

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 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-paris-diderot.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 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 – Gestionnaire de fichiers

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

(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.)

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 (*gedit*, *kwrite* ou *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, dans 10 ans, etc. Sauvegardez le fichier sous le nom "`nom_prenom.txt`", dans votre répertoire `test`.

Autrement dit, vous êtes en train de donner à votre fichier le chemin absolu `/home/i denti fi ant/test/nom_prenom.txt`, ou, si vous utilisez le compte `me`, le chemin absolu `/home/me/i denti fi ant/test/nom_prenom.txt`.

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

Exercice 4 – Messages électroniques

Lancez le navigateur web (*firefox*), et pointez-le sur votre « espace numérique de travail » à l'université (<http://portail.univ-paris-diderot.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 ayant pour objet « [IS1] fiche de nom prénom » ; copiez-y le texte du fichier que vous avez rédigé et attachez-le en pièce jointe, puis envoyez le tout à votre chargé(e) de TP.

Exercice 5 – Le site du cours IS1

1. Inscrivez-vous au cours IS1 sur Didel :
 - allez sur <http://didel.script.univ-paris-diderot.fr/>
 - faites l'identification ENT (cadre en rouge à droite)
 - cliquez sur "S'inscrire à un cours"
 - allez dans Sciences/Département de sciences exactes/ et cliquer sur le crayon à droite en face d'IS1
2. Rattachez-vous à un groupe :
 - allez dans le module IS1
 - dans le menu à gauche, choisissez "Groupes"
 - cliquez sur le crayon en face du groupe voulu
3. Soumettez votre fiche de présentation :
 - allez dans le module IS1
 - dans le menu à gauche, choisissez "Travaux"
 - cliquez sur "Fiche de présentation"
 - cliquez sur votre groupe (qui doit apparaître en gras)
 - cliquez sur "Nouvelle soumission"
4. Modifiez votre soumission :
 - allez dans le module IS1
 - dans le menu à gauche, choisissez "Travaux"

- cliquez sur "Fiche de présentation"
- cliquez sur votre groupe (qui doit apparaître en gras)
- cliquez sur le crayon situé sous votre soumission et effectuez les modifications (ne créez pas de nouvelle soumission)

Ce site est à visiter régulièrement, vous pourrez y trouver les fiches de TP et d'autres informations utiles.

Exercice 6 – Le site du cours IF1

Répétez les questions 1 et 2 de l'exercice précédent pour le cours d'IF1 (51IF1IF1 - IF1 2012-2013 - Introduction à l'Informatique)

Exercice 7 – Le site du département de sciences exactes

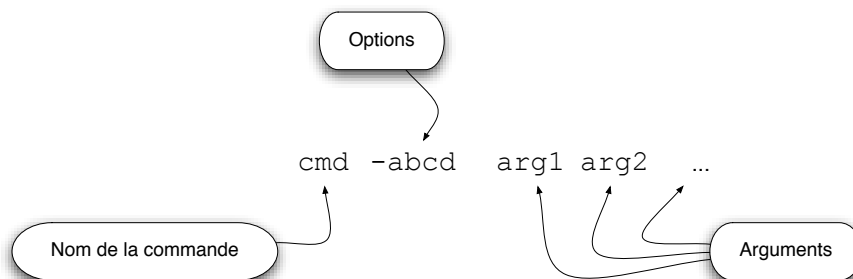
Inscrivez vous aussi pour le 'cours' DSE - Département de Sciences Exactes.

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 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 (`firefox`, `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 (*manual*) 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.

À noter que la commande `w` (pour *what*) est plus complète et vous dit plus précisément qui fait quoi sur la machine.

Exercice 8 – Qui, quoi...

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

Exercice 9 – ... 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_fichier1 mon_fichier2 etc.` permet d'afficher en une seule fois le contenu entier des fichiers `mon_fichier1 mon_fichier2 etc.` Pour afficher un long texte page par page, on utilisera `less mon_fichier`.

