

# TP 1 – Prise en main de l’environnement Unix

15 novembre 2006

Le but de ce premier TP est de commencer à vous familiariser avec l’environnement Unix.

## 1 Ouverture de session

Se connecter est évidemment la première chose à faire. Lorsque vous démarrez l’ordinateur, choisissez FreeBSD comme système d’exploitation. Le système démarre et affiche une fenêtre vous invitant à taper votre identifiant (*login*) et votre mot de passe (*password*), ce qui vous permet d’ouvrir une session. Si vous utilisez votre compte pour la première fois, il y a trois cas possibles :

- *Vous êtes inscrit administrativement et vous avez déjà activé votre compte.*  
Dans ce cas, il n’y aura rien à faire de spécial. Il vous suffira de rentrer votre identifiant (*login*) et votre mot de passe (*password*) lorsque le système d’exploitation sera lancé.
- *Vous êtes inscrit mais vous n’avez pas encore activé votre compte.*  
Pour activer votre compte, vous devez vous connecter sous le nom d’utilisateur *me* (sans mot de passe). Vous vous rendrez alors à l’aide d’un navigateur sur la page <http://portail.univ-paris7.fr> puis suivrez le lien «activer son compte». On vous demandera de rentrer les renseignements inscrits sur votre carte d’étudiant (nom, prénom, numéro d’étudiant et date de naissance) afin de vous identifier et de lancer la procédure d’activation de compte. Après deux ou trois minutes, vous redémarrerez la machine pour que l’activation soit effective.
- *Vous n’êtes pas encore inscrit (ou en attente de carte).*  
Faites appel au début du TP à l’enseignant responsable de votre groupe afin qu’il vous crée un compte temporaire avec lequel vous pourrez faire les premières séances de TP. Par la suite, il sera important d’activer votre compte définitif dès que possible.

## 2 Découverte de l’interface graphique

Une fois votre nom d’utilisateur et mot de passe vérifiés, le gestionnaire de bureau, le programme qui gère l’affichage des menus et des fenêtres (ici KDE) apparaît. Le menu principal (bouton K en bas à gauche) permet de lancer des applications, d’accéder aux outils de paramétrage du système ou encore de fermer la session ou d’éteindre l’ordinateur.

**Note :** la plupart des programmes ainsi que le menu principal disposent d’une entrée «aide» (*help*) permettant d’accéder à l’aide en ligne. Si vous êtes coincé(e), n’hésitez pas à la consulter.

### Exercice 1 – *Éléments du bureau.*

Explorez un peu le menu principal. Lancez quelques applications, et exercez-vous à agrandir, fermer et déplacer les fenêtres. Parcourez l'ensemble des menus disponibles et essayez de deviner à quoi sert chaque fonction.

Avec le bouton droit de la souris, vous pouvez faire apparaître différentes commandes selon l'endroit cliqué (on parle de menus «contextuels»). Essayez à quelques endroits.

### Exercice 2 – *Gestionnaire de fichiers.*

Ouvrez le gestionnaire de fichiers de KDE. Explorez l'arborescence du système de fichiers, en repérant en particulier le répertoire racine ainsi que votre répertoire personnel, ainsi que les principaux menus.

Créez un répertoire appelé `essai` dans votre répertoire personnel, puis renommez-le en `test`.

### Exercice 3 – *Créer un fichier texte.*

Lancez un éditeur de texte (*emacs* ou *kwrite* par exemple), et créez un nouveau fichier. Rédigez un texte d'au plus 4 lignes vous présentant. Sauvegardez le fichier dans votre répertoire personnel dans le répertoire `test` que vous avez créé, sous le nom `presentation.txt`.

Fermez le fichier. Ouvrez-le à nouveau et rajoutez-y le texte «Étudiant en Licence 1 dans le groupe X à l'Université de Paris 7 ». Sauvegardez puis fermez l'éditeur de texte.

### Exercice 4 – *Messages électroniques.*

Lancez le navigateur web (*konqueror*), et pointez-le sur votre «espace numérique de travail» à l'université (<http://portail.univ-paris7.fr/>) ou sur un autre service de messagerie électronique si vous en disposez. Identifiez-vous grâce à vos nom d'utilisateur et mot de passe.

Sous la rubrique «messagerie électronique» (*webmail*), créez un nouveau message et copiez-y le texte du fichier que vous avez rédigé (ou attachez-le en pièce jointe), puis envoyez le tout à votre chargé de TP, avec un petit mot gentil en plus.

## 3 Découverte du *shell*

Une autre façon d'interagir avec le système est d'utiliser un *terminal* (ou *console*) dans lequel vous pourrez taper des commandes (qui ne sont rien d'autre que des programmes). Le programme avec lequel vous interagissez pour exécuter les commandes s'appelle le *shell*. Celui-ci se contente d'attendre que vous lui demandiez d'exécuter une commande pour le faire. Vous lui indiquez que vous avez fini de taper une commande en appuyant sur la touche `enter`.

