

## TP3 - UNIX, un peu plus loin.

### Variables

Lorsqu'on travaille dans le `shell`, on se trouve dans un environnement, c'est-à-dire qu'il existe des variables prédéfinies qui contiennent un certain nombre d'informations importantes.

Le nom des variables est généralement en majuscule et on accède à leur contenu en les faisant précéder du signe `$`.

#### Exercice 1.

1. Testez la commande `env`.
2. En utilisant la commande `echo`, déterminez le contenu de `HOME`, de `HOSTNAME` et de `USER`.
3. Comparez `echo ~` et `echo $HOME`.
4. Affichez le contenu de la variable `PATH`. Quel est le contenu des répertoires contenus dans cette variable? Testez la commande `which` sur les commandes `echo`, `touch`. Que conclure?

### Archiver - désarchiver.

Pour faire cet exercice, nous allons utiliser les commandes `tar` et `gzip`. Consultez la documentation pour connaître leurs diverses utilisations.

#### Exercice 2.

1. Placez vous dans votre répertoire `unix`.
2. Créez une archive `tp.tar` contenant les répertoires `tp1` et `tp2`. A l'aide de la commande `ls` et de son option `-l` comparez les tailles respectives de `tp.tar` et des répertoire qu'il contient. Est-ce que les tailles correspondent? Pourquoi?
3. Listez le contenu de l'archive.
4. Créez un répertoire `tp3`. Ajoutez-le à l'archive compressée en utilisant l'option `-r` de `tar`.
5. Comprimez l'archive `tp.tar`. Comparez la taille de l'archive compressée à la taille de l'archive non compressée.
6. Copiez l'archive `tp.tar.gz` dans votre répertoire maison, décompressez-la et en extraire les fichiers. Vous pouvez utiliser `man` pour trouver les options respectives de `tar` et de `gz` qui permettent de faire tout ça.
7. Pour finir, nettoyez votre répertoire maison en supprimant

## Ssh ou l'art de se connecter à distance.

Dans ce paragraphe, nous utiliserons les commandes `ssh` et `scp`. Consultez l'aide-mémoire et la documentation en ligne sur ces deux commandes.

### Exercice 3.

1. A l'aide de la commande `ssh` établissez une connection avec le serveur `monjetas.informatique.univ-paris-diderot.fr`. Explorez l'arborescence des fichiers. Déconnectez-vous en tapant la commande `exit`.
2. A l'aide de la commande `scp` Copiez via la connection `ssh` votre archive `tp.tar.gz`. Dans votre répertoire `tp3`.

## Gestion des droits.

Dans ce paragraphe, vous aurez besoin des commandes `chmod`, `id` et `chown` en plus de toutes celles que vous connaissez déjà. Consultez la documentation sur ces deux commandes.

### Exercice 4.

Chaque fichier est attribué à un utilisateur (son propriétaire) et à un groupe. Ces informations jouent rôle dans la définition des actions autorisées et interdites.

1. Déterminez votre groupe d'appartenance en utilisant la commande `id`.
2. Utilisez la commande `ls -l` depuis votre répertoire `tp3` et repérez les symboles décrivant les droits, le propriétaire et le groupe d'appartenance du fichier.

### Exercice 5.

1. Placez-vous dans le répertoire `tp3`. A l'aide de la commande `echo "une phrase"> fic` écrire le texte *une phrase* dans le fichier `fic`.
2. A l'aide de la commande `ls -l`, notez les permissions actuelles du répertoire `tp3` et du fichier `fic`.
3. Retirez les droits en lecture et en écriture au le fichier `fic`. Essayez d'afficher le contenu du fichier sur la fenêtre du terminal, puis de le modifier en écrivant le texte *une autre phrase*.
4. Un fichier exécutable est un fichier dont vous avez le droit en exécution. Copiez le fichier `hello.c` du répertoire `tp2` dans votre répertoire `tp3`. Compilez-le avec la commande `gcc hello.c`. Quels sont les droits du fichier `a.out`? Retirez les droits en lecture au fichier `a.out`. Exécutez le fichier `a.out` en tapant `./a.out`. Quel problème rencontrez-vous?
5. Rétablissez le droit en lecture et tentez à nouveau d'exécuter le fichier.

