

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

1	Quelques notions supplémentaires de Linux	2
1.1	Les fichiers cachés	2
2	L'éditeur nano	2
2.1	Premiers pas	2
2.2	Coloration syntaxique	3
2.3	Numérotation des lignes	3
2.4	Indentation	3
2.5	Autres configurations	3
3	Java en mode console	3
3.1	Compiler et exécuter	4
3.2	Produire la Javadoc	5
4	Transfert de fichiers	5
5	Conclusion	5
5.1	Ce qu'il faut connaître	6

Dans votre répertoire `~/dev1`, créez un répertoire `td3`. Il 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 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 1

Les fichiers cachés de la home

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

2 L'éditeur nano

Un **éditeur de texte** (ou, plus court, un **éditeur**) est un programme qui vous permet d'entrer, modifier le contenu d'un fichier texte. Vous connaissez probablement NOTEPAD sous WINDOWS. Sur LINUX, il en existe beaucoup. Celui que nous vous proposons dans un premier temps au laboratoire s'appelle NANO. Il est assez facile pour débiter et faire des choses simples. Par la suite, vous pourrez choisir de rester sous NANO ou d'utiliser VIM.

2.1 Premiers pas

Tutoriel 1

Premiers pas avec nano

- ✍ En préambule, tapez la commande `cd` pour revenir dans *votre home*.
- ✍ Tapez `nano test` pour commencer à éditer le fichier `test` (comme il n'existe pas encore, il est créé).
Une fenêtre s'ouvre. Vous voyez qu'elle est scindée en 2 parties : la partie supérieure où vous écrivez votre texte et la partie inférieure où sont indiquées les différentes commandes (le `^` représente la touche `Ctrl`)
- ✍ Entrez quelques mots.
- ✍ Appuyez sur la combinaison de touches `Ctrl X`, confirmez que vous voulez sauver vos modifications et sortez.
Vous êtes maintenant revenu à l'invite de commande.
- ✍ Tapez à présent la commande `ls`. Vous pouvez constater que le fichier `test` est apparu dans la liste ;)

Pour aller plus loin dans l'apprentissage de cet éditeur, consultez ce document :

<http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/nano-l-editeur-de-texte-du-debutant>.

2.2 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 2

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.

2.3 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`.

2.4 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`.

2.5 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/ .

3 Java en mode console

Avec Netbeans, lorsque vous appuyez sur des boutons (comme celui demandant d'exécuter le projet), cela lancent des commandes qui effectuent la tâche. Il est possible d'exécuter directement ces commandes. C'est ce que nous allons faire ici.

3.1 Compiler et exécuter

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 3 Compiler / exécuter un programme

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

Le fichier `/eCours/dev1/env1/Hello.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/dev1/env1/Hello.java .`
- ✍ Lisez-le et voyez si vous devinez ce qu'il fait : `cat Hello.java`
- ✍ Compilez-le : `java Hello.java`
- ✍ Que fait cette phase, à quoi sert-elle ? Affichez le contenu du dossier pour le vérifier.
- ✍ Exécutez le programme : `java Hello`

Retenez !

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

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

Exercice 2 Un programme Java

Écrivez un programme qui :

- ▷ Affiche votre nom ;
- ▷ Demande un nombre à l'utilisateur ;
- ▷ Affiche l'inverse du nombre (par ex : 0.25 si le nombre entré était 4).

Exercice 3 Comprendre les erreurs (I)

Supposons que vous faites une erreur dans votre programme. Par exemple en écrivant `Public` au lieu de `public`.

À quelle étape le problème va-t'il apparaître ? Sous quelle forme ? Testez !

Exercice 4 Comprendre les erreurs (II)

Supposons que vous demandez un calcul impossible dans votre programme. Par exemple `1/0`.

À quelle étape le problème va-t'il apparaître ? Sous quelle forme ? Testez !

3.2 Produire la Javadoc

Javadoc

`javadoc -doc dossierDoc MaClasse.java` génère la javadoc présente dans le fichier `MaClasse.java` dans le dossier `dossierDoc` (qui doit exister).

Exercice 5 Javadoc de Hello

Produisez la javadoc du programme `Hello` précédent.

Visualiser le résultat

La javadoc est au format HTML et peut être consulté avec un simple navigateur web mais il n'y en a pas sur `linux1`¹.

TODO Aborder le `public_html`.

4 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.

5 Conclusion

1. Un Linux avec environnement graphique dispose bien sûr d'un navigateur web.

5.1 Ce qu'il faut connaître

Notions importantes de ce TD

Voici les notions importantes que vous devez avoir assimilées à la fin de ce TD.

- ▷ La notion de fichier caché et comment les voir.
- ▷ Savoir utiliser **nano** pour éditer un petit programme Java.
- ▷ Savoir compiler et exécuter un programme Java qui n'utilise pas de package.
- ▷ Savoir créer/consulter/modifier une variable d'environnement.