

Proyecto de Grado 2010

CERTIFICACION DE IP4JVM

Anexo III - Instalación de IP4JVM

Autor: Daniel Enrique Rosano Lorenzo

Tutores:

Ariel Sabiguero Yawelak

Leandro Scasso

Instituto de Computación

Facultad de Ingeniería

Universidad de la República

Octubre de 2011

1 Índice

1 Índice.....	2
2 Introducción.....	3
3 Instalación del JDK	3
4 Obtención de IP4JVM.....	5
5 Instalación de ant.....	10
6 Instalación de Open JDK + IP4JVM.....	11
7 Test de la VM modificada	14
8 Solución de errores.....	15
8.1 Error de Jaxp.....	15
8.2 Error de Jaxws.....	15
8.3 Error de pcap.h.....	16
8.4 Error de jni.h.....	16
8.5 Error de la carpeta lib.....	16
9 Glosario.....	17

2 Introducción

El presente apéndice presenta los pasos necesarios para instalar OpenJDK+IP4JVM en OpenSuse 11.3. Este apéndice es una actualización al apéndice de instalación creado por Roger Abelenda e Ignacio Corrales en el año 2007.

Como dato informativo, a continuación se presenta una tabla que resume los productos y las versiones utilizadas.

Producto	Versión actual
OpenSuse	11.3 (Build 2.6.32)
Máquina Virtual Java JDK	1.6.0_22
OpenJDK	1.7 (Build b118)
Ant	1.8.1

3 Instalación del JDK

Previo a la instalación del JDK hay que instalar el sistema operativo y una máquina virtual de Java.

OpenSUSE

El sistema operativo elegido es OpenSuse 11.3. Por detalles de cómo obtener el DVD de instalación, y los diferentes pasos de instalación, referirse a <http://es.opensuse.org/>. La instalación es muy sencilla y así como su interfaz; por lo tanto no nos explayamos al respecto.

Máquina Virtual Java JDK de SUN

Para compilar OpenJDK, es necesario tener instalada una máquina virtual Java JDK, se recomienda la versión 1.5 o superior. En caso de que el sistema operativo utilizado no brinde una se puede descargar la versión 1.6 de la siguiente página:

https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jdk-6u4-b-oth-JPR@CDSCDS_Developer
(Último acceso 10/11/08)

La instalación de la máquina virtual consiste en ejecutar el archivo descargado e instalar la máquina virtual en la ruta que se desee.

Puede ser necesario incluir en la variable PATH la ruta a la máquina virtual. En Linux esto se logra ejecutando el siguiente comando:

```
export PATH=<ruta a la máquina virtual>:$PATH.
```

Este comando puede ser incluido en el archivo `.bashrc` que se encuentra en el directorio `home` del usuario.

Para verificar que la instalación ha sido correcta se puede iniciar una consola y ejecutar el comando:

```
java --version (en Windows java -version).
```

Y verificar que la información desplegada coincide con la versión descargada.

