

Cómo instalar Java SDK en Ubuntu 9.10 (Karmic Koala)

Juan-Antonio Martínez-Comeche

- **Problema:** En muchas ocasiones se necesita tener instalada una Máquina Virtual Java para poder ejecutar aplicaciones o software basados en el lenguaje de programación Java. La Máquina Virtual Java (JVM) es un programa independiente de la plataforma (Windows, Linux, Mac OS...) que permite ejecutar instrucciones escritas en el código binario de Java (el Java bytecode). Si, con el tiempo, deseamos también modificar el código de esas aplicaciones para adaptarlas a nuestras necesidades específicas, o bien queremos desarrollar programas en Java, una opción aceptable consiste en instalar Java SDK (Java Software Development Kit), pues incluye, además del JRE (Java Runtime Environment) que permite que corran las aplicaciones o software desarrollados en Java, un grupo de utilidades que son de gran ayuda a la hora de programar. Sin embargo, resulta conveniente configurar correctamente algunas variables de entorno para lograr que este programa de código libre cedido por Sun sea perfectamente operativo.
- **Solución:** Para elaborar el procedimiento que figura a continuación de instalación y configuración de la Máquina Virtual Java de Sun (Java SDK versión 6) en Ubuntu 9.10 (Karmic Koala), acudiendo directamente al sitio web de Sun, descargando la última versión disponible y logrando su correcto funcionamiento, se han consultado las siguientes páginas:
 - <http://www.ubuntu-es.org/node/55/>
 - <http://www.kevinboone.com/classpath.html/>
 - <http://www.guia-ubuntu.org/index.php?title=java>
 - <http://java.sun.com/javase/6/webnotes/install/jdk/install-linux.html>
- **Otra Solución:** En este mismo sitio figura otro procedimiento de instalación y configuración de la Máquina Virtual Java de Sun, siguiendo un procedimiento mucho más sencillo y seguro, consistente en descargarse la versión de Sun disponible en los repositorios de Ubuntu, lo que implica que no coincide con la última versión disponible en el sitio web de Sun. Si no considera necesario instalar la última versión de JDK, puede consultar http://www.comeche.es/documents/Software/como_instalar_java_sdk_ubuntu_804.pdf

Proceso de Instalación:

- **Paso 1:** Comprobar que están habilitados los repositorios multiverse en el archivo /etc/apt/sources.list. Para ello, en Sistema → Administración → Orígenes del software, en la pestaña “Software de Ubuntu”, debe estar marcada la opción “Software restringido por copyright o cuestiones legales (multiverse)”.
- **Paso 2:** Descargar la última versión de Java SDK de la página de Sun <http://java.sun.com/javase/downloads/index.jsp>. En esta página, dentro de la sección “Java SE Development Kit (JDK)”, pinchar en “Download”. En la ventana que surge, pinchar en “Continue”. A continuación, pinchar en la última versión para Linux con extensión “.bin” disponible, en nuestro caso, jdk-6u17-linux-i586.bin, y lo descargamos en el Escritorio.
- **Paso 3:** Comprobar que está instalado el paquete “java-package”, bien con apt-get o con el programa Synaptic. Con este último, por ejemplo, elegimos Sistema → Administración → Gestor de paquetes Synaptic. En la ventana que surge, elegimos Editar → Buscar. En el área de texto, introducimos “java-package” y pinchamos en “Buscar”. Si está instalado, aparecerá con el cuadrado a la izquierda del paquete en verde. Si no está instalado, aparecerá con el cuadrado a la izquierda del paquete en blanco. Para instalarlo, emplead “apt-get install” o el propio Synaptic. Con este último, por ejemplo, se marca con el ratón el cuadrado a la izquierda del paquete “java-package”, y en el menú “Paquete” se selecciona “Marcar para instalar”. Si deben instalarse paquetes adicionales aparecerán en la nueva ventana que surge. En ese caso, pinchad en “Marcar”. A continuación, haced clic en “Aplicar”. En la ventana de confirmación, pinchad en “Aplicar”. Este programa permite ejecutar “make-jpkg archivobinario.bin”, creando un archivo “.deb” que contiene Java SDK y que emplearemos para realizar una instalación limpia en nuestro sistema.
- **Paso 4:** Trasladar el archivo jdk-6u17-linux-i586.bin al directorio donde vamos a instalarlo. Por defecto, lo instalaremos donde suele hacerse normalmente, esto es, en /usr/lib/jvm. Para ello, en primer lugar, debemos comprobar que el directorio /usr/lib tiene todos los permisos otorgados al usuario habitual del sistema, en nuestro caso “juan”, y no a “root”. Para comprobarlo, en un terminal se teclaea:

```
# cd /usr  
# ls -l
```

Tanto el propietario como el grupo del directorio “lib” debe ser “juan” y no otro como “root”. Es decir, debe aparecer una línea como la siguiente:

```
drwxr-xr-x 202 juan juan 69632 2009-11-08 19:18 lib
```

