

Introduction aux systèmes d'exploitation (IS1)

TP 1 – Prise en main de l'environnement Unix

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

1 Ouverture de session FreeBSD

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 rendez alors à l'aide d'un navigateur sur la page

<http://portail.univ-paris7.fr>

puis suivez 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 votre session pour que l'activation soit effective.

- Vous n'êtes pas encore inscrit (ou en attente de carte).

Utilisez le compte *me* provisoirement, en attendant la réception de votre carte. Attention, le compte *me* ne permet pas de sauvegarder vos fichiers (mais ce n'est pas un problème pour la séance d'aujourd'hui, étant donné qu'il n'y a rien à sauvegarder).

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 »).

Exercice 2 – Créer un fichier texte

Lancez l'éditeur de texte *xemacs* et créez un nouveau fichier. Rédigez un (petit) texte vous présentant (vos nom, prénom, votre scolarité passée, vos connaissances en informatique, ce que vous aimeriez être en train de faire à l'instant présent, etc...). Sauvegardez le fichier sous le nom "presentation.txt", dans un répertoire appelé *test* que vous aurez créé dans votre *répertoire maison* (aussi appelé *répertoire personnel* ou *répertoire de login*).

(Si vous utilisez le compte *me*, créez d'abord un répertoire portant votre nom dans le répertoire maison. Vous travaillerez jusqu'à la fin du tp, dans ce répertoire.)

Autrement dit, donnez à votre fichier le chemin absolu `/home/toto/test/presentation.txt` (en remplaçant *toto* par votre login), ou, si vous utilisez le compte *me*, le chemin absolu `/home/me/toto/test/presentation.txt` (en remplaçant *toto* par votre nom).

Par la suite, nous pourrions parler de nom de fichier pour parler d'un chemin (absolu ou relatif) de ce fichier.

Exercice 3 – 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 (attention, le service de messagerie *hotmail* est considéré comme trop "hot" pour le filtre des serveurs du SCRIPT, et n'est, par conséquent, pas accessible depuis les salles de tp). Identifiez-vous grâce à vos nom d'utilisateur et mot de passe.

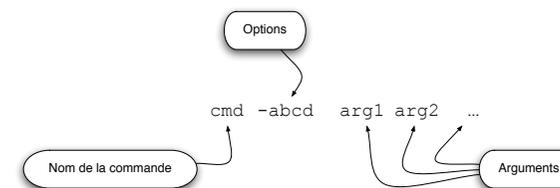
Sous la rubrique « messagerie électronique » (*webmail*), créez un nouveau message et copiez-y le texte du fichier que vous avez rédigé (ou attachez-le en pièce jointe), puis envoyez le tout à votre chargé(e) de TP. Cela lui permettra d'une part, de mieux vous connaître, et d'autre part, d'avoir la liste de vos adresses électroniques s'il (ou elle) a besoin de vous faire part d'informations en ce qui concerne le cours.

3 Découverte du shell

Une autre façon d'interagir avec le système (et la seule que l'on utilisera dans ce cours) est d'utiliser un *terminal* (ou *console*) dans lequel vous pouvez 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 Entrée (*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.

Pour affiner le comportement d'une commande, on peut lui adjoindre ce que l'on appelle des *options*, représentées par un tiret suivie d'une ou plusieurs lettre(s). On peut également utiliser certaines commandes avec des *arguments*. La forme générale d'une ligne de commande est souvent (mais pas toujours) la suivante :



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, xemacs, etc.). 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, etc.).

Les premières commandes

Commençons par la commande la plus utile :

man : Le manuel en ligne pour toutes les commandes accessibles depuis le terminal. Il suffit de taper `man cmd` pour accéder à la description complète de la commande `cmd`.

Les pages de manuel sont très complètes, et détaillent chacune des options disponibles pour une commande donnée. N'hésitez pas non plus à aller glaner des informations sur Internet via votre moteur de recherche favori. Intéressons-nous maintenant à deux commandes relativement simples :

whoami : Une commande qui vous renvoie votre identifiant.