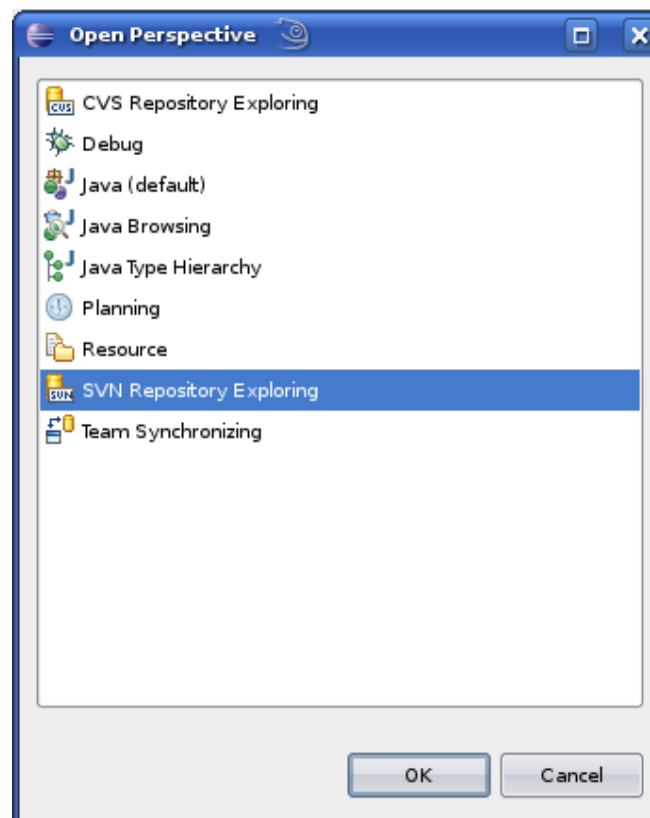
4 Obtención de IP4JVM

Para bajar el proyecto en primer lugar se debe tener un usuario con los permisos adecuados para gestionar el repositorio del proyecto IP4JVM ya que el mismo es un proyecto privado por el momento. El proyecto se encuentra en gforge y su página es la siguiente:

<https://gforge.inria.fr/projects/ip4jvm/>

Los pasos a seguir son:

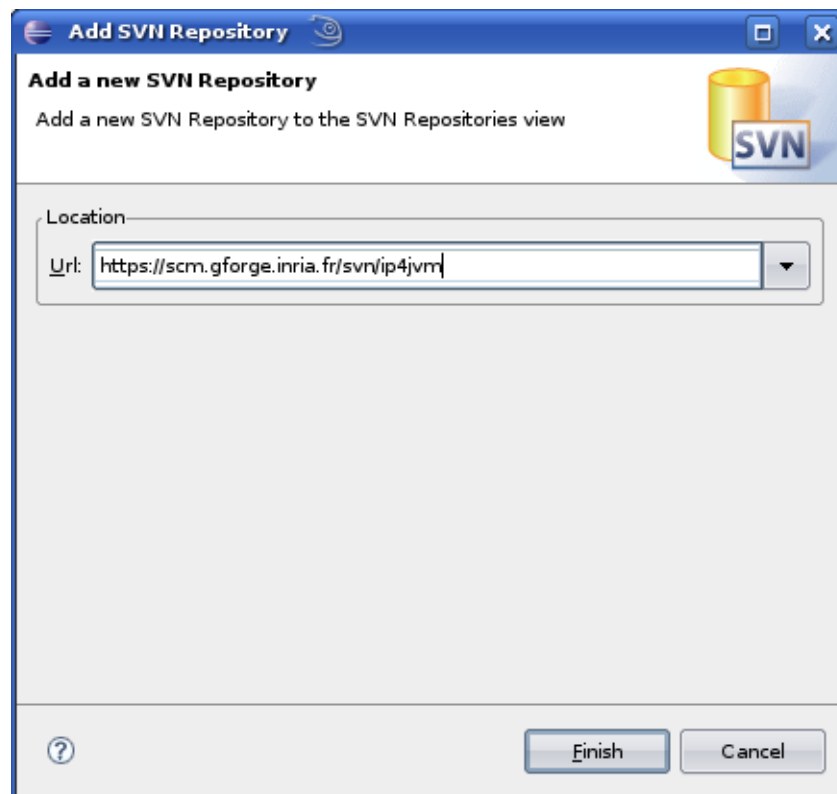
1. En Eclipse ir al menú Window -> Open Perspective -> Others
Seleccionar "SVN Repository Exploring".



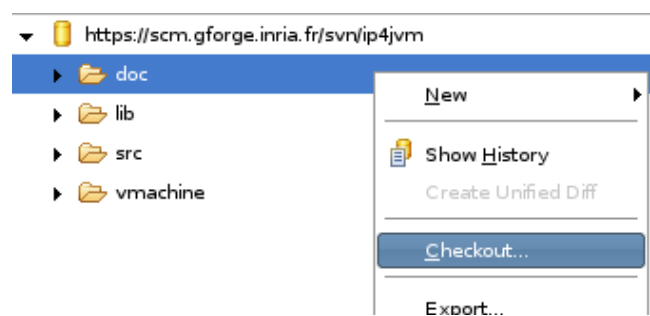
2. Luego de esto, dar clic en el botón que se indica en la siguiente imagen:



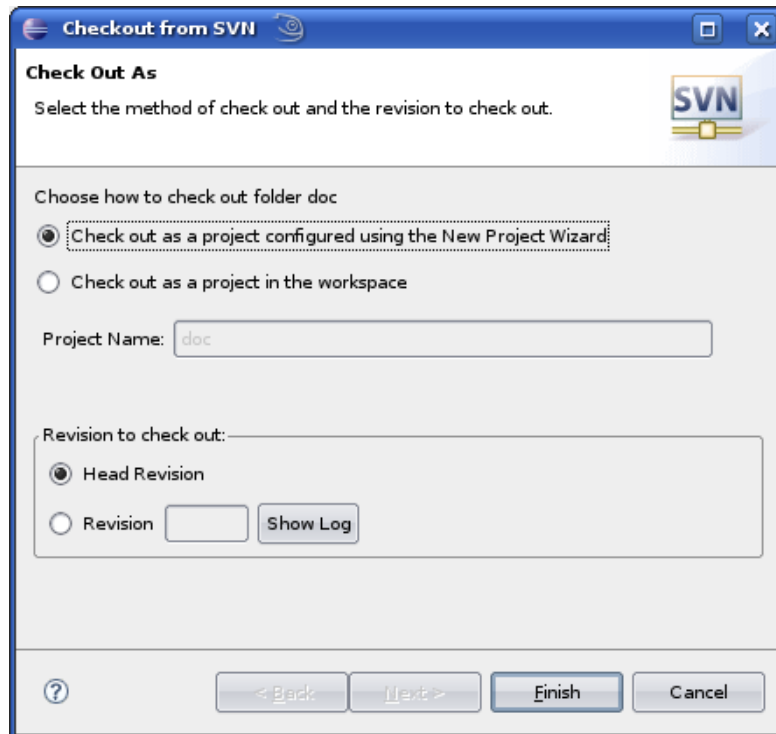
3. Ingresar la ruta “<https://scm.gforge.inria.fr/svn/ip4jvm>” en el campo URL y hacer clic en Finish en la siguiente pantalla:



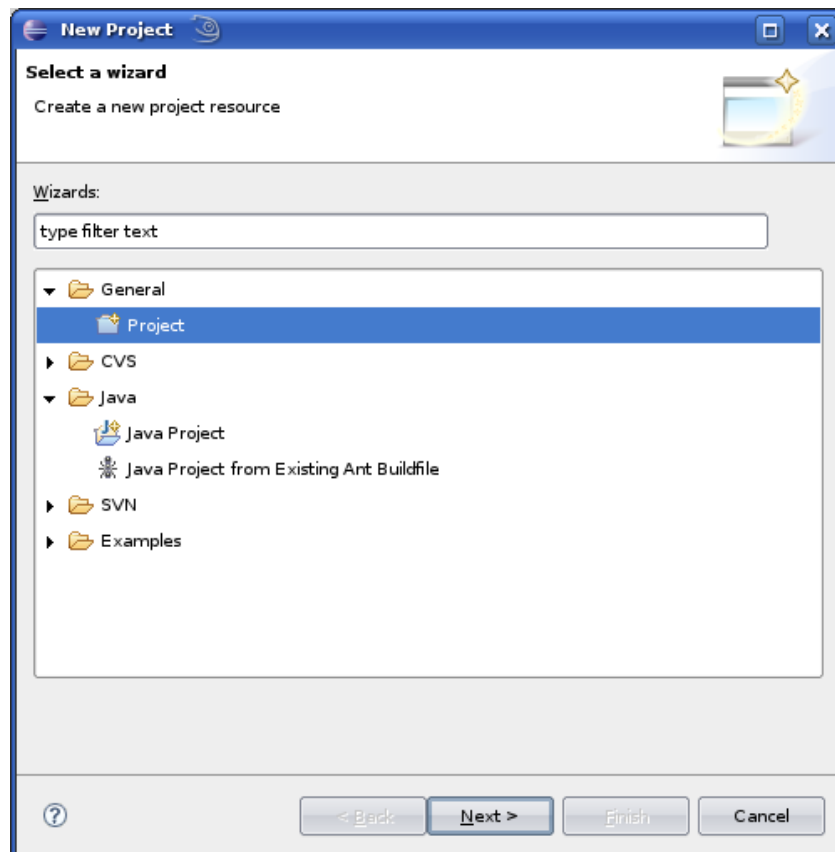
4. Abrir el árbol, dar clic derecho en la carpeta doc y seleccionar Checkout