Si le sens de chaque droit pour un fichier ordinaire est assez clair, il est parfois moins intuitif pour un répertoire comme vous le verrez dans l'exercice suivant.

#### **Exercice 6.**

1. Dans le répertoire `tp3`, créez un répertoire `test`, copiez le fichier `hello.c` dans le répertoire `test`.
2. Depuis votre répertoire `tp3`, retirez-vous les droits en lecture pour le répertoire `test`. Essayez de lister le contenu du répertoire, puis d'afficher et d'exécuter le fichier `hello.c`. Qu'en déduisez-vous? Rétablissez le droit en lecture pour le répertoire `test`
3. Créez un fichier `nouveau` dans `test`. Retirez à ce fichier `et` au répertoire `test` le droit en écriture. Essayez de modifier le fichier `nouveau`. Rétablissez le droit en écriture au répertoire `test`. Essayez de modifier le fichier `nouveau`, puis de le supprimer. Que constatez-vous?
4. Depuis votre répertoire `tp3` retirez le droit en exécution au répertoire `test`. Essayez de créer, supprimer ou modifier un fichier dans le répertoire `test`, puis de vous y déplacer et d'en faire la liste.
5. Lancez `emacs`. Dans le répertoire `tp3` et grâce à `emacs` créez un fichier `bilan_droits.txt`. Rédigez un récapitulatif succinct sur les droits des répertoires et des fichiers. Sauvez-le, puis en utilisant votre boîte mail habituelle, envoyez-le en pièce jointe à votre chargée de tp.

#### **Exercice 7 (Donner accès à ses fichiers.).**

1. Attribuez au fichier `nouveau` les droits suffisants pour qu'une autre personne de votre groupe Unix puisse y accéder en lecture (mais pas en écriture).
2. Les fichiers des utilisateurs et en particuliers vos fichiers, sont visibles de tous les ordinateurs de la salle. Demandez à votre voisin(e) de lire votre fichier `nouveau` depuis sa machine.
3. Essayez d'accéder au répertoires personnels d'autres étudiants et de consulter leurs fichiers.

## Processus.

Dans ce paragraphe, nous allons utiliser la commande `top`. Consultez la documentation sur celle-ci.

**Exercice 8** (Liste actualisée des processus.).

1. Lancez la commande `top`. Dans quel ordre les processus affichés sont-ils classés ?
2. Quand `top` est actif, un appui sur la touche `[u]` permet de sélectionner uniquement les processus appartenant à un utilisateur donné. Affichez uniquement vos processus.
3. Cet utilitaire offre un grand nombre de fonctionnalités, comme celle de modifier la colonne de tri, sélectionner les colonnes à afficher, etc. Explorez la page de manuel et testez certaines de ces options.

**Exercice 9** (Tuer un processus avec `style`.).

Il existe plusieurs techniques pour terminer des processus comme appuyer sur le bouton « fermer » dans le cas d'un processus fenêtré). Nous allons en voir une qui utilise Unix.

1. Lancez deux ou trois processus quelconques depuis le terminal ou l'interface graphique. Dans un terminal, lancez la commande `top`. La touche `k` vous permet d'indiquer que vous souhaitez terminer un processus. Il vous est ensuite demandé de désigner un processus par son identificateur, puis de spécifier un signal à lui envoyer. Terminez ainsi les programmes que vous venez de lancer.
2. Dans certains cas, il est plus aisé de désigner un processus non pas par son identifiant mais en pointant une fenêtre appartenant au processus avec la souris. Pour ce faire, il existe la commande `xkill`. Lancez la commande `vi` depuis un terminal. Pour information, `vi` est un traitement de texte dont les commandes ne sont pas intuitives au premier abord. Ouvrez un autre terminal et utilisez `xkill` pour quitter le processus `vi`

## Communication entre commandes.

Consultez votre aide-mémoire sur les filtres et les pipes.