Exercice 10 – Chemin relatif, chemin absolu

En utilisant la commande `less`, affichez de deux façons différentes le contenu de votre fichier `nom_prenom.txt` créé à l'exercice 3 (c'est-à-dire avec le chemin absolu du fichier, puis avec un de ses chemins relatifs).

Exercice 11 – Interrompre un programme, le symbole &

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

Correction. L'esperluette (le symbole `&`) permet de dire au shell de reprendre la main sans attendre la fin de la tâche lancée. On dit que la tâche est lancée *en arrière-plan*.

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 se révèle souvent bien plus rapide qu'utiliser l'interface graphique.

`pwd` : Utilisée sans argument, vous indique dans quel répertoire vous vous trouvez actuellement, autrement dit le répertoire courant (*print working directory*).

À l'ouverture d'un nouveau terminal, le répertoire courant est toujours votre répertoire personnel.

`cd` : La commande `cd rep` change (*change directory*) 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 (*list directory*) le contenu du répertoire courant : ses fichiers et ses sous-répertoires.

Astuce : pour aller plus vite, vous pouvez indiquer le début d'un nom de fichier ou de répertoire et essayer de le compléter avec la touche Tab (tabulation). Si plusieurs fins sont possibles elles vous seront proposées.

Exercice 12 – Tour du propriétaire

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

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

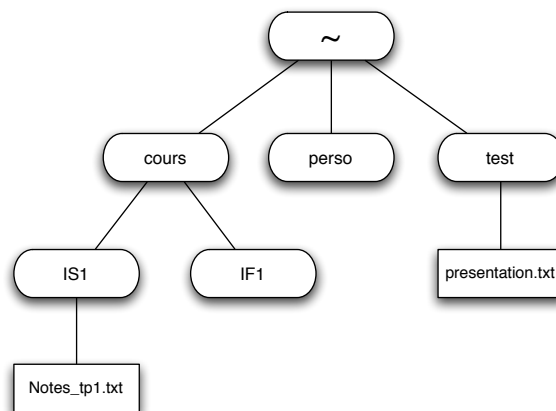
Exercice 13 – Chemin relatif, chemin absolu

En utilisant la commande `less`, affichez de deux façons différentes le contenu de votre fichier `nom_prenom.txt` créé à l'exercice 3 (c'est-à-dire avec le chemin absolu du fichier, puis avec un de ses chemins relatifs).

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

Exercice 14 – 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.)



Correction.

```
mkdir -p cours/IS1 cours/IF1 perso test  
touch cours/IS1/Notes_tp1.txt test/presentation.txt
```

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, par exemple dans un fichier intitulé `Notes_tp1.txt` ; vous pouvez aussi créer un répertoire par tp, à vous de vous organiser...

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 fi c1 fi c2` déplace (*move*) le fichier de chemin *fic1* dans l'arborescence pour lui attribuer le chemin *fic2*. Après exécution de la commande, *fic2* est un chemin de l'arborescence qui mène au fichier mais plus *fic1*.

