Utilisation de Java sur Linux

Article pour l'Echo de LINUX (*Août 1996*)
Frédéric Couchet (`mad@bocal.cs.univ-paris8.fr`)

Cet article explique comment installer et utiliser le Java Developer's Kit (JDK) sur un système Linux.
Le portage du JDK sur Linux a été effectué par Randy Chapman (chapman@u.washington.edu).

---

1. Introduction

Java est un langage orienté objet développé par Sun Microsystems. Les programmes Java sont soit des applications, soit des applets. Les *applications* sont des programmes autonomes (par exemple HotJava, le navigateur de Sun). Les *applets* sont similaires aux applications mais elles sont exécutées par un navigateur compatible Java (par exemple Netscape, HotJava).

Le JDK contient principalement les librairies de classes standard (*java.lang, java.util, java.io, java.net, java.awt, java.applet*) avec leurs sources, le compilateur (*javac*), l'interpréteur (*java*) permettant d'exécuter des applications, un interpréteur d'applets (*appletviewer*), un générateur de documentation (*javadoc*), un générateur de fichiers header C (pour l'intégration du code C dans le code Java), et un debugger.

---

2. Installation de Java

Dans un premier temps récupérez le JDK pour Linux sur l'un des sites suivants (liste non exhaustive):

- ftp://ftp.u−bordeaux.fr/pub/Linux/Java (répertoire JDK−1.0.1)
- ftp://ftp.ese_metz.fr/pub/Linux/Apps/java
- ftp://ftp.linux.org.uk/pub/linux/java

Vous avez besoin des fichiers suivants:

- linux.jdk−1.0.1−try1.common.tar.gz
- et soit linux.jdk−1.0.1−try3.shared−motif.tar.gz (si vous avez les librairies partagées Motif 2.0 sur votre système)

- soit linux.jdk−1.0.1−try3.static−motif.tar.gz (dans le cas contraire)

Pour installer le JDK (de préférence dans le répertoire /usr/local), exécutez les commandes suivantes:

```
droopy# cd /usr/local
droopy# tar zxvf linux.jdk-1.0.1-try1.common.tar.gz
et soit:
droopy# tar zxvf linux.jdk-1.0.1-try3.shared-motif.tar.gz
soit:
droopy# tar zxvf linux.jdk-1.0.1-try3.static-motif.tar.gz
```

Les différents binaires de java (compilateur, previewer ...) se trouvent alors dans le répertoire */usr/local/java/bin*. Ajoutez ce repertoire dans votre PATH (en bash):

```
export PATH=${PATH}:/usr/local/java/bin
```

Installez un navigateur Web compatible Java, par exemple Netscape (les versions de Netscape à partir de la 2.0b3 sont compatibles Java). Vous pouvez, par exemple, récupérez netscape sur ftp://ftp.ibp.fr/pub/www/netscape/navigator. L'installation est très simple:

```
droopy# cd /usr/local
droopy# mkdir netscape
droopy# cd netscape
droopy# tar xzvf netscape-v20-export.i486-unknown-linux.tar.gz
```

Pour utiliser le JDK vous devez disposer d'un système ELF basé sur un noyau 1.2.13 ou ultérieur. De plus, vous devez disposer des librairies suivantes :

```
libc.5 = /lib/libc.so.5.2.18 (librairie C standard)
libX11.so.6 = /usr/X11/lib/libX11.so.6.0 (librairie X11R6)
libXt.so.6 = /usr/X11/lib/libXt.so.6.0 (librairie Athena)
libXext.so.6 = /usr/X11/lib/libXext.so.6.0 (librairie extensions X)
libXpm.so.4 = /usr/X11/lib/libXpm.so.4.3 (librairie pixmap)
libdl.so.1 = /lib/libdl.so.1.7.11 (loader dynamique Linux)
```

Vérifiez que ces librairies sont installées sur votre système (les versions des librairies sont les versions minimales pour faire fonctionner Java). Si vous ne disposez pas de l'ensemble de ces librairies, installez celles qui manquent.

Les mises à jour dépendent de la distribution que vous utilisez. Je vais passer en revue le cas de la Slackware et de la Red Hat qui semblent les deux distributions les plus utilisées.

2.1 Slackware

Avec la distribution Slackware 3.0 ELF kernel 1.2.13 vous devez mettre à jour la librairie C et le loader (les versions fournies avec la distribution sont des versions antérieures).

Pour installer la librairie C, récupérez les fichiers libc−5.2.18.bin.tar.gz et release.libc−5.2.18 sur le site ftp suivant: ftp://ftp.ibp.fr/pub/linux/GCC . Le fichier release.libc−5.2.18 contient les instructions d'installation.
Pour le loader, récupérez le fichier ld.so.1.7.14.tar.gz sur le site ftp suivant: ftp://ftp.ibp.fr/pub/linux/GCC . Exécutez les commandes suivantes:

```
droopy# tar xzvf ld.so.1.7.14.tar.gz
droopy# cd ld.so.1.7.14
droopy# ./instldso.sh
```

Une fois les librairies installées, mise à jour des librairies:

```
droopy# ldconfig
```

2.2 Red Hat 2.1

Avec la distribution Red Hat 2.1 ELF kernel 1.2.13 vous devez également mettre à jour la librairie C et le loader. Pour la mise à jour récupérez les fichiers rpm correspondants sur un mirroir Red Hat, et utiliser la commande **rpm** pour les installer. En outre vous devez faire un **chmod 666 /dev/zero** si vous voulez utiliser java autrement que sous root (ceci concerne principalement l'appletviewer).

