Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2019 – 2020
DEV1 – ENVL

DEV1 – ENVL – Laboratoire d'environnement système

TD 3 – Nano et Java sur GNU/Linux

Résumé

Lorsqu'on programme en Java sur GNU/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, en utilisant, entre autres choses, l'éditeur de code GNU nano. Nous en profiterons pour apprendre de nouvelles notions liées à GNU/Linux.

Table des matières

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

Version du document
21 septembre 2019

1 Introduction à l'éditeur de code 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. Lorsqu'il se spécialise dans l'édition d'un code informatique, on parlera plutôt d'*éditeur de code*. Vous connaissez probablement NOTEPAD sous WINDOWS. Sur GNU/LINUX, il en existe beaucoup. Celui que nous vous proposons dans un premier temps au laboratoire s'appelle GNU NANO ou simplement NANO. Il est assez facile pour débuter et faire des choses simples. Par la suite, vous pourrez choisir de rester sous NANO ou d'utiliser VIM.

Pas à pas 1

Premiers pas avec nano

- Entrez `mkdir -p ~/dev1/td3` pour créer un dossier dédié à ce TD.
- Entrez `cd ~/dev1/td3` pour vous y placer.
- 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 ;)

C'est un bon début. Plus loin dans ce TD, nous verrons quelques facilités supplémentaires.

2 Java en mode console

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

Pas de package
Nous allons aborder uniquement des programmes JAVA *sans package*. L'utilisation de l'instruction `package` en début de code complique la tâche et nous n'aurons pas le temps d'aborder cette situation.

2.1 Compilation et exécution

Java sans package

Si le programme JAVA n'utilise pas de package :

- ▷ `javac MaClasse.java` compile, à destination de la machine virtuelle JAVA, le fichier source `MaClasse.java` contenant du code écrit en langage JAVA ;
- ▷ `java MaClasse` exécute, dans la machine virtuelle JAVA, le programme se trouvant dans la classe `MaClasse`, préalablement compilée.

Pas à pas 2

Compiler / exécuter un programme

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

Le fichier `/eCours/dev1/envl/Hello.java` contient le code source d'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/envl/Hello.java`
- ✎ Lisez-le et voyez si vous devinez ce qu'il fait : `cat Hello.java`
- ✎ Compilez-le : `javac Hello.java`
- ✎ Que fait cette phase, à quoi sert-elle ? Affichez le contenu du dossier pour le vérifier.
- ✎ Exécutez la classe : `java Hello`

Retenez !
On *compile un fichier* mais on *exécute une classe*[a].

a. Depuis `JDK 11`, il est possible d'écrire directement `java Hello.java` qui va compiler le fichier puis l'exécuter.

2.2 Production et visualisation de la javadoc

Javadoc

`javadoc -d dossierDoc MaClasse.java` génère la javadoc présente dans le fichier `MaClasse.java` dans le dossier `dossierDoc`. Ce dernier doit exister.

Exercice 1

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 en ouvrant le fichier `index.html` mais il n'y en a pas sur `linux1`[1].

1. Un GNU/LINUX avec environnement graphique dispose bien sûr d'un navigateur web. Si vous avez installé un GNU/LINUX avec environnement graphique sur votre PC, il suffit de double-cliquer sur le fichier `index.html` produit.

Pas à pas 3

Visualiser la documentation sur Windows

Voici comment procéder pour visualiser la documentation produite à partir d'un navigateur sur WINDOWS.

- Entrez `mkdir -p ~/public_html/doc`.
  C'est dans le dossier `public_html` qu'il faut placer les documents accessibles à distance[2]. Plus précisément, nous placerons la documentation dans le sous-dossier `doc`. Comme c'est la toute première fois, il faut le créer.
- Entrez `javadoc -d ~/public_html/doc Hello.java`
  pour créer la javadoc en la plaçant au bon endroit.
- Lancez votre navigateur préféré sur WINDOWS.
- Entrez comme URL : `http://linux1/~<votre login>/doc`. Vous devriez voir la javadoc s'afficher.

3 Éditeur de programme nano

Voyons quelques facilités proposées par l'éditeur NANO pour simplifier le développement 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`.

Pas à pas 4

Introduire la coloration sytaxique

- 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` ou, mieux, ajouter `set constantshow`[3] au fichier fichier `~/.nanorc`.

Méthode 2
Dans l'éditeur, appuyez sur `CTRL-c`.

Par ailleurs, il est possible[4] d'indiquer les numéros des lignes dans la marge gauche. Pour cela, lancer NANO avec l'option `-l` : `nano -l monFichier` ou, mieux, ajouter `set linenumbers` au fichier fichier `~/.nanorc`.

2. Parce que le serveur web APACHE a été installé sur `linux1` et qu'il a été configuré ainsi.
3. Ou `set const` avant la version 2.5 de GNU NANO.
4. Depuis la version 2.7.1 de GNU NANO.

3.3 Indentation

Selon Wikipédia[5] :

> « En informatique, l'indentation consiste en l'ajout de tabulations ou d'espaces dans un fichier texte. [...]
> L'indentation se définit par la manière d'arranger les blocs de code, mais surtout par le nombre d'espaces utilisés à chaque niveau. »

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 Pour aller plus loin

Voici quelques ressources pour aller plus loin dans votre apprentissage de l'éditeur :

- Le manuel : `man nanorc`.
- Un petit tutoriel :
  http://fr.openclassrooms.com/informatique/cours/reprenez-le-controle-a-l-aide-de-linux/nano-l-editeur-de-texte-du-debutant.
- Une *quick ref* en ligne :
  www.codexpedia.com/text-editor/nano-text-editor-command-cheatsheet/

4 Quelques exercices

Vous allez à présent écrire et modifier des programmes afin de vous aider à bien distinguer les phases de compilation et d'exécution.

Exercice 2 Comprendre les erreurs (I)

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

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

Exercice 3 Comprendre les erreurs (II)

Et si vous demandez un calcul impossible (par exemple `1/0`).

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

Exercice 4 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).

5. https://fr.wikipedia.org/wiki/Style_d%27indentation

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.

Pas à pas 5

Transférer des fichiers

- Ouvrez l'explorateur de fichier de WINDOWS.
- Dans le champ d'adresse, tapez l'adresse `ftp://linux1`.
- Une boite de dialogue vous demande vos login et mot de passe (sur `linux1`).
- Vous voyez apparaitre 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.

6 Conclusion

Notions importantes de ce TD

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

- ▷ Savoir utiliser NANO pour éditer un petit programme JAVA.
- ▷ Savoir compiler et exécuter un programme JAVA qui n'utilise pas de package.
- ▷ Savoir générer et visualiser la javadoc.