



Initiation UNIX – Informatique première année

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

Chaque utilisateur est identifié par une procédure de login : nom d'utilisateur (username) valide suivi du mot de passe (password) correspondant. C'est l'administrateur système qui assigne le nom d'utilisateur et le mot de passe initial. Chaque station a un nom de machine (hostname). L'Université Paris-Est Marne-la-Vallée utilise des machines sous Linux (Debian).

Se connecter est évidemment la première chose à faire. 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.

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 Gnome) apparaît. Le menu principal (boutons Applications, Raccourcis et Système en haut 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 – Créer un fichier texte

A l'aide du menu principal, lancez l'éditeur de texte *emacs22* 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*».

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-Est Marne-la-Vallée ». Sauvegardez puis fermez l'éditeur de texte.

Exercice 3 – Messages électroniques

Lancez le navigateur web (*firefox*, aussi appelé *iceweasel*), et pointez-le sur votre « espace numérique de travail » à l'université (<https://ent.univ-mlv.fr/>) 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 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*. Lorsque vous avez fini de taper une commande appuyez sur la touche Entrée ou *enter* pour demander au *shell* de l'exécuter.

Exercice 4

En utilisant le menu des applications ouvrez un terminal.

Le terminal 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* (*iceweasel*, *emacs*, 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

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

```
> whoami
monit
```

`who` : Permet de connaître la liste des 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)
martin   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 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, mais peuvent paraître obscures au premier abord. N'hésitez pas non plus à aller glaner des informations sur Internet via votre moteur de recherche favori.

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, `less` permet de l'afficher page par page.

Exercice 7

En utilisant la commande `cat`, affichez le contenu de votre fichier `presentation.txt` créé à l'exercice 2.

Exercice 8

Créez et ouvrez un fichier `notes` à l'aide de votre éditeur de texte. Affichez le manuel de `cat` dans le terminal. À l'aide de la souris copiez l'intégralité cette page de manuel dans le fichier `note`. Testez les commandes `cat` et `less` sur le fichier `notes`.

Fichiers et répertoires

Comme vous l'avez vu en cours, les données sont stockées sous forme de fichiers qui sont organisés de manière arborescente à partir du répertoire racine, désigné par le symbole `«/»`. (Ce symbole est aussi utilisé comme séparateur dans la description d'un chemin.)

Un **répertoire** (directory) est un type de fichier particulier qui peut contenir un ensemble de fichiers. Un répertoire contenant un sous-répertoire est dit père de celui-ci. Tout répertoire contient deux répertoires `.` (point) et `..` (deux points). Le répertoire `.` désigne le répertoire lui-même, le répertoire `..` désigne le répertoire père.

Un **fichier** est désigné par son chemin d'accès constitué de son nom précédé de celui des répertoires qui le contiennent. Le caractère `«/»` sert de séparateur entre ces noms. (Ainsi des fichiers différents peuvent avoir le même nom du moment qu'ils sont dans les répertoires différents.)

Le chemin `« »` désigne en général un raccourci vers votre répertoire personnel. D'une manière générale le chemin `« toto »` désigne le répertoire personnel de l'utilisateur `toto`.

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 de la ligne de commande.

`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` : Cette commande permet de se déplacer dans l'arborescence. On l'utilise en donnant comme argument le chemin du nouveau répertoire courant. Si le chemin donné comme argument n'est pas un chemin absolu le nouveau répertoire courant est calculé relativement au répertoire courant actuel.

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 9 – Tour du propriétaire

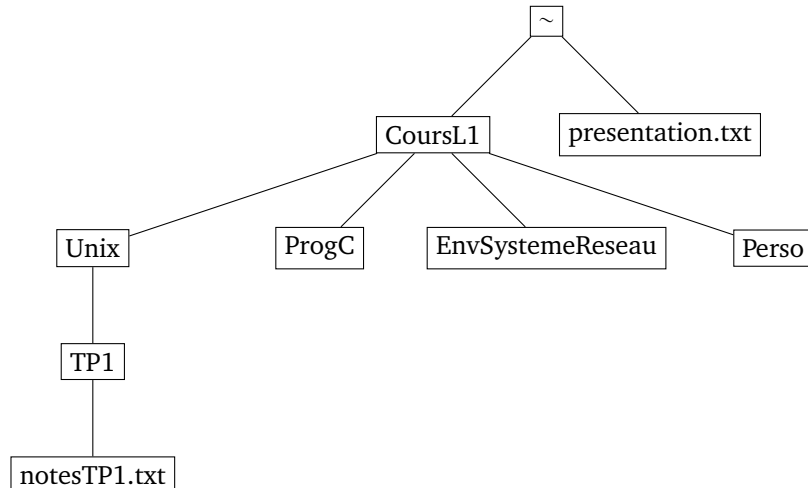
En utilisant les trois commandes ci-dessus, promenez-vous un peu dans l'arborescence en affichant le contenu de quelques répertoires.