**Exercice 10** (Rediriger l'entrée ou la sortie d'un processus.).

1. Faites la liste détaillée (option `-l`) du contenu du répertoire `/usr/lib`.
2. Redirigez la sortie de la commande précédente vers un fichier `liste`, puis affichez le contenu de ce fichier avec la commande `less`.
3. Redirigez l'entrée standard de la commande `cat` vers le fichier `liste`. Quelle est la différence avec la commande précédente ? Comme pour un certain nombre de commandes, le même résultat peut être obtenu en passant le fichier `liste` directement en argument de la commande `cat` (sans symbole de redirection). Essayez.

4. Qu'y a-t-il dans le fichier `liste` si l'on exécute maintenant la commande :  
`echo une phrase > liste` ?
5. La syntaxe `>> fic` a un comportement différent. Testez-la comme dans la question précédente et concluez.
6. Sans utiliser `emacs`, ajoutez au fichier `liste` la ligne "*C'était vraiment très intéressant*".

**Exercice 11** (Erreur standard et sortie standard.).

Nous avons tester les commandes permettant de rediriger l'entrée ou la sortie standard d'un processus vers des fichiers quelconques. Pour rediriger la sortie standard d'erreur, il faut utiliser une syntaxe explicite faisant apparaître son numéro de descripteur, autrement dit la syntaxe `2>`.

1. Choisissez un nom de fichier qui n'existe pas (ici, `rEp3`) et testez la commande `ls rEp3`. Que se passe-t-il ?
2. Prédisez le résultat de la commande `ls rEp3 > toto` puis exécutez-la. Que se passet-il ? Que contient le fichier `toto` ?
3. Comment modifier la ligne précédente pour que le message d'erreur s'écrive dans le fichier `erreurs` ?
4. Ecrivez une ligne de commande qui liste le répertoire courant et un répertoire inexistant (par exemple `rEp3`) et stocke la sortie standard dans `liste` et les erreurs dans `erreurs`. Vérifiez le résultat.
5. Pour rediriger la sortie d'erreur et la sortie standard vers le même fichier `fic`, on utilise la syntaxe abrégée `&& fic2`. Redirigez les deux canaux de sortie de la commande précédente vers le fichier `liste`, affichez le contenu de ce fichier et expliquez le résultat.

L'entrée et la sortie standard peuvent être dirigées non pas vers un fichier, mais vers un autre programme. Pour effectuer la redirection, on utilisera un pipe noté `|`. Par exemple, la ligne `cat /etc/services | grep 25 | more` permet d'afficher toutes les lignes de `/etc/services` contenant le mot `25`.

**Exercice 12** (Tubes).

1. Placez-vous dans votre répertoire `unix`. En utilisant un tube, combinez les commandes `tar` et `gz` pour archiver et compresser vos répertoires `tp`.
2. A l'aide d'un tube, décompressez et désarchivez l'archive `tp.tar.gz`.

## Liens.

Sous Unix, on peut lier deux fichiers. Il existe deux sortes de liens physiques et symboliques. La commande `ln` permet de les créer. Allez voir sa syntaxe dans votre aide mémoire ou dans le manuel en ligne.

Dans les exercices suivants, nous allons étudier la différence entre les liens physiques et les liens symboliques.

### Exercice 13 (Expériences physiques).

1. Dans le répertoire `tp3` créez un fichier `fic`. Placez-vous dans votre répertoire maison (`~`).
2. Donnez trois façon de désigner le fichier `fic` depuis `~`.
3. Avec la commande `ln`, créez dans votre répertoire maison, un lien physique `lfic` vers le fichier `fic`.
4. A l'aide de la commande `echo` et d'une redirection de l'entrée standard, écrire une phrase à la fin du fichier `lfic`. Affichez dans le terminal le contenu du fichier `fic`. Recommencez en échangeant `lfic` et `fic`.
5. Dans un terminal, modifiez les droits d'accès au fichier `fic` pour les membres du groupe. Quels sont les droits du fichier `lfic`.

