

# TP 1 : Manipulation des fichiers et droits

## 1 Remarque générale

L'ensemble des manipulations doivent être réalisées dans le shell (sans passer par l'interface graphique). Dès que possible, pensez à utiliser les facilités offertes par le shell (historique, auto-complétion, caractères jokers). Pensez à utiliser le manuel (`man`) pour connaître toutes les options d'une commande.

## 2 Utilisation du manuel

Objectifs : se familiariser avec le manuel (commande `man`) et découvrir la commande `ls`.

Cours : slides 1 à 15.

Question :

1. Lancez un terminal.
2. Utilisez la commande `man` pour connaître le rôle de la commande `ls`. Que fait la commande `ls` ?
3. Fermez le manuel (sans fermer le terminal).
4. Testez la commande `ls`.
5. Relancez le manuel pour la commande `ls`.
6. La commande `ls` dispose d'une option permettant d'obtenir un listing détaillé, utilisez la fonction de recherche du manuel pour chercher le mot *listing*.
7. La première occurrence trouvée ne correspond pas à l'option cherchée, cherchez l'occurrence suivante en utilisant les fonctions du manuel. Quelle est l'option qui permet d'obtenir un listing détaillé avec `ls` ?
8. Fermez le manuel (sans fermer le terminal).
9. Testez la commande `ls` avec l'option permettant d'avoir un listing détaillé.

## 3 Auto-complétion et historique

Objectifs : comprendre le principe de l'auto-complétion (touche tabulation) et de l'historique.

Cours : slides 16 à 17.

Question :

1. Utilisez la touche tabulation pour trouver la seule commande dont le nom commence par `whoa`. Quelle est cette commande ? A quoi sert-elle ?
2. Utilisez la touche tabulation pour trouver toutes les commandes dont le nom commence par `who`. Combien y'en a-t-il ?
3. Appuyer sur les flèches haut et bas du clavier, que se passe-t-il ?
4. Testez la commande `history`. Quelle est la première commande que vous avez exécutée d'après cette historique ?
5. Re-exécuter l'avant-avant-avant dernière commande utilisée en utilisant le caractère `!!`.
6. Re-exécuter la dernière commande exécutée commençant par `ma` en utilisant le caractère `!!`.

## 4 Manipulations de fichiers

Objectifs : comprendre l'organisation des fichiers, savoir créer, manipuler et détruire des fichiers et des répertoires. Commandes `pwd`, `cd`, `ls`, `touch`, `mkdir`, `rmdir`, `rm`. Fichiers spéciaux : `"."`, `".."` et `"~"`.

Cours : slides 18 à 25.

Effectuez ensuite les opérations suivantes :

1. Lancez un terminal.
2. Afficher le répertoire absolu dans lequel vous vous trouvez.
3. Créez un sous-répertoire nommé `TPUnix`.
4. Déplacez vous dans ce sous-répertoire.
5. Créez maintenant 3 sous répertoires : `TP1`, `TP2` et `tmp`
6. Vérifier que les 3 répertoires sont bien créé en affichant la liste des fichiers contenus dans le répertoire courant.
7. Déplacez vous dans le sous-répertoire `tmp`
8. Créez un nouveau fichier vide `test`.
9. Faites une copie de ce fichier dans le même répertoire en l'appelant cette fois `test.bak`.
10. Déplacer le fichier `test` dans le répertoire parent `TPUnix`.
11. Revenez dans le répertoire `TPUnix`.
12. Supprimez le fichier `test`.
13. Supprimez le répertoire `tmp`.
14. Revenez à la racine de votre répertoire personnel.

## 5 Caractères jokers

Objectifs : savoir utiliser les caractères jokers `"*"` et `"?"`.

Cours : slide 26.

Effectuez ensuite les opérations suivantes :

1. Créez les fichiers `file1`, `file2`, `file11`, `file12`, `file22` et `essai`.
2. Quel paramètre de la commande `ls` taper pour que ne s'affichent que les fichiers désirés dans chacun des cas suivants (en utilisant les caractères jokers) :
  - `file1 file2`
  - `file1 file11 file12 file2 file22`
  - `file1 file2 essai`
  - `file1 file11 file12`
  - `file12 file22`
  - `file11 file12 file22`

## 6 Recherche

Objectifs : utiliser la commande `find` pour retrouver des fichiers. Comprendre à quoi correspondent les fichiers dont le nom commence par un point.