Exercice 15 – Déplacement de fichiers dans l'arborescence

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

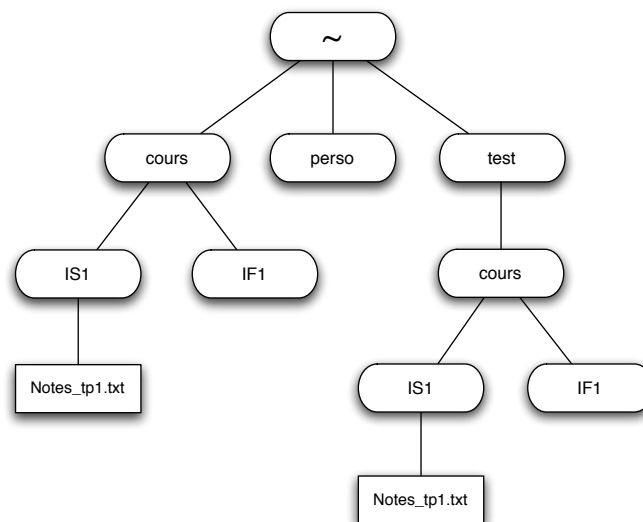
Exercice 16 – Fichiers cachés

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 fi c1 fi c2` duplique (*copy*) le fichier de nom *fic1* en un nouveau fichier de nom *fic2*. Après exécution de la commande, *fic1* et *fic2* sont deux fichiers de même contenu mais complètement autonomes.

Exercice 17 – 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 :



Correction.

```
cp -R cours test/
```

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

Exercice 18 – 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` ?

Correction. Il faut d'abord vider le répertoire qu'on veut supprimer avec `rmdir` ; pour supprimer un répertoire avec son contenu, utiliser `rm -r`.

Exercice 19 – Déplacement de fichiers et répertoires

Expérimentez la commande `mv` en déplaçant des fichiers et des répertoires dans votre arborescence.

`diff` : La commande `diff fic1 fic2` compare ligne à ligne *fic1* et *fic2*.

`touch` : La commande `touch fic` permet, en outre, de créer un fichier vierge (de taille 0) appelé *fic*.

Exercice 20 – Comparaison de fichiers et répertoires

Copiez un répertoire, changez légèrement le contenu d'un fichier et retrouvez à l'aide de `diff` quelles sont les différences que vous avez opérées. Ajoutez maintenant un fichier à l'un des deux répertoires et repérez-le avec `diff`.

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 :

* : représente n'importe quelle suite de zéro, un ou plusieurs symboles

? : représente exactement un symbole quelconque.

Ainsi, si le répertoire courant contient `tp1.ps`, `tp1.ps`, `tp-i s1.ps`, `truc`, `trucage`, `trucmuche` et `struck`. Ecrire `tp?.ps` désigne `tp1.ps`, `tp1.ps` mais pas `tp-i s1.ps`, et `truc*` désigne `truc`, `trucage`, `trucmuche` mais pas `struck`.

Exercice 21

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).

Correction.

```
ls /usr/bin/k?????  
ls /usr/lib/*.so
```

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 22 – 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 23 – 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 24 – 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` ?

Correction.

```
ls -G test
```

Exercice 25 – 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`.

Correction.

```
rm -r test/cours
```

Exercice 26 – 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.

Correction.

```
rm -i test/nom_prenom.txt
```


Espace disque

Exercice 27 – quota

Que fait la commande quota ?

Correction. quota vous permet de voir quel pourcentage de disque vous occupez sur ce qui vous est alloué.

Exercice 28 – du

Que fait la commande du ? et son option -h ? comment régler la profondeur ?

Correction. du permet de calculer l'espace disque utilisé par un fichier.
du -d 2 par exemple.

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 <i>fic1 fic2</i> ... dans toto.tar tar xvf toto.tar : extrait dans le répertoire courant les fichiers contenus dans toto.tar

Exercice 29 – 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 des commandes cat puis less. Que constatez-vous ?

Exercice 30 – 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. Essayez aussi avec la commande less. Cherchez enfin 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.

Correction.

```
tar tvf cours.tar
```

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 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 31 – Mes propres commandes

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

Correction.

```
alias la="ls -a"
```

`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 32 – 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.

Correction.

```
alias rm="rm -i"
alias ls="ls -G"
```

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

Exercice 33 – 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 34 – Mes propres fichiers de configuration

Éditez le fichier `.bashrc` situé dans votre répertoire personnel¹ avec `xemacs` pour rajouter les commandes `ions`

6 Se déconnecter

Pour quitter l'environnement graphique, il faut d'abord avoir quitté toutes les applications (pour quitter un terminal, taper la commande `exit`). 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.