Le *shell* vous indique qu'il est prêt en affichant une invite (ou *prompt*), en général le caractère dollar (\$) ou supérieur (>), en début de ligne. On peut distinguer diverses sortes de commandes :

- des commandes « simples » permettant souvent de manipuler des fichiers sur le disque dur et qui affichent leur résultat directement dans le shell (`ls`, `cp`, ...). Ces commandes, puisqu'elles utilisent le shell pour leurs interactions, ne peuvent être exécutées en dehors d'un terminal,
- des commandes plus sophistiquées qui n'utilisent pas le shell (`konqueror`, `emacs`, ...). Ces programmes peuvent également être exécutés à l'aide d'une icône de l'interface graphique,
- des commandes internes au shell (*built-in commands*) qui permettent d'utiliser certaines facilités offertes par le shell (`alias`, `exit`, ...).

## Les premières commandes

`whoami` : Commençons par une commande simple qui vous renvoie votre identifiant.

```
> whoami
monit
```

`who` : Permet de connaître la liste de personnes actuellement connectées sur la machine.

```
> who
monit pts/2      Sep 22 11:40 (pps.jussieu.fr)
dupont pts/5      Sep 16 22:20 (fbx.proxad.net)
matin  pts/6      Sep 22 10:28 (abo.wanadoo.fr)
:
```

Évidemment, pour l'instant vous êtes seul sur votre machine ! À noter que la commande `w` (pour *what*) est plus complète et vous dit plus précisément qui fait quoi sur la machine.

`man` : Le manuel en ligne pour toutes les commandes accessibles depuis le terminal. Il suffit de taper `man ma_commande` pour accéder à la description complète de *ma\_commande*.

### Exercice 5 – Qui, quoi ...

Testez les commandes `whoami`, `who` et `w` et observez les informations affichées.

### Exercice 6 – ...et comment.

Ouvrez la page de manuel de la commande `who` en tapant `man who`. Vous pouvez faire défiler le texte avec les flèches haut et bas, ou les touches *page up* et *page down*. Pour sortir, pressez la touche `q`.

`cat`, `less` : La commande `cat mon_fichier` permet d'afficher en une seule fois le contenu entier du fichier texte `mon_fichier`.

Pour des textes plus longs, on utilise `less`, dont l'interface est la même que celle de `man`.

### Exercice 7 – Nom relatif, nom absolu.

En utilisant la commande `cat`, affichez de deux façons différentes le contenu de votre fichier `presentation.txt` créé à l'exercice 3 (c'est à dire avec le nom absolu du fichier, puis avec son nom relatif).

### Exercice 8 – Interrompre un programme, le symbole `&`.

Attention, exercice **utile** !

1. Ouvrez le fichier texte `presentation.txt` avec l'éditeur de votre choix en tapant `kwite nom du fichier` ou `emacs nom du fichier`.
2. Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande (par exemple `whoami`). Que se passe-t-il ?
3. Toujours sous le terminal, gardez la touche `Ctrl` enfoncée puis pressez et relâchez la touche `c`. Que se passe-t-il ?
4. Répétez les étapes 1 et 2, en ajoutant cette fois le caractère `&` en fin de commande (soit par exemple `kwite nom du fichier &`). Voyez-vous une différence ?

## Fichiers et répertoires

A l'aide de quelques commandes de base, vous serez en mesure d'effectuer toutes les opérations utiles sur le système de fichiers (parcours, copie, déplacement, etc.) par l'intermédiaire de la ligne de commande.

`pwd` : Cette commande, utilisée sans argument, vous indique dans quel répertoire vous vous trouvez actuellement (répertoire courant, ou *working directory*). A l'ouverture d'un nouveau terminal, le répertoire courant est toujours votre répertoire personnel.

`cd` : La commande `cd rep` change le répertoire courant en le répertoire `rep`. Elle permet de se déplacer dans l'arborescence. Utilisée seule ou avec l'argument `~`, elle vous ramène dans votre répertoire personnel. Utilisée avec l'argument `..`, elle vous fait remonter d'un niveau dans l'arborescence (au répertoire parent).

`ls` : Cette commande affiche le contenu du répertoire courant, fichiers et sous-répertoires.

### Exercice 9 – Tour du propriétaire.

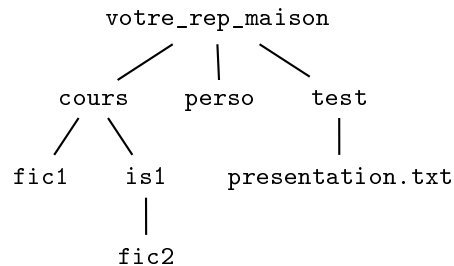
En utilisant les trois commandes ci-dessus, promenez-vous un peu dans l'arborescence en affichant le contenu de quelques répertoires. Si vous êtes perdu(e), tapez simplement `cd` pour rentrer chez vous.