Si en lugar de “juan” aparece “root”, tendremos que modificarlo. Para ello, en un terminal se teclaea:

```
# sudo chown -R juan:juan /usr/lib
```

Una vez comprobado de nuevo (mediante “ls -l”) que el directorio /usr/lib tiene todos los permisos otorgados a “juan”, creamos el subdirectorio /usr/lib/jvm. Para ello, en un terminal tecleamos:

```
# mkdir /usr/lib/jvm
```

Ahora podemos trasladar el archivo recién descargado allí. Para ello, basta teclear en un terminal:

```
# mv /home/juan/Escritorio/jdk-6u17-linux-i586.bin /usr/lib/jvm/
```

- **Paso 5:** Comprobar que este archivo binario que acabamos de trasladar a su directorio de instalación /usr/lib/jvm tiene permiso de ejecución. Para ello, en un terminal se teclaea:

```
# chmod +x /usr/lib/jvm/jdk-6u17-linux-i586.bin
```

Si todo ha ido bien, tecleamos en un terminal:

```
# cd /usr/lib/jvm  
# ls -l
```

y debemos obtener una línea como la siguiente:

```
-rwxr-xr-x 1 juan juan 81792634 2009-11-08 19:04 jdk-6u17-linux-i586.bin
```

- **Paso 6:** Ir al directorio donde tenemos el archivo binario (/usr/lib/jvm) y allí ejecutamos “make-jpkg”:

```
# cd /usr/lib/jvm  
# make-jpkg jdk-6u17-linux-i586.bin
```

Contestad “Y” a la primera pregunta y pulsad “Return”. Una vez leída la licencia, compuesta de 13 puntos y suplementos A-I, tecleamos “yes” para señalar que la aceptamos. Veremos en pantalla el aviso “Java™ SE Development Kit 6 successfully installed”. Presionamos “Enter” para continuar. Una vez concluido el proceso, podemos leer el mensaje “Removing temporary directory: done”. Este proceso ha creado en el directorio /usr/lib/jvm el archivo “.deb” correspondiente, en nuestro caso, “sun-j2sdk1.6_1.6.0+update17_i386.deb”

- **Paso 7:** Mediante el comando “dpkg” instalamos el paquete “.deb” creado anteriormente. En nuestro caso:

```
# sudo dpkg -i /usr/lib/jvm/sun-j2sdk1.6_1.6.0+update17_i386.deb
```

Empleamos “sudo” porque esta operación precisa privilegios de superusuario.

Puede que recibamos el siguiente mensaje final: “Se encontraron errores al procesar: sun-j2sdk1.6” No importa. Posteriormente volveremos a instalar el programa para solucionarlo. Este proceso ha creado la carpeta “j2sdk1.6-sun” en /usr/lib

- **Paso 8:** Instalar el paquete creado con sus extensiones. Para ello, teclear en un terminal:

```
# sudo apt-get install sun-j2sdk1.6
```

Puede que recibamos el siguiente mensaje final: “Se encontraron errores al procesar: sun-j2sdk1.6” No importa. A continuación volveremos a instalar el programa para solucionarlo.

- **Paso 9:** Instalar de nuevo el programa SDK. Ello porque en muchos casos no instala bien el programa hasta que no están instaladas las extensiones; sin embargo, las extensiones no se pueden instalar sin el SDK. De ahí que haya, por seguridad, que instalar “2 veces” el SDK. Podemos hacerlo siguiendo el Paso 7 anterior, o bien podemos emplear el archivo binario inicial. Basta ir al directorio /usr/lib/jvm donde está el archivo binario y ejecutarlo. En un terminal, pues, teclear:

```
# cd /usr/lib/jvm  
# ./jdk-6u17-linux-i586.bin
```

En este último comando, no olvidar el punto y la barra iniciales (./) De nuevo se muestra la licencia. Una vez leída, tecleamos “yes” para señalar que la aceptamos. Aparecerá el mensaje “Java™ SE Development Kit 6 successfully installed”. Presionamos “Enter” para continuar. Se carga la página de registro de Sun. Cerramos el navegador, con el fin de registrarnos más tarde. Una vez concluido el proceso, podemos leer el mensaje “Done”. El programa se ha instalado finalmente en el subdirectorio /usr/lib/jvm/jdk1.6.0_17

- **Paso 10:** Borrar los archivos “.bin” y “.deb” Para ello, en un terminal, teclear:

```
# rm /usr/lib/jvm/jdk-6u17-linux-i586.bin  
# rm /usr/lib/jvm/sun-j2sdk1.6_1.6.0+update17_i386.deb
```

- **Paso 11:** Comprobar, mediante el gestor de paquetes Synaptic, que el programa sun-j2sdk1.6 está perfectamente instalado. Para ello, en Sistema → Administración → Gestor de paquetes Synaptic, seleccionamos Editar → Buscar, e introducimos sun-j2sdk1.6 Tras pinchar en “Buscar” deberá aparecer dicho paquete con el cuadrado a la izquierda en verde y a la derecha con la versión recién instalada, en nuestro caso, “1.6.0+update17”.
- **Paso 12:** Establecer el Java SDK como nuestra Máquina Virtual por defecto. Para ello instalamos previamente el paquete “galternatives”:

```
# sudo aptitude install galternatives
```

Contestar “Y” cuando pregunte. Recibiremos el siguiente mensaje final: “Escribiendo información de estado extendido... Hecho”. También se puede instalar con Synaptic, siguiendo un proceso similar al señalado en el Paso 3.

