

Séance 1: PREMIERS PAS

Université Paris-Diderot

Objectifs:

- | | |
|----------------------------------|--------------------------------------|
| — S'inscrire au MOODLE du cours | — Utiliser le compilateur JAVA |
| — Apprendre à manipuler le Shell | — Savoir exécuter un programme donné |

Première connexion

Pour utiliser un système UNIX, la première chose à faire est de se connecter. 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 permet d'ouvrir une session.

Une fois votre nom d'utilisateur et mot de passe vérifiés, le gestionnaire de bureau apparaît. (Il s'agit du programme qui prend en charge l'affichage des menus et des fenêtres, ici XFCE.) Le menu principal (bouton 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.

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 (Inscription sur le site du cours IP1 Java, ☆)

1. Lancez le navigateur Firefox.
2. Ouvrez la page de la plateforme MOODLE : <http://moodle.script.univ-paris-diderot.fr/>.
3. Cliquez sur « Connexion » (en haut à droite) pour vous identifier.
(Une fois connecté, vous serez redirigé vers l'ENT.)
4. Allez dans « Sciences » puis « Département de formation de Licence L1 L2 sciences exactes (SE) » et enfin « L1 ».
5. Cliquez sur « Initiation à la Programmation en Java » et inscrivez-vous.
6. Dans la section « Supports TP », vous trouverez les énoncés des différents TP ainsi que les programmes donnés.

□

À la découverte du *Shell*

Commandes. Il est possible d'interagir avec le système d'exploitation de manière plus fine que *via* une interface graphique classique en utilisant un *terminal* dans lequel peuvent être tapées des *lignes de commande*. Ces lignes sont interprétées par un programme appelé *shell* dont le rôle est d'attendre que vous lui demandiez

d'exécuter une commande pour le faire. La fin de la saisie d'une ligne de commande est indiquée en appuyant sur la touche `Entrée` (`Enter`).

Le shell indique qu'il est prêt en affichant en début de ligne une *invite de commande* (ou *prompt*) se terminant en général par le caractère « dollar » (\$) ou « supérieur » (>).

Le langage du *shell* est un véritable langage de programmation : il est très utilisé pour écrire des *scripts* servant à automatiser certaines tâches courantes. Nous nous contenterons d'utiliser le *shell* de la façon la plus simple qui soit pour le moment en nous focalisant sur l'invocation de commandes. La forme la plus courante d'une invocation de programme en ligne de commande est :

`cmd opts args`

où

- `cmd` est le nom de la commande à exécuter ;
- `opts` est une liste (éventuellement vide) d'options, permettant d'affiner le comportement de la commande ; en général, une option débute par un tiret (par exemple, `-a`, `-l`, `--help`, `-prefix ...`) ;
- `args` est une liste (éventuellement vide) d'arguments en fonction desquels la commande agit.

La commande peut en particulier être le nom de n'importe quelle application qui pourrait être lancée depuis l'interface graphique (comme `firefox` par exemple), mais il existe également tout un ensemble de *commandes UNIX* qui interagissent avec le terminal.

La combinaison de touches `C-c` (c'est-à-dire obtenu en appuyant sur la touche `ctrl` et la touche `c`) permet souvent d'interrompre le programme en cours d'exécution.

Édition de ligne. Si on se trompe en tapant une commande, et que l'on s'en aperçoit avant d'appuyer sur `Entrée`, on peut utiliser les touches `←` et `→` pour déplacer le curseur à l'endroit où est l'erreur.

La combinaison de touches `C-a` (c'est-à-dire obtenu en appuyant sur la touche `ctrl` et la touche `a`) permet de se rendre au début de la ligne.

La combinaison de touches `C-e` (c'est-à-dire obtenu en appuyant sur la touche `ctrl` et la touche `e`) permet de se rendre à la fin de la ligne.

La combinaison de touches `C-d` (c'est-à-dire obtenu en appuyant sur la touche `ctrl` et la touche `d`) permet de supprimer le caractère suivant.

La combinaison de touches `C-t` (c'est-à-dire obtenu en appuyant sur la touche `ctrl` et la touche `t`) permet d'échanger le caractère suivant et le caractère précédent.

(Pour des compléments sur l'édition de ligne :

http://www.math.utah.edu/docs/info/features_7.html.)

Historique. Si on ne s'aperçoit de l'erreur qu'après avoir invoqué la commande, on veut souvent lancer une autre commande corrigée. Au lieu de tout retaper, on peut utiliser la touche `↑`, qui rappelle la commande précédente (puis la commande d'avant, etc., si on appuie plusieurs fois). La touche `↓` permet de redescendre dans l'historique des commandes, vers les commandes les plus récentes.