Cours : slide 27.

Effectuez les opérations suivantes :

1. Trouver tous les fichiers nommés `"stdio.h"` dans le répertoire `/bin/include`.
2. Trouvez tous les fichiers dont le nom commence par un point dans votre répertoire personnel.

3. Listez avec `ls` les fichiers contenus dans votre répertoire personnel. Que constatez vous ?
4. Trouvez quelle option doit être utilisée avec `ls` pour voir les fichiers cachés.
5. Trouvez tous les fichiers vides contenus dans votre répertoire personnel (utiliser le manuel pour trouver quelle option de `find` utiliser).

## 7 Gestion des droits

Objectifs : comprendre le fonctionnement des droits sous Unix et savoir modifier les droits d'un fichier avec la commande `chmod`.

Cours : slides 28 à 38.

Effectuez les opérations suivantes :

1. Quels sont les droits attachés à mon répertoire personnel (`/nfs/user/pers/info/perretb/`) ?
2. Pouvez-vous lire le contenu de mon répertoire personnel ?
3. Pouvez-vous lire le contenu du sous répertoire `public_html` (`/nfs/user/pers/info/perretb/public_html`) ?
4. Essayez d'afficher le contenu du fichier `/nfs/user/pers/info/perretb/public_html/index.html`. Comment expliquez vous que vous puissiez accéder à ce fichier ?
5. Vérifier les droits qui sont attachés à votre répertoire personnel. Si ce n'est pas déjà le cas positionnez les à la valeur octale 711. À quoi cette valeur correspond-elle ?
6. Si ce n'est pas déjà fait, créez un dossier `public_html` dans votre répertoire personnel et assurez vous que le groupe et les autres ne peuvent que traverser ce répertoire.
7. Si ce n'est pas déjà fait, créez un fichier `index.html` dans le répertoire `public_html` et assurez vous que tout le monde a un accès en lecture seul à ce fichier.
8. Modifier le fichier `public_html` en y ajoutant votre nom et prénom.
9. Vérifiez que vous pouvez accéder à la page créée par Internet : `http://www.esiee.fr/~votreLogin/`

## 8 Processus

Objectifs : connaître les processus en cours d'exécution avec les commandes `ps` et `top`, manipuler les processus de premier et arrière plans, tuer un processus avec la commande `kill`.

Cours : slides 39 à 45.

Effectuez les opérations suivantes :

1. Exécutez la commande `top` et trouvez le processus qui utilise le plus de ressources CPU.
2. Quittez la commande `top` (appuyer sur la touche "q").
3. Lancez votre navigateur web favori depuis votre terminal (exécutez la commande `firefox` ou `chrome` par exemple).
4. Sans fermer le navigateur que vous venez de lancer, essayer d'exécuter une nouvelle commande dans le terminal. Vous ne pouvez pas car le navigateur est l'application de premier plan et le terminal attend que ce processus se termine avant d'accepter de nouvelle commande.
5. Sans fermer le navigateur que vous venez de lancer, appuyez simultanément sur le touche `Ctrl+z` (controle et z) : l'exécution du navigateur est suspendu et il est de nouveau possible d'exécuter des commandes dans le terminal.
6. Essayez maintenant d'utiliser le navigateur : ce n'est plus possible car son exécution est suspendue.
7. Exécutez maintenant la commande `bg`, cela relance l'exécution du navigateur mais cette fois en arrière plan. Il est maintenant possible d'utiliser simultanément le navigateur et le terminal.
8. Fermez le navigateur.

9. Relancez le navigateur depuis le terminal mais en ajoutant cette fois le caractère "&" ("et" commercial) à la fin de la commande (par exemple `firefox &`). De cette manière le navigateur est immédiatement lancé en arrière plan et il est possible de servir du terminal et du navigateur.
10. Avec la commande `ps`, déterminez le PID du navigateur.
11. Tuez le navigateur avec la commande `kill`.