- **Paso 13:** Lanzar el programa galternatives:

```
# sudo galternatives
```

- **Paso 14:** En el menú gráfico que obtendremos, en el marco izquierdo seleccionamos “java” y en el marco derecho seleccionamos la nueva versión de Java si aparecen varias, esto es, `/usr/lib/j2sdk1.6-sun/bin/java`. A continuación, cerramos el programa (Fichero → Salir).
- **Paso 15:** Comprobar que estamos usando la versión correcta, esto es, la recién instalada. Para ello, basta teclear en un terminal:

```
# sudo java -version
```

y obtendremos:

```
java version “1.6.0_17”  
Java™ SE Runtime Environment (build.....)  
Java HotSpot™ Server VM (build .....
```

- **Paso 16:** La variable de entorno `JAVA_HOME` nunca es configurada por ningún JDK. Hay muchos programas que, sin embargo, necesitan tener configurada la variable `JAVA_HOME` para poder ser instalados correctamente. En consecuencia, conviene configurar dicha variable adecuadamente. Para ello debemos anotar en qué directorio se ha instalado la Máquina Virtual Java, esto es, en el que se halla el Java SDK. En nuestro caso, `“/usr/lib/j2sdk1.6-sun”`. Ésta será la ruta que deberá poseer dicha variable.
- **Paso 17:** La variable de entorno `CLASSPATH` no es tampoco configurada durante la instalación previamente realizada. Sin embargo, dicha variable debe contener la ruta de los directorios en los que el compilador de Java (`javac.exe`) podrá localizar los archivos `.java` (código fuente de programas Java) para compilar, así como cualquier otro archivo `class` (programa ya compilado) referenciado o empleado en el programa fuente. De ahí que en `CLASSPATH` deban figurar también las rutas de las librerías que se empleen en nuestros programas. Si se desean señalar varias rutas, deberemos separarlas en Linux mediante `‘:’` (dos puntos). Además, conviene terminar las rutas de `CLASSPATH` siempre con un punto `‘.’` (punto), pues ello nos permitirá compilar y/o ejecutar cualquier programa en Java situado en el directorio donde trabajemos en ese momento. En consecuencia, en nuestro caso, incluiremos en este ejemplo tres rutas en nuestra variable `CLASSPATH`:
 - El directorio donde se hallan las librerías de Java de la Máquina Virtual recién instalada: `/usr/lib/j2sdk1.6-sun/lib`
 - El directorio donde vayamos a incluir los programas Java que desarrollemos en

- el futuro, por ejemplo: /home/juan
- El directorio actual, representado mediante un punto

En consecuencia, CLASSPATH deberá figurar en nuestro caso de la siguiente manera: "/usr/lib/j2sdk1.6-sun/lib:/home/juan:."

- **Paso 18:** Para el establecimiento de estas variables de entorno, basta modificar el archivo ".bashrc", localizado y oculto en el directorio raíz de cada usuario. Por ejemplo: /home/juan/.bashrc, donde "juan" es el nombre de nuestra carpeta de usuario. Abrimos con gedit el archivo ".bashrc" tecleando "sudo gedit /home/juan/.bashrc".
- **Paso 19:** Al final del archivo añadimos las siguientes líneas de texto:
JAVA_HOME='/usr/lib/j2sdk1.6-sun'
CLASSPATH='/usr/lib/j2sdk1.6-sun/lib:/home/juan:.'
export CLASSPATH
export JAVA_HOME
PATH=\$JAVA_HOME/bin:\$PATH
export PATH

La tercera variable de entorno PATH tiene una finalidad importante. De hecho, para poder ejecutar programas Java desde cualquier carpeta, la carpeta con los archivos "javac.exe" y "java.exe" debe estar incluida en el PATH del sistema.

- **Paso 20:** Una vez añadidas estas líneas al final del archivo, guardamos los cambios. De esta manera, hemos instalado y configurado la Máquina Virtual Java adecuadamente.
- **Paso 21:** Es importante reiniciar el sistema para que todas las variables de entorno adquieran los valores recién introducidos. Para comprobar que todo está correcto, abrimos de nuevo un terminal y tecleamos "echo \$JAVA_HOME". Deberá darnos la ruta introducida anteriormente. De igual forma, si tecleamos "echo \$CLASSPATH" deberá mostrar las rutas tecleadas previamente.