

**DEV1 – ENVL – Laboratoires d'environnement****TD 3 – Java sur linux****Résumé**

Lorsqu'on programme en JAVA sur LINUX, on peut, comme sur WINDOWS, utiliser un environnement de développement comme NETBEANS. Mais on peut aussi tout faire en mode console, c'est ce que nous allons voir ici. Nous en profiterons pour apprendre de nouvelles notions liées à Linux.

**Table des matières**

<b>1</b>	<b>Quelques notions supplémentaires de Linux</b>	<b>2</b>
1.1	La ligne de commande . . . . .	2
1.2	Les fichiers cachés . . . . .	3
<b>2</b>	<b>Compiler et exécuter du Java</b>	<b>3</b>
<b>3</b>	<b>L'éditeur nano</b>	<b>4</b>
3.1	Coloration syntaxique . . . . .	4
3.2	Numérotation des lignes . . . . .	4
3.3	Indentation . . . . .	4
3.4	Autres configurations . . . . .	4
<b>4</b>	<b>Développer un programme Java</b>	<b>5</b>
<b>5</b>	<b>Transfert de fichiers</b>	<b>5</b>

Dans votre répertoire `home/dev1`, créez un répertoire `td3`. Ce répertoire contiendra tous les fichiers que vous allez créer aujourd'hui.

# 1 Quelques notions supplémentaires de Linux

Avant d'attaquer le cœur du TD, à savoir le compilation et l'exécution de programmes Java, voyons quelques éléments de Linux qui vous seront utiles.

## 1.1 La ligne de commande

Commençons par quelques astuces pour vous faciliter la vie lorsque vous entrez des commandes.

### La complétion de commandes

Parfois, vous devez entrer une commande assez longue parce que les noms de fichiers sont longs et/ou nombreux. Linux offre plusieurs facilités pour simplifier l'entrée de longues commandes.

Lorsque vous appuyez sur la touche **TAB**, le shell tente de compléter le début de commande que vous avez déjà tapé. Si plusieurs possibilités existent, elles sont affichées si vous appuyez 2x sur **TAB**.

#### Tutoriel 1

### La complétion de la commande

Supposons que vous ne vous rappeliez plus très bien de la commande qui permet de modifier le mot de passe. Vous vous rappelez juste qu'elle commence par **pas**.

- ✍ Tapez **pas** puis appuyez 2x sur la touche **TAB**.
- ✍ Entrez un **s** puis appuyez à nouveau sur **TAB**.

La touche de tabulation permet également de compléter un nom de fichier.

#### Exercice 1

### La complétion des noms de fichiers

1. Dans votre dossier **td3**, copiez le fichier `monfichier aunom tellement long qu'il me paraît peu probable de le taper 2x sans erreur` qui se trouve dans le dossier `/eCours/java/td/td3`.
2. Affichez le contenu de ce fichier en évitant de retaper son nom.

### Revenir à une commande précédente

Il arrive souvent qu'il faille entrer une commande qu'on a déjà écrite il y a peu (ou en tout cas fort proche de ce qu'on a déjà écrit). C'est là que les flèches viennent à notre secours.

La flèche vers le haut permet de revenir aux commandes précédentes et de les modifier. À utiliser sans modération...

### Historique des commandes

- ▷ **history** : affiche la liste des dernières commandes entrées. Chaque commande est numérotée.
- ▷ **!numéro** : ré-exécute la commande de numéro donné.
- ▷ **!cmd** : ré-exécute la dernière commande qui commençait par **cmd**.

## 1.2 Les fichiers cachés

### Les fichiers cachés

Un **fichier caché** est un fichier dont le nom commence par « . » (un point).  
Idem pour un dossier.

Par défaut, les fichiers cachés ne sont pas montrés par la commande `ls`. Pour les voir, utiliser l'option `a` : `ls -a`. C'est surtout utilisé pour des fichiers de configuration.

#### Exercice 2

### Les fichiers cachés de la home

Regardez s'il existe des fichiers cachés dans votre répertoire personnel.

## 2 Compiler et exécuter du Java

### Java sans package

Si le programme Java n'utilise pas de package :

- ▷ `javac MaClasse.java` compile le programme Java du fichier donné
- ▷ `java MaClasse` : exécute le programme Java se trouvant dans la classe donnée.

#### Tutoriel 2

### Compiler / exécuter un programme

Commençons par un programme correct et tentons de l'exécuter.

