entier passé en paramètre.

Exercice 7:

Ecrire un script qui calcule le PGCD de 2 nombres passés en paramètres.

Exercice 8 :

1. Ecrire une fonction qui détermine si une année passée en paramètre est bissextile ou non (retourne 1 si bissextile, 0 sinon).
2. Ecrire un script qui saisit une date sous le format jjmmaaaa et qui détermine si elle est valide ou non.

Exercice 9:

Ecrire un script qui liste un répertoire fourni en paramètre, le répertoire courant sinon, en faisant apparaître d'abords les fichiers et ensuite les répertoires.

Exercice 10:

Ecrire un script qui liste un répertoire fourni en paramètre, le répertoire courant sinon, et qui calcule et affiche sa taille en octets.

Exercice 11:

Ecrire un script qui liste les fichiers exécutables d'un répertoire fourni en paramètre, du répertoire courant sinon et produit un fichier contenant cette liste.

Exercice 12:

Ecrire un script qui admet en paramètre le nombre de disques des tours de Hanoï et propose la solution.

Exercice 13:

Ecrire un script qui affiche la taille du répertoire passé en paramètre, du répertoire courant sinon. La taille doit tenir compte de celles des sous répertoires.