On peut aussi consulter l'historique en tapant :

```
1 history
```

La combinaison de touches `C-r` (c'est-à-dire obtenu en appuyant sur la touche `ctrl` et la touche `r`) fait passer le shell dans un mode "recherche dans l'historique" : en tapant un mot, on fait apparaître la dernière commande contenant ce mot. Pour interrompre cette recherche, il suffit de faire `C-c`.

Complétion. Lorsque l'on veut taper le nom d'un fichier existant, on peut taper le début du nom du fichier puis appuyer sur la touche *tabulation* (marquée Tab ou désignée parfois avec une grande flèche vers la droite). Le *shell* insère alors la fin du nom (s'il y a plusieurs possibilités, le *shell* complète seulement le plus long préfixe commun). La complétion a deux avantages : elle permet de moins taper, et elle assure que le nom complété existe.

La commande `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`.

Par exemple, la ligne de commande `man ls` permet d'obtenir la documentation de la commande `ls`. On peut faire défiler le texte à l'aide de la barre d'espace et des flèches ↑ et ↓, et quitter à l'aide de la touche q. La touche / permet de lancer la recherche d'un mot dans la page de manuel.

Exercice 2 (Utilisation de `man`, ☆)

1. Tapez `man ls` et analysez la structure de la page de manuel.
2. Que fait la commande `ls` ?
3. À quoi sert l'option `-l` de la commande `ls` ?
4. Que fait la commande `man man` ?
5. Lisez les premières lignes des manuels des commandes suivantes : `java`, `javac`, `emacs`.
À quoi servent-elles ?

□

Fichiers. Un fichier est une suite de données, représentant par exemple un texte, une image, etc. Chaque fichier possède un nom, conventionnellement terminé par un point suivi d'une suite de caractères – son extension – indiquant le type de données qu'il contient. Par exemple, le fichier qui contient l'énoncé de ce TP s'appelle `IP1-tp-1-sujet.pdf`, et son nom indique qu'il est au format PDF.

Les systèmes UNIX (comme ceux de la salle TP) font une différence entre majuscules et minuscules : `tp0.pdf`, `TP0.pdf` et `TP0.PDF` désignent trois fichiers différents.

Répertoires. Sur les systèmes UNIX, les fichiers sont organisés sous forme d'un *arbre* : chaque fichier est stocké dans un *répertoire* (aussi appelé *dossier*) et les répertoires peuvent eux-mêmes contenir d'autres répertoires.

Le répertoire *home* Le répertoire dit *home* (« maison », ou parfois *répertoire personnel*), noté « ~ », est l'endroit où vous pouvez stocker vos fichiers personnels. Où que vous soyez, si vous tapez « `cd ~` », vous vous retrouverez dans le répertoire *home*.

Quelques commandes utiles

- La commande `mkdir rep` permet de créer un répertoire `rep`, sous-répertoire de votre répertoire courant.
- La commande `cd rep` permet de « descendre » dans le répertoire `rep` (**Attention** : il faut que ce répertoire soit un sous-répertoire du répertoire où vous vous trouvez).
- La commande `cd ..` permet de remonter au répertoire parent.
- La commande `pwd` permet de savoir où vous trouvez dans l'arbre.
- La commande `unzip fichier.zip` décompresse un fichier d'archivage au format zip dans le répertoire courant.

Exercice 3 (Création de répertoires, ☆)

1. Allez à votre répertoire *home*.
2. Exécutez `ls -l`.
3. Créez un répertoire `IP1-Java`.
4. Exécutez `ls -l`. Qu'observez-vous ?
5. Descendez dans le répertoire `IP1-Java`.

6. Exécutez `pwd`.
7. Exécutez `cd ...` Où vous trouvez-vous dans l'arborescence ? Vérifiez-le en faisant `pwd`.
8. Descendez dans le répertoire `IP1-Java` et créez un sous-répertoire `TP1`.
9. Exécutez `ls -l` et vérifiez ainsi que le répertoire `IP1-Java` contient un sous-répertoire `TP1`.

□

Important : Prenez l'habitude de créer un sous-répertoire par TP.

Le système d'exploitation que vous utilisez met à votre disposition différents éditeurs de texte (*kwrite*, *gedit*, *vi*, *Emacs* etc.). Nous vous proposons d'utiliser *Emacs*, qui est particulièrement bien adapté à la programmation. Pour le lancer, il suffit de taper « `emacs &` ».