Le fichier `/eCours/java/td/td3/Ex.java` contient un petit programme Java tout simple qui affiche un message de bienvenue.

- ✍ Si ce n'est pas le cas, placez-vous dans le dossier `td3`.
- ✍ Copiez le fichier indiqué dans `cd` dossier : `cp /eCours/java/td/td3/Ex.java .`
- ✍ Lisez-le et voyez si vous devinez ce qu'il fait : `cat Ex.java`
- ✍ Compilez-le : `java Ex.java`
- ✍ Que fait cette phase, à quoi sert-elle ? Affichez le contenu du dossier pour le vérifier.
- ✍ Exécutez le programme : `java Ex`

Lorsqu'on compile, il faut mettre le `.java` mais lorsqu'on exécute, il ne faut pas mettre le `.class`.

### Retenez !

On **compile un fichier** mais on **exécute une classe**.

### 3 L'éditeur nano

Un éditeur de texte, même simple comme nano, peut apporter quelques facilités dans l'écriture de programmes.

#### 3.1 Coloration syntaxique

La coloration syntaxique signifie utiliser des couleurs pour mettre en évidence certaines parties d'un code : mots clés, constantes...

Pour utiliser cette facilité, il faut configurer NANO. Cette configuration se fait dans le fichier `~/.nanorc`.

##### Tutoriel 3

#### Introduire la coloration syntaxique

- ✍ Tapez `nano ~/.nanorc` pour éditer le fichier de configuration de *nano*.
- ✍ Ajoutez-y la ligne : `include "/usr/share/nano/java.nanorc"`
- ✍ Quittez l'éditeur.
- ✍ Ouvrez le fichier `Ex.java` ; il devrait être coloré.

#### 3.2 Numérotation des lignes

Nano peut également indiquer le numéro de la ligne sur laquelle se trouve le curseur, ce qui sera pratique pour corriger vos erreurs. Pour cela, il existe deux méthodes.

- ▷ **Méthode 1** : Lancer nano avec l'option `-c` : `nano -c monFichier`.
- ▷ **Méthode 2** : Dans l'éditeur, appuyez sur `CTRL-c`.

#### 3.3 Indentation

Pour qu'un programme soit lisible, il doit être *indenté*. Ce qui serait pratique lorsqu'on code ce serait qu'un retour à la ligne positionne automatiquement le curseur de façon à être aligné avec la ligne précédente. Pour que nano fasse ça pour nous, il suffit d'ajouter ceci à son fichier de configuration : `set autoindent`.

#### 3.4 Autres configurations

Si vous désirez connaître d'autres possibilités de configuration de nano, vous pouvez lire le manuel : `man nanorc`.

Pour une *quick ref* en ligne, consultez (par exemple) :

[www.codexpedia.com/text-editor/nano-text-editor-command-cheatsheet/](http://www.codexpedia.com/text-editor/nano-text-editor-command-cheatsheet/) .

## 4 Développer un programme Java

Vous allez à présent écrire votre premier programme de bout en bout sur linux1.

### Exercice 3

#### Un programme Java

Écrivez un programme qui :

- ▷ Affiche votre nom ;
- ▷ Affiche la valeur de  $7^2$  ;
- ▷ Affiche la valeur de  $2^7$  ;
- ▷ Demande un entier à l'utilisateur et affiche s'il est pair.

#### Conseils

Procédez par étape et vérifiez correctement votre programme. Ce n'est pas parce qu'il compile qu'il est correct. Et ce n'est pas parce qu'il affiche quelque chose que c'est la bonne réponse.

## 5 Transfert de fichiers

Vous n'avez peut-être pas fini. Pour pouvoir continuer à la maison sans tout recommencer, il serait bon de pouvoir récupérer ce que vous avez déjà fait sur linux1. Voici une façon de le faire.

### Tutoriel 4

#### Transférer des fichiers

- ✍ Ouvrez l'explorateur de fichier Windows (par exemple en cliquant sur l'icône "My Computer").
- ✍ Dans le champ d'adresse, tapez l'adresse `ftp://linux1`.
- ✍ Une boîte de dialogue vous demande votre login et mot de passe (sur *linux1*).
- ✍ Vous voyez apparaître votre dossier personnel sur *linux1*.
- ✍ Vous pouvez y prendre/déposer des fichiers comme vous le feriez pour un dossier Windows. Vous pouvez par exemple les mettre sur une **clé USB**, sur le cloud ou vous les envoyer par mail.