`mkdir` : La commande `mkdir rep` crée un répertoire vide *rep*. Si ce répertoire existe déjà vous obtiendrez bien sûr une erreur.

`touch` : Taper la commande `touch fic` permet de créer un fichier vierge (de taille 0) appelé *fic*.

#### Exercice 10 – Créer une arborescence

À partir de votre répertoire maison reproduisez l'arborescence suivante :



Un moyen simple de contrôler que vous n'avez rien oublié est de taper `ls -R` depuis votre répertoire personnel. Dans cette commande, `-R` est appelée une *option*. Pouvez-vous expliquer son rôle ?

`cp` : La commande `cp fic1 fic2` crée un nouveau fichier identique à *fic<sub>1</sub>* appelé *fic<sub>2</sub>*. On peut aussi écrire `cp fic1 ... ficn dest` qui copie les fichiers *fic<sub>1</sub>* à *fic<sub>n</sub>* dans le répertoire *dest*.

#### Exercice 11 – Copie et suppression de fichiers

Créez une copie de `presentation.txt` appelée `.plan` (le point est voulu) *dans votre répertoire personnel*.

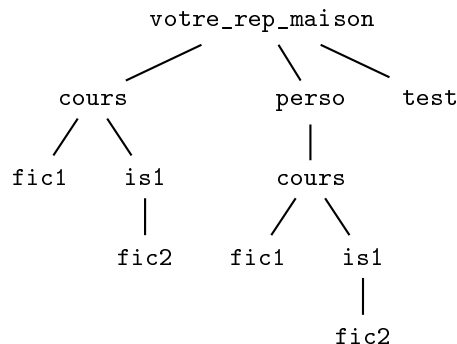
Supprimez ensuite le fichier `presentation.txt` à l'aide de la commande `rm`.

#### Exercice 12 – Fichiers cachés

Affichez le contenu de votre répertoire personnel avec `ls`. Que constatez-vous ? Réessayez en ajoutant l'option `-a` à `ls`. Concluez.

#### Exercice 13 – Copie de répertoires

Par défaut `cp` ne copie que des fichiers. L'option `-r` permet de forcer `cp` à copier les répertoires et sous-répertoires de ce qui lui est donné en argument (autrement dit à copier *récurivement*). Utilisez la pour obtenir l'arborescence suivante :



#### Exercice 14 – Suppression de répertoires avec `rmdir`

À partir de votre répertoire maison, tentez de détruire le répertoire `test` puis la nouvelle copie du répertoire `cours` à l'aide de `rmdir`. Que se passe-t-il ? Remédiez au problème.

`mv` : La commande `mv` fonctionne de manière similaire à `cp`, mais elle déplace les fichiers concernés plutôt que de les recopier.

#### Exercice 15 – Déplacement de fichiers et répertoires.

Expérimentez la commande `mv` en déplaçant des fichiers dans votre arborescence. Essayez également de déplacer un répertoire. Que doit-on faire pour que cela fonctionne ?

## 4 Aller un peu plus loin

### Manipuler des ensembles de fichiers

Il existe un certain nombre de symboles spéciaux appelés *Jokers* ou *wildcards*, qui permettent de désigner plusieurs fichiers à la fois. Il en existe plusieurs, mais les principaux sont :

- \* : dans un nom de fichier, représente n'importe quelle suite de zéro, un ou plusieurs symboles ;
- ? : représente exactement un symbole quelconque.

Ainsi, écrire `tp?.ps` désigne `tp1.ps`, `tpi.ps` mais pas `tp-is1.ps`, et `truc*` désigne `truc`, `trucage`, `trucmuche` mais pas `struck`.

#### Exercice 16

1. Affichez la liste de tous les fichiers dans le répertoire `/usr/bin` dont le nom commence par `k` et contient exactement 6 caractères.
2. Affichez la liste de tous les fichiers dont l'extension est `.so` dans le répertoire `/usr/lib` (*note* : ces fichiers sont des bibliothèques).

## Utiliser l'historique

Il arrive parfois que l'on aie à utiliser une même commande plusieurs fois, ou que l'on souhaite corriger une commande tapée précédemment. A cette fin, sur la plupart des *shells* modernes il existe un ensemble de «raccourcis» permettant d'accéder à l'historique des commandes.

### Exercice 17 – Naviguer dans l'historique

En utilisant les flèches haut et bas, on peut faire défiler les commandes précédentes, de la plus récente à la plus ancienne. Utilisez ce mécanisme pour tester les commandes de l'exercice précédent dans d'autres répertoires.