2.3 Red Hat 3.0

Avec la distribution RedHat 3.0 aucune mise à jour n'est nécessaire. Par contre, vous devez tout de même faire un **chmod 666 /dev/zero**.

---

3. Voir les démos

Sous netscape ouvrez le répertoire /usr/local/java/demo. Quelques démos faites avec java s'y trouvent. Dans les sous−répertoire, pour lancer les applets ouvrez les divers fichiers HTML.

---

4. Faire un programme Java

Nous allons dans un premier temps créer, compiler et exécuter un programme Java.

Pour notre premier programme Java, nous allons faire le traditionnel programme "Hello World". Créons le fichier **hello.java** (qui contient le code source java) dont le contenu est le suivant:

```
import java.io.*;

class HelloWorldApp {
        public static void main(String args[]){
                System.out.println("Hello World ! ");
        }
}
```

La compilation de notre programme s'effectue grâce à la commande **javac** avec le nom du fichier source comme argument:

```
droopy# javac hello.java
```

Cette commande crée un fichier semi−compilé qui porte le même nom que la classe qu'il décrit, suffixé par **.class**: dans notre exemple on obtient le fichier **HelloWorldApp.class**.

Pour exécuter le programme on utilise la commande **java** de la façon suivante:

```
droopy# java HelloWorldApp
```

Vous noterez que l'argument de l'interpréteur Java est le nom de la classe définie, et non pas le nom du fichier.

5. Faire une applet Java

Une applet Java est une application qui est chargée par votre navigateur (Netscape, HotJava), à travers le réseau, et qui ne réside donc pas sur le disque dur de la machine qui l'exécute.

Une applet est appelée à partir d'un fichier source HTML. Par exemple créez le fichier **applet.html** dont le contenu est le suivant:

```
<TITLE>Applet Java </TITLE>
<H1>Applet Hello World</H1>
<APPLET CODE="HelloWorld.class" WIDTH=150 HEIGHT=50>
</APPLET>
```

On voit apparaitre une nouvelle balise HTML: **APPLET**. L'option **CODE** donne le nom de la classe qu'il faut charger. Le navigateur va chercher cette classe dans le répertoire où il a trouvé la page HTML.

Le code du source Java est le suivant (fichier **hello−world.java**):

```
import java.awt.Graphics;
import java.applet.*;

public class HelloWorld extends Applet  {
  public void init() {
                resize(150,25);
        }

  public void paint(Graphics g){
                g.drawString("Hello World ! ",50,25);
        }
}
```

Vous compilez votre source java comme vu précédemment.

Pour exécuter votre applet, lancez netscape et ouvrez le fichier: **applet.html**. Dans la fenêtre netscape un message "Hello World !"va s'afficher.

Vous pouvez également visualiser le résultat de votre applet sans navigateur. Vous devez utiliser pour cela l'*appletviewer* (situé dans le même répertoire que le compilateur java). :

```
droopy# appletviewer HelloWorld
```

6. Java et noyau 2.0.0

Pour les heureux possesseurs de la version 2.0.0 du noyau Linux il est possible d'intégrer le support des binaires Java dans le noyau. Les étapes nécessaires sont décrites dans les sources du noyau (fichier Documentation/java.txt).

Vous devez dans un premier temps compiler votre noyau avec le support Java. Au niveau de la configuration du noyau par la commande **make config**

```
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N/y/?] (NEW)
```

A ce point vous devez répondre **y**.

```
Kernel support for JAVA binaries (CONFIG_BINFMT_JAVA) [N/y/m/?] (NEW) y
```

Vous répondez **y**.

Vous compilez votre noyau: **make dep; make clean; make zImage**.

Vous devez ensuite dire au noyau où se trouve l'interpréteur et l'appletviewer JAVA. Il y a deux façons de faire ceci. La première est d'éditer le fichier **fs/binfmt_java.c** (dans les sources du noyau) et de modifier les définitions de _PATH_JAVA et de _PATH_APPLET (vous recompilez ensuite le noyau). La deuxième méthode est, en étant root, de lancer la commande (on suppose que les binaires de l'interpréteur et de l'appletviewer sont dans /usr/local/java/bin/):

```
droopy# cat /usr/local/java/bin/java > /proc/sys/kernel/java-interpreter
droopy# cat /usr/local/java/bin/appletviewer > /proc/sys/kernel/java-appletviewer
```

Cette deuxième méthode ne fonctionne pas si vous avez déclarer le support Java en module.

Vous rendez exécutable vos fichier '*.class', par exemple:

```
droopy# chmod +x HelloWorldApp.class
```

Et vous pouvez l'exécuter directement:

```
droopy# ./HelloWorldApp.class
```

Vous pouvez faire de même avec vos applets. Pour cela vous insérez une nouvelle première ligne dans le fichier source HTML:

```
<!--applet-->
```

Vous rendez exécutable le fichier '.html', par exemple:

```
droopy# chmod +x applet.html
```

Et vous pouvez l'exécuter directement:

```
droopy# ./applet.html
```

7. Références

L'objet de cet article n'étant pas de décrire le langage Java en détail je vous conseille les quelques références suivantes:

- Le Java on Linux HOWTO de Eric S. Raymond que vous pouvez trouver sur ftp://ftp.ibp.fr/pub/linux/tsx-11/docs/HOWTO
- http://www.labri.u-bordeaux.fr/java
- http://pausejava.u-strasbg.fr
- http://www.gamelan.com
- http://www.javasoft.com
- http://www.blackdown.org

- news://fr.comp.lang.java