### Exercice 14 (Expériences symboliques).

1. Quelle option de `ln` vous permet de créer des liens symboliques ?
2. Dans votre répertoire maison, créez un lien symbolique `sfic` vers le fichier `fic`.
3. Dans `xemacs`, ouvrez le fichier `sfic` et observez le message dans le mini-buffer.
4. A l'aide de la commande `ls -l` comparez les informations concernant les deux fichiers lien.
5. Essayez de modifier les droits d'accès au fichier `sfic`.
6. Modifiez les droits d'accès au répertoire `rep` pour ne plus y avoir accès. Essayez d'afficher le contenu de `lfic` et de `sfic`. Que constatez-vous ?
7. Nettoyez votre répertoire maison en effaçant le répertoire `rep` et son contenu.
8. Concluez en imaginant les différentes utilisations des liens.

## Pour ceux qui pensent avoir fini.

### Exercice 15 (Filtres et tubes usuels).

Lisez les pages de man des commandes `wc`, `sort`, `cat`, `uniq` et répondez aux questions suivantes :

1. A quoi servent ces fonctions ?
2. Que doit on faire avant d'utiliser `uniq` ?
3. Quelles sont les enchaînements de commandes (une seule ligne) nécessaires pour effectuer les opérations suivantes :
  - \* Comptez le total des lignes des fichiers dans `/etc`.
  - \* Comptez le nombre de lignes contenant 25 dans `/etc/services`.
  - \* Triez le fichier `/etc/hosts`.
  - \* Affichez le nombre de fichiers n'appartenant pas à root dans `/etc`.

### Exercice 16 (Encore des tubes).

1. Documentez-vous sur les commandes `sort` et `grep`
2. Que fait la ligne de commande `ls /usr/bin | grep em` ?
3. Triez le contenu de votre répertoire personnel (home directory) par ordre alphabétique.
4. En une seule commande composée, cherchez dans `/bin` tous les noms de fichier contenant la lettre `a` et triezy-les par ordre alphabétique inverse.

### Exercice 17 (César).

Le code de César est une méthode de « chiffrement » rudimentaire : il effectue un simple décalage circulaire de lettres<sup>3</sup>. Ainsi, si le décalage choisi est d'une seule lettre, « abricot » est codé par « bcsjdpu » et « Z » devient « A ». La commande `tr` permet d'effectuer un codage très simplement. À l'aide de la page de manuel de cette commande, chiffrez un fichier de votre choix en utilisant des redirections.

Dans les exercices suivant, on utilisera les commandes `chmod` et `umask`.

### Exercice 18 (Droits par défaut, réglage du masque.).

1. Définissez un `umask` très restrictif qui interdit à quiconque à part vous l'accès en lecture ou en écriture, ainsi que la traversée de vos répertoires. Testez sur un nouveau fichier et un nouveau répertoire.
2. Définissez un `umask` très permissif qui autorise tout le monde à lire vos fichiers et traverser vos répertoires, mais n'autorise que vous à écrire. Testez sur un nouveau fichier et un nouveau répertoire.
3. Définissez un `umask` équilibré qui vous autorise un accès complet et autorise un accès en lecture aux membres de votre groupe Unix. Testez sur un nouveau fichier et un nouveau répertoire.

**Exercice 19** (Droits en octal).

Les commandes `chmod` et `umask` permettent aussi d'utiliser une syntaxe de la forme `chmod nnn`, où chaque `n` est un nombre entre 0 et 7 (notation octale). Lisez le manuel de ces deux commandes, testez cette syntaxe et décrivez son fonctionnement dans votre résumé sur les droits.