Avec les flèches gauche et droite, vous pouvez également corriger un détail de chaque ligne de commande, ce qui peut être plus rapide que de retaper une longue commande presque identique.

### Exercice 18 – Visualiser l'historique

La commande `history` affiche une liste des commandes récentes, chacune précédée d'un numéro. En tapant `! n` où `n` est un numéro dans l'historique, on rappelle la commande correspondante.

## Rechercher de nouvelles options

Les exercices suivants nécessitent la lecture du manuel avec la commande `man`. Essayez de prendre l'habitude de trouver par vous-même les informations dont vous avez besoin.

### Exercice 19 – Rendre le listing plus lisible.

Par défaut, le résultat de la commande `ls` est un peu aride. Une option permet de colorer l'affichage pour repérer plus rapidement le type de fichiers présents dans un répertoire. Quelle est cette option de `ls` ?

### Exercice 20 – Suppression de répertoires avec `rm`.

Quelle est l'option de `rm` permettant d'effacer *récurivement* (c'est à dire en descendant dans les sous-répertoire) un répertoire ? Appliquez-la sur le répertoire `rep3`.

### Exercice 21 – Confirmation avant suppression.

Par défaut, `rm` ne demande pas de confirmation lorsque vous tentez de supprimer un fichier. Ceci peut se révéler assez dangereux. Trouvez l'option qui permet de demander confirmation.

## 5 Personnaliser son environnement

### Les alias

Certaines commandes sont plutôt longues à taper, notamment lorsqu'il y a des options. Une commande permet de palier à cet inconvénient en créant ses propres commandes : `alias`.

Utilisation : `alias ma_commande=commande_complète`. Attention, si `commande_complète` contient des espaces (par exemple, s'il y a des options), il faut alors l'encadrer avec des guillemets.

#### Exercice 22 – Mes propres commandes

Créez une commande `la` qui liste tous les fichiers d'un répertoire (y compris les fichiers cachés commençant par un point).

`alias` permet aussi de donner une autre signification à une commande existante. On utilise en général cette caractéristique pour rajouter des options systématiquement à une commande

#### Exercice 23 – Changez une commande

Changez le fonctionnement de la commande `rm` pour qu'elle demande systématiquement confirmation lors d'une suppression. Changez ensuite le fonctionnement de la commande `ls` afin d'utiliser systématiquement l'affichage coloré.

À noter que la commande `alias` sans argument liste tous les raccourcis.

#### Exercice 24 – Effectivité des changements

Testez les raccourcis que vous venez de créer. Puis ouvrez un nouveau terminal et testez-les dans celui-ci. Que constatez vous ?

### Les fichiers de configurations

Le problème avec la commande `alias` est que dès qu'on se déconnecte, tout est à refaire. Pour stocker ce genre de paramètres une bonne fois pour toute, on utilise des *fichiers de configurations* liés au shell. Pour le shell `bash`, ces fichiers sont `.bash_profile` et `.bashrc`. Ce sont des fichiers texte contenant des commandes qui sont lues au moment de l'ouverture d'un terminal. Tout changement dans ces fichiers ne sera donc pris en compte que si on relance le shell (via l'ouverture d'un nouveau terminal par exemple).

#### Exercice 25 – Mes propres fichiers de configuration

Récupérez tout d'abord les deux fichiers `.bash_profile` et `.bashrc` situés dans le répertoire `/users/monitIS1/` en les copiant dans votre répertoire maison. Éditez ensuite le fichier `.bashrc` avec Emacs pour rajouter les commandes des exercices précédents (`ls` avec couleur, `rm` avec confirmation et `la`) et personnaliser vos sessions avec les exemples vus en cours (message de bienvenue, affichage du répertoire, ...). Pensez à ouvrir un nouveau terminal afin de rendre les changements effectifs.



### Exercice 26 – En savoir plus sur les utilisateurs.

La commande `finger machin` permet d'afficher le nom ainsi que plusieurs statistiques de l'utilisateur *machin*.

Avec le fichier `.plan` toujours dans votre répertoire personnel, lancez cette commande avec votre propre nom d'utilisateur (ou sans argument si cela ne fonctionne pas). Que constatez-vous ? Que peut-on en déduire au sujet du fichier `.plan` ?

## 6 Se déconnecter

Pour quitter l'environnement graphique, il faut d'abord avoir quitté toutes les applications (`C-x C-c` dans Emacs et `exit` ou `C-d` dans un terminal). Ensuite, vous pouvez éteindre l'ordinateur à partir du menu de l'interface graphique.

**NB :** Unix, comme tout système moderne, doit être arrêté proprement et non pas en éteignant physiquement la machine.