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

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.

Exercice 4 – Qui, quoi...

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

Exercice 5 – ...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, `less` permet de l'afficher page par page.

Exercice 6 – Chemin relatif, chemin absolu

En utilisant la commande `cat`, affichez de deux façons différentes le contenu de votre fichier `presentation.txt` créé à l'exercice 2 (c'est-à-dire avec le chemin absolu du fichier, puis avec un de ses chemins relatifs).

Exercice 7 – Interrompre un programme, le symbole &

1. Ouvrez à nouveau votre fichier texte `presentation.txt` en tapant dans le terminal la ligne de commande : `xemacs presentation.txt`
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 `z`. 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 `xemacs presentation.txt &`). Voyez-vous une différence ?

Fichiers et répertoires

À 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 du shell. Et vous verrez que cela s'avère souvent bien plus rapide que d'utiliser l'interface graphique.

pwd : Utilisée sans argument, vous indique dans quel répertoire vous vous trouvez actuellement (répertoire courant, ou *working directory*).

À 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 : Affiche le contenu du répertoire courant : ses fichiers et ses sous-répertoires.

Exercice 8 – Tour du propriétaire

En utilisant les trois commandes ci-dessus, promenez-vous un peu dans l'arborescence en affichant le contenu de quelques répertoires. Vous pouvez, par exemple, commencer par repérer la localisation du répertoire `mon_tIS1` (ce qui vous sera utile par la suite), et afficher son contenu. (Souvenez-vous de l'arborescence vue dans le cours précédent).

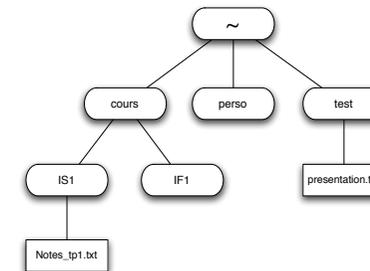
Si vous êtes perdu(e), tapez simplement `cd` pour rentrer chez vous.

mkdir : La commande `mkdir rep` crée un répertoire vide `rep`. Si un répertoire de même nom existe déjà à cet emplacement, vous obtiendrez bien sûr une erreur.

Exercice 9 – Créer une arborescence

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

(Si vous utilisez le compte `me`, créez d'abord un répertoire de nom votre login, puis créez l'arborescence suivante dans ce répertoire.)



Remarque : nous ne vous fournirons pas toujours de corrigé, et nous vous conseillons donc de noter vous-même ce qui vous a paru important pour chaque tp (dans un fichier intitulé `Notes_tp1.txt`, par exemple).

Un moyen simple de contrôler que vous n'avez rien oublié est de taper `ls -R` depuis votre répertoire personnel. Pouvez-vous expliquer le rôle de l'option `-R` ?

`mv` : La commande `mv fic1 fic2` attribue au fichier (ou répertoire) dont le chemin était jusque là `fic1` dans l'arborescence, le nouveau chemin `fic2`. Après exécution de la commande, `fic2` est un chemin de l'arborescence qui mène au fichier (ou répertoire) mais plus `fic1`.

Exercice 10 – Déplacement de fichiers dans l'arborescence

Changez le chemin de votre fichier `presentation.txt` en (le chemin relatif) : `~/plan` (sans omettre le point).

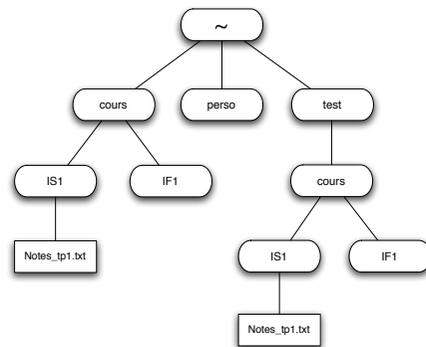
Exercice 11 – Fichiers cachés

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

`cp` : La commande `cp fic1 fic2` attribue (en plus) au fichier (ou répertoire) dont le chemin était jusque là `fic1` dans l'arborescence, le nouveau chemin `fic2`. Après exécution de la commande, `fic1` et `fic2` sont deux chemins de l'arborescence qui mènent au fichier (ou répertoire) .

