

## TP Algorithmique programmation

### TP 0 : Prise en main de l'environnement de travail

Vous pourrez trouver toutes les ressources concernant les TP (planning des séances, feuilles d'exercices, corrigés, ...) sur la page web suivante :

<http://www.irisa.fr/prive/jraclet/CNAM/>

Vous allez travailler sous le système d'exploitation **Unix** (Linux - Fedora Core) . Les programmes Java seront édités avec le logiciel **xemacs** sous le mode **JDE**. Le but de cette feuille est de constituer un aide mémoire pour l'utilisation de ces outils.

## 1 Le shell Unix

Vous aurez besoin de taper des commandes Unix pour plusieurs raisons : lancer xemacs, éventuellement compiler puis exécuter vos programmes, gérer vos fichiers, ... Pour cela, lancez une fenêtre *shell* (sous Fedora, cliquez sur "Applications" dans la barre en haut puis sélectionnez "Terminal" dans le menu). Le *shell* permet d'exécuter immédiatement des commandes en tapant leur nom, et éventuellement en précisant sur quoi elles doivent agir. Lorsqu'on le lance, il affiche une invite, par exemple :

```
[24007580@e106m01]$
```

On peut alors taper une commande, et appuyer sur la touche "entrée" pour l'exécuter :

```
[24007580@e106m01]$ ls -l 
```

### Commandes du shell

Voici quelques commandes utiles et leur syntaxe :

<code>pwd</code>	affiche le chemin (nom complet) du répertoire courant
<code>ls -l</code>	afficher la liste des fichiers dans le répertoire courant
<code>cp fichier-original nom-de-la-copie</code>	faire une copie d'un fichier
<code>mv ancien-nom nouveau-nom</code>	renommer un fichier
<code>rm</code>	fichier supprimer définitivement un fichier
<code>mkdir nom-de-répertoire</code>	créer un nouveau répertoire dans le répertoire courant
<code>cd nom-de-répertoire</code>	changer de répertoire courant
<code>cd ..</code>	aller au répertoire parent du répertoire courant
<code>rmdir nom-de-répertoire</code>	supprimer un répertoire vide
<code>xemacs &amp;</code>	lancer xemacs
<code>javac Toto.java</code>	compiler le programme java Toto
<code>java Toto \textit{arg0 arg1}</code>	exécuter le programme java Toto

Il y a bien sûr beaucoup d'autres commandes... (environ 5000 sur la machine où ce texte a été tapé).

## Quelques aides pour taper les commandes

**Édition de ligne** : si on se trompe en tapant une commande, et qu'on s'en aperçoit avant d'appuyer sur "entrée", on peut utiliser les touches flèches gauche et droite pour déplacer le curseur à l'endroit où est l'erreur.

**Historique** : si on ne s'aperçoit de l'erreur qu'après avoir démarré la commande, on veut souvent lancer une autre commande corrigée. Au lieu de tout retaper, on peut utiliser la flèche vers le haut, qui rappelle la commande précédente (puis la commande d'avant, etc., si on appuie plusieurs fois).

**Complétion** : lorsqu'on veut taper le nom d'un fichier existant, on peut taper le début du nom du fichier puis appuyer sur la touche tabulation. 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.

**Interrompre une commande** : une autre touche à connaître est C-c (contrôle-C : enfoncer la touche "Ctrl", puis appuyer ponctuellement sur "c", et relâcher "Ctrl"). Cette touche interrompt le *shell* ; elle annule la commande en cours, et le shell affiche une nouvelle invite.

## 2 Xemacs et le mode JDE

**Lancement** : l'éditeur de texte xemacs se lance en tapant la command `xemacs &` dans une fenêtre *shell*.

**Créer/Ouvrir un fichier** : **Fichier-Ouvrir** dans le menu d'xemacs ou raccourci clavier C-x C-f. Indiquer dans le minibuffer le chemin et le nom du fichier Java que l'on veut ouvrir. Pour accéder plus facilement au fichier désiré, on peut utiliser la complétion automatique (touche "Tabulation"). Si le fichier que l'on essaie d'ouvrir n'existe pas, il est alors créé.

**Éditer un fichier** : au début de chaque nouvelle ligne, utiliser la touche "Tabulation" afin d'utiliser l'indentation automatique.

Saisir le code source en tenant compte des conventions du langage Java.

Sauver le fichier source à l'aide de la commande **Fichier-Enregistrer** ou du raccourci clavier C-x C-s.

**Commandes utiles** : abandonner la commande en cours C-g.

Changer de buffer actif : menu **Buffers** puis cliquer sur le buffer désiré.

Supprimer le buffer actif : menu **Buffer-Détruire le buffer** ou raccourci clavier C-k.

Partager l'écran : menu **View-Diviser en fenêtre** ou raccourci clavier C-x 2.

Afficher une seule fenêtre : **View-Agrandir la sous-fenêtre** ou raccourci clavier C-x 1.

Pour rectifier l'indentation d'une zone du texte source, sélectionner cette zone à la souris puis choisir la commande **Java-Indent line or region**.

**Compiler** : pour compiler le code source du buffer actif : menu **JDE-Compile** ou raccourci clavier C-c C-v C-c. Si le fichier source n'a pas été sauvegardé, une sauvegarde est proposée.

Si la compilation génère des erreurs, celles-ci sont affichées en rouge dans la fenêtre de compilation. En cliquant dans la fenêtre de compilation, un nouveau menu **Compile** apparaît. La commande **Compile-Next Error** (raccourci clavier C-x AltGr-7) permet de passer dans la fenêtre du code source à la ligne où se situe l'erreur.

Après correction des erreurs et nouvelle sauvegarde, effectuer une nouvelle compilation.

**Exécuter** : le fichier ayant été correctement compilé, exécuter la classe avec la commande **JDE-Run App** (raccourci clavier C-c C-v C-r). La saisie des arguments est proposée dans le minibuffer. S'il n'y a pas d'arguments, valider par "Entrée" après avoir éventuellement supprimé les arguments proposés.