Exercice 4 (Création d'un premier fichier avec Emacs, ★)

1. À l'aide d'EMACS, créez un fichier `poeme.txt` qui sera à mettre dans le répertoire `IP1-Java/TP1/` ; assurez-vous que le tampon est bien en mode Text (regardez en bas de la fenêtre). Tapez votre poème favori (quelques vers suffiront). Sauvegardez le fichier dans le répertoire `IP1-Java/TP1/poeme.txt`, mais ne le fermez pas.
2. Toujours dans le même EMACS, créez un fichier `chanson.txt` où vous taperez les paroles de votre chanson favorite (quelques vers suffiront). Sauvegardez.
3. Revenez au tampon du fichier `poeme.txt`. (Indication : regardez dans le menu *Buffers*.)
4. Dans le terminal, allez voir avec la commande `ls -l` si dans le répertoire `IP1-Java/TP1/` il y a bien les fichiers `poeme.txt` et `chanson.txt`.
5. Fermez EMACS.
6. Dans le terminal, dans le répertoire, `IP1-Java/TP1/`, tapez `emacs poeme.txt &` . Qu'observez-vous ?
7. Fermez EMACS.

□

Exercice 5 (Compilation dans le terminal et dans Emacs, ★)

1. Dans un fichier nommé `Hello.java`, écrivez le programme suivant :

```

1 public class Hello {
2     public static void main (String[] args) {
3         System.out.println ("Hello!");
4     }
5 }
```

2. Exécutez la commande suivante dans le terminal :

```
1 javac Hello.java
```

3. Observez les fichiers créés par cette commande. À quoi correspondent-il d'après vous ?
4. Exécutez maintenant la commande suivante dans le terminal :

```
1 java Hello
```

5. Qu'observez-vous ?
6. On modifie maintenant le fichier pour y introduire une erreur de syntaxe. Pour cela, supprimer la première accolade fermante. Compilez de nouveau dans le terminal. Que se passe-t-il ? Quelle est la ligne de l'erreur de syntaxe détectée par le compilateur ? Savez-vous retrouver cette ligne dans EMACS ?

7. Maintenant, dans EMACS, tapez la combinaison de touches M-x (en appuyant simultanément sur la touche alt et la touche x). EMACS va placer le curseur dans une petite zone d'édition située en bas de la fenêtre. Tapez alors compile puis appuyez sur entrée et tapez ensuite javac Hello.java. Que se passe-t-il ?
8. L'intérêt de la compilation dans EMACS est de pouvoir se déplacer automatiquement à l'erreur suivante de compilation grâce à la combinaison de touches C-x suivie de la touche '. Utilisez ce raccourci clavier pour vous rendre à l'endroit du programme où se situe l'erreur de syntaxe, corrigez-la puis compilez de nouveau dans EMACS pour vérifier qu'elle a effectivement disparu.

□

Exercice 6 (Modification d'un programme, **)

1. Dans un fichier nommé Repetition.java, écrivez le programme suivant :

```
1 public class Repetition {
2     public static void main (String[] args) {
3         for( int i =0; i<10;i++) {
4             System.out.println("Bonjour!");
5         }
6     }
7 }
```

2. Compilez et exécutez de programme.
3. Que fait ce programme ?
4. Modifiez le fichier pour qu'il affiche 20 fois la chaîne "Bonjour!" et vérifiez en compilant et exécutant de nouveau.
5. Comment faire pour faire afficher au programme 23 fois la chaîne "Repeat" ?

□

Exercice 7 (Raccourcis claviers d'Emacs, **)

Vous serez plus productif en utilisant le moins possible la souris. Pour cela, il faut éviter d'utiliser les menus et préférer les raccourcis claviers, des combinaisons de touches qui sont équivalentes à la sélection d'une commande dans le menu. Pour terminer ces travaux pratiques, expérimentez donc les raccourcis claviers suivant :

1. C-x C-s : sauvegarde le fichier courant.
2. C-x C-f : ouvre un fichier en demandant de tapant son nom. Si ce nom n'existe pas alors un nouveau fichier est créé.
3. C-x C-c : quitte EMACS.
4. C-x _ : revient en arrière dans l'édition.
5. C-k : coupe.
6. C-y : colle.

(Pour plus de raccourcis, consultez par exemple :

https://www.shortcutworld.com/en/linux/Emacs_23.2.1.html)

□