Exercice 12 – 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 :



`rm` : La commande `rm fic` efface le fichier `fic` (nous verrons plus tard qu'il est plus juste de dire qu'elle supprime le chemin `fic` de l'arborescence).

`rmdir` : La commande `rm rep` efface le répertoire `rep` (même remarque que précédemment, le chemin `rep` est supprimé de l'arborescence).

Exercice 13 – Suppression de répertoires

À partir de votre répertoire maison, tentez de détruire le répertoire `test` puis la nouvelle copie du répertoire `cours` à l'aide de la commande `rmdir`. Que se passe-t-il ? Comment faut-il faire pour supprimer le répertoire `test` à l'aide de la commande `rmdir` ?

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`, `trucmucho` mais pas `struck`.

Exercice 14

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

Utiliser l'historique

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

Exercice 15 – 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 16 – 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 17 – 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 18 – Suppression de répertoires avec rm

Quelle est l'option de `rm` permettant d'effacer *récurivement* (c'est à dire en descendant dans les sous-répertoires) un répertoire ? Appliquez-la sur la copie du répertoire `cours` que vous aviez placée dans le répertoire `test`.

Exercice 19 – 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.

Archivage

Il peut arriver de vouloir regrouper plusieurs fichiers en un seul, par exemple pour l'envoi de pièces jointes par courrier électronique, ou pour la mise à disposition d'un ensemble de fichiers sur internet. On dispose pour cela d'un outil appelé `tar` (pour *tape archive*, cet outil était auparavant destiné à l'archivage sur bande magnétique). On utilise le suffixe `.tar` pour distinguer les archives.

```
tar : Commande servant à archiver plusieurs fichiers en un seul.
```

Exemples d'utilisation :

```
tar -cvf toto.tar fic1 fic2 ... : archive les fichiers fic1 fic2... dans toto.tar
tar -xvf toto.tar : extrait dans le répertoire courant les fichiers contenus dans toto.tar
```

Exercice 20 – Création d'une première archive

Créez, à l'aide d'`xemacs` quatre fichiers : `fic1` contenant le mot "Ceci", `fic2` contenant le mot "est", `fic3` contenant le mot "une", et `fic4` contenant le mot "archive". Archivez ces quatre fichiers dans une archive que vous nommerez `test.tar`.

Examinez le contenu de l'archive à l'aide de la commande `cat`. Que constatez-vous ?

Exercice 21 – Archivage d'une arborescence

La commande `tar` permet également d'archiver des répertoires entiers directement. Créez une archive que vous appellerez `cours.tar` contenant votre répertoire `cours`. Examinez son contenu à l'aide de la commande `cat`. Même si cela reste lisible, cela commence à être inefficace de lire directement le fichier archive. Cherchez dans la page de manuel de `tar` l'option permettant d'examiner le contenu d'une archive sans lire directement le contenu du fichier archive, et sans extraire cette archive.

5 Personnaliser son environnement

Les alias

Certaines commandes sont plutôt longues à taper, notamment lorsqu'il y a des options. Une commande permet de pallier 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 qu'elle n'agit que dans le terminal dans lequel on est en train d'écrire : dès qu'on se déconnecte, tout est à refaire. Pour stocker ce genre de paramètres une bonne fois pour toutes, 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

Éditez le fichier `.bashrc` situé dans votre répertoire personnel¹ avec `xemacs` pour rajouter les commandes des exercices précédents (`ls` avec couleur, `rm` avec confirmation et `la`). Pensez à ouvrir un nouveau terminal afin de rendre les changements effectifs. Mais faites attention à ne pas effacer de fichiers importants si vous voulez tester votre nouvelle version de `rm` !

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 `xemacs` 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.

¹ou bien créez-en un nouveau s'il n'existe pas