Si vous êtes perdu(e), tapez simplement `cd` pour revenir à votre répertoire personnel.

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

Exercice 10 – Créer une arborescence

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



Remarque : nous ne vous fournirons pas de corrigé, et nous vous conseillons donc de noter vous même ce qui vous a paru important pour chaque tp (dans un fichier intitulé `notesTP1.txt`, par exemple).

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 ?

Exercice 11 – Nom relatif, nom absolu

Placez vous dans le répertoire Unix. En utilisant la commande `cat`, affichez de deux façons différentes le contenu du fichier `presentation.txt` créé à l'exercice 2 (c'est à dire avec le nom absolu du fichier, puis avec son nom relatif).

`cp` : La commande `cp fic1 fic2` crée un nouveau fichier identique à `fic1` appelé `fic2`. On peut aussi écrire `cp fic1 ... ficn dest` qui copie les fichiers `fic1` à `ficn` dans le répertoire `dest`.

`rm` : La commande `rm fic` efface le fichier `fic` (attention, utilisez cette commande avec précaution, une fois le fichier effacé, il ne sera plus possible de le récupérer).

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

Exercice 12 – déplacement de fichiers

Placez le fichier `presentation.txt` dans le répertoire Perso sans changer de répertoire courant. ■

Exercice 13 – Copie et suppression de fichiers

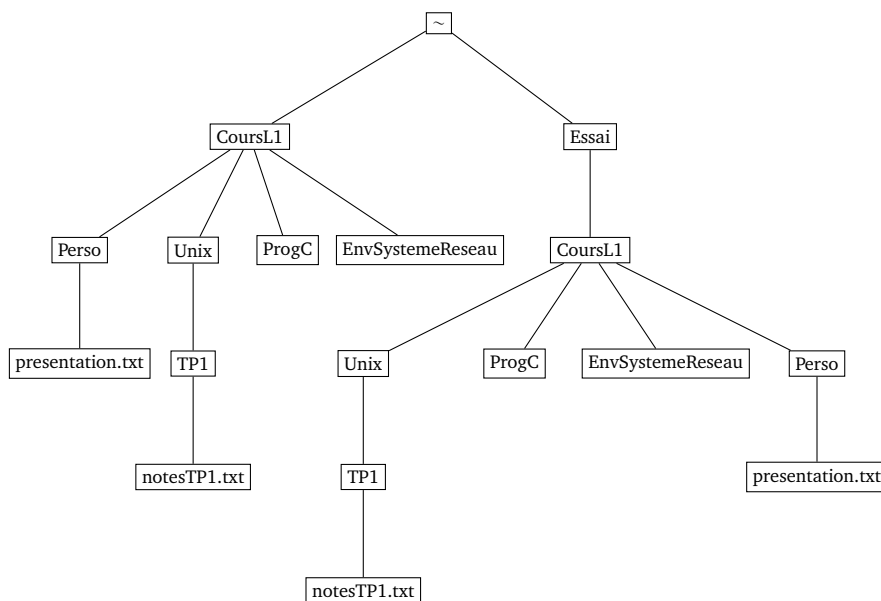
1. Créez une copie de `presentation.txt` appelée `.presentation` (le point est voulu) dans votre répertoire personnel. Supprimez ensuite le fichier `presentation.txt` à l'aide de la commande `rm`.
2. Comment auriez-vous pu faire cela en une seule ligne de commande ?

Exercice 14 – Fichiers cachés

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

Exercice 15 – 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 16 – Suppression de répertoires avec `rmdir`

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

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 17

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 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 18 – 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 19 – 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 20 – *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 21 – *Explorer l'arborescence en profondeur*

Par défaut, le résultat de la commande `ls` donne la liste des fichiers et répertoires contenus dans le répertoire courant. Trouvez comment afficher la liste complète des fichiers contenus dans le répertoire `CoursL1` y compris ceux des sous-répertoires et ce quelque soit le répertoire courant.

Exercice 22 – *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.

Exercice 23 – *Interrompre un programme, le symbole &*

1. Ouvrez à nouveau le fichier texte `presentation.txt` en tapant `emacs presentation.txt`.
2. Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande (par exemple `whoami`). Que se passe-t-il ? Fermez l'éditeur de texte.
3. Répétez les étapes 1 et 2, en ajoutant cette fois le caractère `&` en fin de commande (soit par exemple `emacs presentation.txt &`). Voyez-vous une différence ?
4. Placez vous dans le répertoire racine `/` et tapez `ls -R`. Que se passe-t-il ? Vous pouvez utiliser la commande `Ctrl + c`¹ pour arrêter une tâche en cours d'exécution.

5 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.

¹Gardez la touche `Ctrl` enfoncée, puis pressez et relâchez la touche `c`.