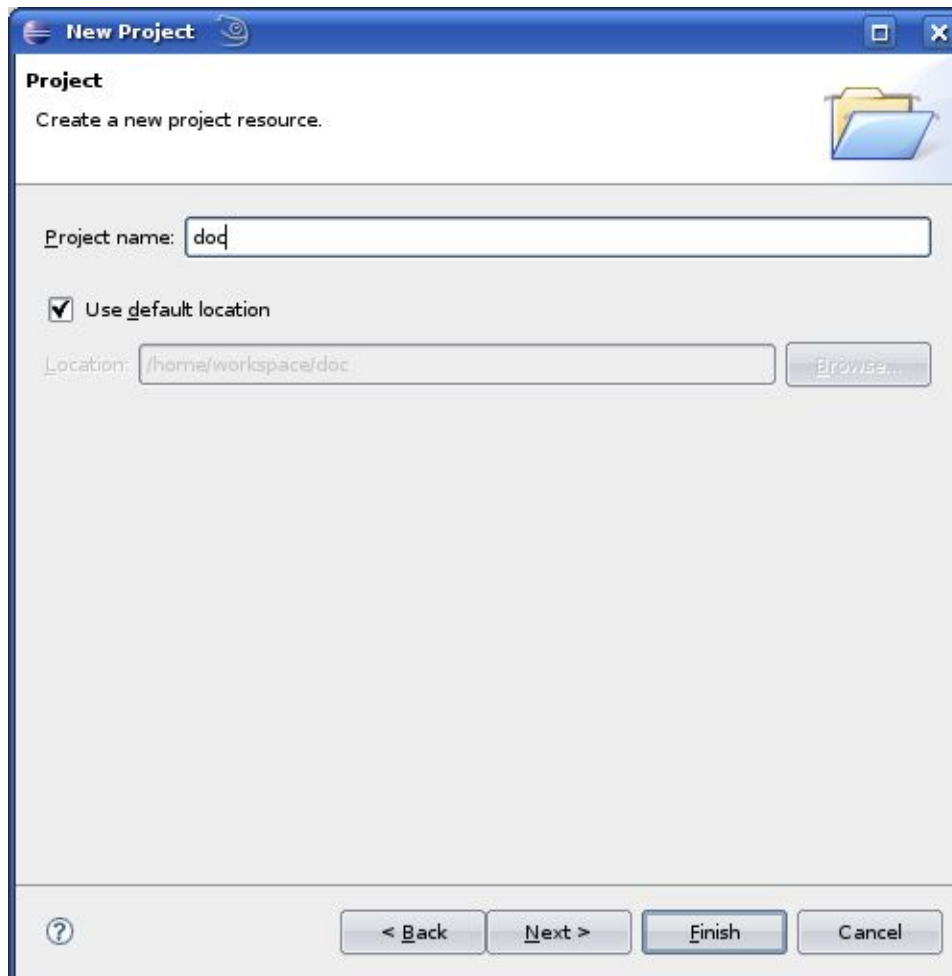
5. En la siguiente ventana hacer clic en Finish:



6. A continuación seleccionar Project y dar clic en Next.

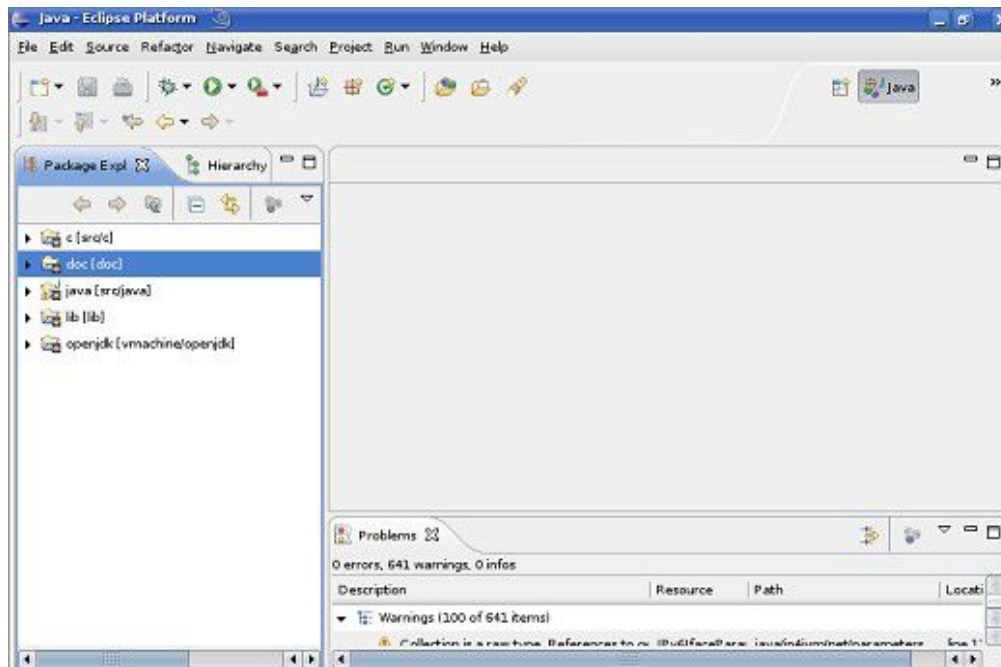


7. En la siguiente ventana ingresar el nombre del proyecto (por ejemplo doc) y hacer clic en Finish.



8. Repetir los pasos a partir del 4 pero esta vez para lib y src/c.

9. Finalmente repetir los pasos desde el 4 al 7 para src/java pero seleccionando tipo de proyecto Java Project. Una vez finalizado el procedimiento, al ingresar a la vista de exploración de paquetes del eclipse, deberían verse todos los proyectos de la siguiente manera:



5 Instalación de ant

Bajar e instalar ant. Luego de bajar el archivo .tar.gz, descomprimirlo en una carpeta a elección (Por ejemplo USER_HOME/Desktop, donde USER_HOME es la ruta al directorio home del usuario) y luego inicializar la variable de entorno ANT_HOME (si se descomprimió en Desktop: export ANT_HOME=USER_HOME/apache). Si se quiere automatizar el proceso de inicialización de la variable se debe ingresar el comando utilizado al final del archivo .bashrc (que se encuentra debajo de la carpeta USER_HOME). Agregar PATH=\$ANT_HOME/bin:\$PATH al archivo .profile que se encuentra en USER_HOME.

Ruta para descargar ant:

<http://ant.apache.org/bindownload.cgi>

(Último acceso 20/11/2010)

Para confirmar la instalación basta con abrir una terminal y obtener la siguiente salida:

```
> ant -version
```

```
Apache Ant version 1.8.1 compiled on April 30 2010
```

6 Instalación de Open JDK + IP4JVM

1. Bajar los archivos fuente (src) y los binary plugs de OpenJDK. En este documento se utilizó la versión 1.7 **b118** (11 de Noviembre de 2010) de OpenJDK.

Ruta para descargar openjdk

<http://download.java.net/openjdk/jdk7/promoted/b118>

(Último acceso 20/11/2010)

2. Si no se tiene instalado JDK 1.6 o superior, instalarlo. Se aconseja instalarlo bajo la carpeta /opt ya que los scripts de compilación de OpenJDK han sido modificados de manera tal de tomar esta carpeta como carpeta de instalación. En este documento se utilizó la versión 1.6.0_22.

Para ello:

- Descargar el archivo .bin correspondiente al jdk 1.6.0_22
- Moverlo a la carpeta opt y dentro de una terminal ejecutar `chmod +x ARCHIVO.BIN`
- Ejecutar `./ARCHIVO.BIN` esto creara la VM
- Editar .bashrc cambiando la variable `JAVA_HOME` y modificando el `PATH`
- `export JAVA_HOME=/opt/jdk1.6.0_22/`
- `export PATH=$JAVA_HOME/bin:$PATH`

3. Se requiere tener instalados todos los paquetes de los cuales depende OpenJDK. Esto se puede realizar con la utilidad de gestión de paquetes que brinda OpenSuse. Los paquetes que son necesarios son:

- cups, cups-devel
- gnu make
- gcc, gcc-c++
- glibc, glibc-devel
- alsa, alsa-devel
- freetype2, freetype2-devel
- zip
- openmotif, openmotif-devel (y sus dependencias)
- libpcap-devel

Puede que no sea necesario instalar todos estos paquetes dependiendo de los que se tengan ya instalados. No se brindan referencias para descargar los paquetes, pues como se mencionó anteriormente los mismos pueden ser descargados con las herramientas de gestión de paquetes que brinda el sistema operativo nombrado con anterioridad.