## 9 Éditeur de texte en mode ... texte : *vi* (optionnel)

*vi* est l'éditeur de texte historique que vous trouverez sur toutes les machines de type Unix. Son utilisation n'est pas très... intuitive mais en connaître les rudiments peut vous tirer d'une mauvaise situation (si il est impossible de lancer l'interface graphique du système par exemple). Voilà la base :

- Au début, *vi* est en mode *commande* ce qui veut dire qu'il attend que vous lui donniez un ordre. Vous ne pouvez pas écrire de texte dans votre fichier dans ce mode.
- Pour passer en mode *insertion* (pour pouvoir écrire du texte) appuyez sur la touche *i* : vous pouvez maintenant écrire dans votre fichier et vous déplacer avec les flèches.
- Pour quitter le mode insertion appuyez sur *Echap*.
- Pour enregistrer : passez en mode *commande* et écrivez `:w`, finissez par *Entree*.
- Pour quitter : passez en mode *commande* et écrivez `:q`, finissez par *Entree*.

Maintenant une petite mise en pratique. Ouvrez votre fichier `~/public_html/index.html` et modifiez le en y ajoutant un message de votre choix.

## 10 Liens durs et liens symboliques (optionnel)

Sous Unix, chaque fichier est représenté par un i-node avec un numéro unique (nœud d'index). Un répertoire est alors simplement un fichier qui contient une liste d'i-node. Lorsque vous créez un fichier, un nouvel i-node est créé et un lien vers cet i-node est ajouté dans le répertoire courant. Il est possible de créer un lien vers un fichier qui existe déjà avec la commande `ln`, nous aurons alors deux entrées distinctes dans un même répertoire ou des répertoires différentes qui pointent vers le même fichier.

1. Créez un fichier `text.txt`
2. Utilisez la commande `ls` pour connaître le nombre de liens existant vers le fichier `text.txt`
3. Créez un lien s'appelant `text2.txt` vers le fichier `text.txt`
4. Vérifiez le nombre de liens existant vers les fichiers `text.txt` et `text2.txt`. Comparez la taille et les droits des deux fichiers.
5. Modifiez le contenu de `text2.txt`, enregistrez et quittez. Vérifiez que le fichier `text.txt` a bien été modifié.
6. Supprimez le fichier `text2.txt`, vérifiez que `text.txt` existe toujours et que nombre de liens est correcte.
7. Trouvez maintenant l'option de `ln` permettant de créer un lien symbolique (soft link) vers un fichier et utilisez la pour créer un nouveau lien symbolique vers le fichier `text.txt` s'appelant `soft.txt`
8. Affichez la liste détaillée des fichiers de votre répertoire. Que remarquez vous concernant la taille et les droits de `soft.txt` et `text.txt`? Peu-t-on voir que `soft.txt` est un lien symbolique?
9. Supprimez le fichier `text.txt`. Que se passe-t-il pour `soft.txt`?
10. Essayez de créer un lien dure et un lien symbolique vers `/etc/passwd`. Que se passe-t-il? Expliquez.
11. Essayez de créer un lien vers votre répertoire `~/TPUnix/`. Que se passe-t-il? Pourquoi cette règle à votre avis?

## 11 Sticky bit (optionnel)

1. Déplacez vous dans le répertoire `/tmp/`.

2. Créez un fichier `tmp_votrelogin.txt` et supprimez le pour vérifier que vous avez bien le droit d'écriture sur le répertoire.
3. Essayez de supprimer un fichier ne vous appartenant pas. Que se passe-t-il ?
4. Ce comportement peut-il s'expliquer avec la gestion des droits habituels (rwx pour les 3 classes d'utilisateurs).

Le sticky bit permet de moduler le droit d'écriture sur un dossier en fixant que tous les utilisateurs possédant le droit d'écriture peuvent créer des fichiers mais qu'ils ne peuvent supprimer que leurs propres fichiers. Cette technique est généralement utilisée pour les répertoires temporaires.

1. Trouvez comment positionner et ôter le sticky bit sur l'un de vos répertoires.
2. Comment voir si le sticky bit est positionné sur un répertoire.