4. Descomprimir el archivo .zip de los fuentes de OpenJDK donde se desee (en el presente documento se refiere a dicha ruta como `OPENJDK_HOME`).

5. Ejecutar el archivo .jar de los binary plugs correspondiente (en este caso el de Linux) y seleccionar la ruta de instalación que se prefiera. El archivo `jdk_generic_profile.sh` contenido en la carpeta `vmachine/openjdk/make` está modificado de tal manera que los binary plugs deben estar instalados en /opt, en caso de que se instalen en otra ruta se

debe asignar el valor de la ruta a la variable ALT_BINARY_PLUG usada en el script dado.

6. Copiar el directorio src/java/ip4jvm contenido dentro del proyecto IP4JVM al directorio OPENJDK_HOME/jdk/src/share/classes (esta ruta debería existir). Hacer lo mismo con las carpetas src/java/dhcpv6 y src/java/common.

Crear las carpetas OPENJDK_HOME/jdk/make/ip4jvm, OPENJDK_HOME/jdk/make/dhcpv6 y OPENJDK_HOME/jdk/make/common (si no existe)

Crear los archivos OPENJDK_HOME/jdk/make/ip4jvm/Makefile, OPENJDK_HOME/jdk/make/dhcpv6/Makefile y OPENJDK_HOME/jdk/make/common/Makefile

Los archivos deben contener las siguientes líneas (se da como ejemplo para el directorio ip4jvm, lo que luego se cambia por dhcpv6 y common en el directorio respectivo):

```
BUILDDIR = ..  
MODULE = ip4jvm  
PACKAGE = ip4jvm  
include $(BUILDDIR)/common/Defs.gmk  
AUTO_FILES_JAVA_DIRS = ip4jvm  
include $(BUILDDIR)/common/Classes.gmk
```

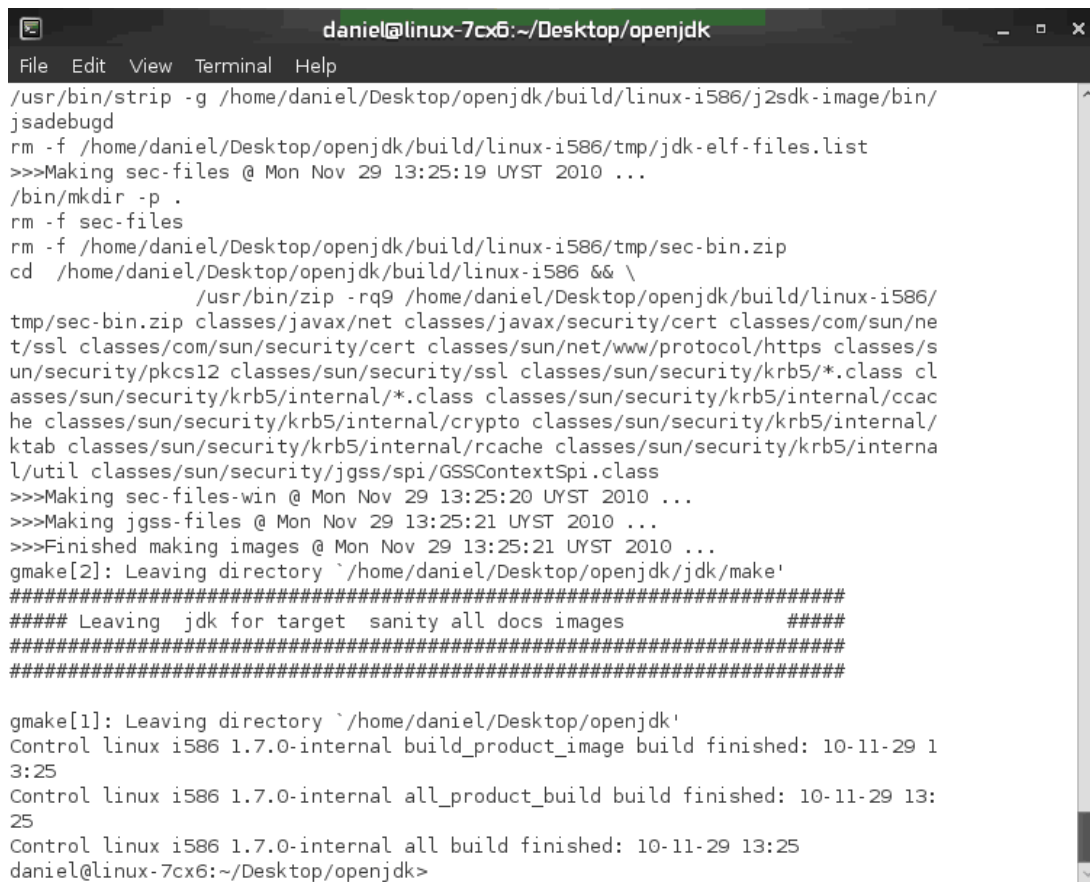
7. Ejecutar en una consola posicionada en la ruta OPENJDK_HOME/make el comando: *gmake*. En alrededor de una hora (dependiendo del hardware) se tendrá todo satisfactoriamente compilado.

NOTA: Es importante comentar que se necesita disponer de 1.4 Gb para realizar la compilación de la VM.

Luego de compilada la máquina virtual, poner la nueva máquina virtual Java en la variable PATH. La carpeta de máquina virtual JDK Java compilada se encuentra en la ruta OPENJDK_HOME/build/linux-i586/j2sdk-image/bin.

Además es necesario setear la variable JAVA_HOME.

Tras compilar satisfactoriamente se debe obtener la siguiente salida



```
daniel@linux-7cx6:~/Desktop/openjdk
File Edit View Terminal Help
/usr/bin/strip -g /home/daniel/Desktop/openjdk/build/linux-i586/j2sdk-image/bin/
jsadebugd
rm -f /home/daniel/Desktop/openjdk/build/linux-i586/tmp/jdk-elf-files.list
>>>Making sec-files @ Mon Nov 29 13:25:19 UYST 2010 ...
/bin/mkdir -p .
rm -f sec-files
rm -f /home/daniel/Desktop/openjdk/build/linux-i586/tmp/sec-bin.zip
cd /home/daniel/Desktop/openjdk/build/linux-i586 && \
    /usr/bin/zip -rq9 /home/daniel/Desktop/openjdk/build/linux-i586/
tmp/sec-bin.zip classes/javax/net classes/javax/security/cert classes/com/sun/ne
t/ssl classes/com/sun/security/cert classes/sun/net/www/protocol/https classes/s
un/security/pkcs12 classes/sun/security/ssl classes/sun/security/krb5/*.class cl
asses/sun/security/krb5/internal/*.class classes/sun/security/krb5/internal/ccac
he classes/sun/security/krb5/internal/crypto classes/sun/security/krb5/internal/
ktab classes/sun/security/krb5/internal/rcache classes/sun/security/krb5/interna
l/util classes/sun/security/jgss/spi/GSSContextSpi.class
>>>Making sec-files-win @ Mon Nov 29 13:25:20 UYST 2010 ...
>>>Making jgss-files @ Mon Nov 29 13:25:21 UYST 2010 ...
>>>Finished making images @ Mon Nov 29 13:25:21 UYST 2010 ...
gmake[2]: Leaving directory `/home/daniel/Desktop/openjdk/jdk/make'
#####
#### Leaving jdk for target sanity all docs images #####
#####
#####

gmake[1]: Leaving directory `/home/daniel/Desktop/openjdk'
Control linux i586 1.7.0-internal build_product_image build finished: 10-11-29 1
3:25
Control linux i586 1.7.0-internal all_product_build build finished: 10-11-29 13:
25
Control linux i586 1.7.0-internal all build finished: 10-11-29 13:25
daniel@linux-7cx6:~/Desktop/openjdk>
```

8. Ir a la carpeta *c* del proyecto IP4JVM y en una consola (posicionada en esta ruta) correr los comandos “make -f makefile.read install” y “make -f makefile.write install”.

Luego de esto deberían crearse 2 archivos llamados libnetmanagerread.so y libnetmanagerwrite.so bajo la carpeta lib.

En el caso de que se haya instalado la máquina virtual base (JDK 1.6) en otro directorio diferente de /opt se deberá modificar el valor asignado a la variable JAVA_HOME en estos archivos antes de compilarlos.

9. Como último paso copiar los archivos libnetmanagerread.so y libnetmanagerwrite.so a la carpeta OPENJDK_HOME/build/linux-i586/j2sdk-image/jre/lib/i386.

7 Test de la VM modificada

Luego de compilada la máquina virtual y de inicializada la variable de entorno PATH de manera tal que apunte a la nueva máquina virtual compilada, se puede compilar y correr el programa testPing6v2.

```
javac -bootclasspath OPENJDK_JRE_HOME/lib/rt.jar -encoding utf-8  
test/testPing6v2.java
```

Para correr testPing6v2 se debe ejecutar en la línea de comandos el siguiente comando, cambiando la dirección de destino del ping: “java testPing6v2 ping6 eth0 ::1”. Este programa solo verificará que las clases del stack IP4JVM se encuentran en el classpath por defecto.

Si no se obtiene ningún error, entonces se ha integrado exitosamente IP4JVM a OPENJDK.

8 Solución de errores

8.1 Error de Jaxp

En caso de obtener este error:

BUILD FAILED

/home/daniel/Desktop/openjdk/jaxp/build-defs.xml:70: ERROR: Cannot find source for project jaxp.

HINT: Try setting drops.dir to indicate where the bundles can be found, or try setting the ant property allow.downloads=true to download the bundle from the URL.

e.g. ant -Dallow.downloads=true -OR- ant -Ddrops.dir=some_directory

- a. Descargar la última versión de JAX-WS (la usada en este documento fue jaxp-1_4_4), en un cierto directorio (suponemos USER_HOME para este documento)
- b. Editar OPENJDK_HOME/jaxp/make/Makefile cambiando las líneas que se muestran a continuación.

```
ifdef ALT_DROPS_DIR
    DROPS_DIR = $(ALT_DROPS_DIR)
else
    DROPS_DIR = USER_HOME
endif
```

8.2 Error de Jaxws

En caso de obtener este error:

BUILD FAILED

/home/daniel/Desktop/openjdk/jaxws/build-defs.xml:70: ERROR: Cannot find source for project jaxws.

HINT: Try setting drops.dir to indicate where the bundles can be found, or try setting the ant property allow.downloads=true to download the bundle from the URL.

e.g. ant -Dallow.downloads=true -OR- ant -Ddrops.dir=some_directory

- a. Descargar la última versión de JAX-WS (la usada en este documento fue la jdk7-jaf-2010_08_19.zip disponible en <http://java.net/downloads/jax-ws/JDK7/>)
- b. Editar OPENJDK_HOME/jaxws/make/Makefile

```
ifdef ALT_DROPS_DIR
```

```
DROPS_DIR = $(ALT_DROPS_DIR)
else
DROPS_DIR = USER_HOME
endif
```

8.3 Error de pcap.h

Si obtiene el siguiente error:

```
ip4jvm_javafwrk_NetManagerRead.c:7:18: fatal error: pcap.h: No such file or directory
```

Está faltando instalar el paquete pcap-devel.

8.4 Error de jni.h

Si obtiene el siguiente error:

```
ip4jvm_javafwrk_NetManagerRead.c:1:17: fatal error: jni.h: No such file or directory
```

Esta faltando cambiar la ruta JAVA_HOME en makefile.read y makefile.write.

8.5 Error de la carpeta lib

Si obtiene el siguiente error:

```
cp: cannot create regular file `../lib/libnetmanagerread.so': No such file or directory
```

Hay que crear la carpeta ../lib (mkdir ../lib)

9 Glosario

Ant: es una herramienta usada en programación para la realización de tareas de compilación

Gforge: es un software libre basado en la Web para la gestión colaborativa de proyectos de software creada originalmente para SourceForge

Jaxp(Java Api for XML Processing): API Java que sirve para la manipulación y el tratamiento de archivos XML. Necesario para la compilación de OpenJDK.

Jaxws (Java Api for XML Web Services): API Java que se utiliza para la creación de Web Services. Necesario para la compilación de OpenJDK.

JDK (Java Development Kit): es un conjunto de herramientas de desarrollo para la creación de programas en Java.

Open JDK: es la versión libre de la plataforma de desarrollo Java.

OpenSuse: es una distribución de un sistema operativo basado en Linux.

SVN (Subversion): es un sistema de control de versiones. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser el nombre de la herramienta utilizada en la línea de comandos.