



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO
LICENCIATURA EN INFORMÁTICA ADMINISTRATIVA



SISTEMAS OPERATIVOS DE RED

REPORTE DE PRÁCTICA - INSTALACIÓN DEL JDK DE JAVA



Docente: LIA. ELIZABETH EVANGELISTA NAVA

Discente:

Karen Sidery Yazbeck Garduño Miranda

6TO. SEMESTRE LIA GRUPO: I8
PERIODO 2014A (CICLO ESCOLAR 2013-2014)

FECHA: 11 de Marzo del 2014.

PROPÓSITO:

El alumno pondrá en práctica algunos comandos adquiridos durante la sesión teórica e instalará el jdk de java, posteriormente creará un archivo fuente y lo ejecutará

ALCANCES:

El alumno sabrá asignar permisos, crear un código fuente mediante un editor de textos, y ejecutarlo con JAVA

REQUERIMIENTOS:

- ✓ Sistema operativo Linux
- ✓ Jdk versión
- ✓ Código fuente

DESARROLLO

En la Fig1. Se muestra por ir creando una carpeta denominada "java" para guardar todos los archivos necesarios del jdk a instalar para tener un orden en caso de usarlos posteriormente.



```
root@localhost:~# ls
anaconda-ks.cfg  Documentos  install.log  Plantillas
descargas        Escritorio  install.log.xptlog  Pkgs
Descargas        Descargas  musica       videos
root@localhost:~# cd /usr/local
root@localhost:local# ls
bin  etc  games  include  lib  libexec  sbin  share  src
root@localhost:local# mkdir java
root@localhost:local# ls
bin  etc  games  include  java  lib  libexec  sbin  share  src
root@localhost:local#
```

Fig1. Se direcciona al directorio USR/LOCAL y se crea la carpeta llamada java

Enseguida se copia el archivo que se va a utilizar a la carpeta de java, utilizando del comando cp_nombre del archivo seguido de la direccion a mover (usr/local/java) y para verificar dicha accion nos dirigimos a la carpeta java y se busca el archivo, como se muestra en la Fig2.



```
root@localhost:~# cp /usr/local/jdk-7u45-linux-x64.gz /usr/local/java
root@localhost:~# cd /usr/local/java
root@localhost:local# ls
jdk-7u45-linux-x64.gz
root@localhost:local# tar -xzf jdk-7u45-linux-x64.gz
```

Fig2. Ubicación del archivo jdk

Antes de trabajar con el archivo se le tienen que asignar permisos para que el archivo se pueda manejar así que nos dirigimos a la carpeta java y con el comando `chmod 777_nombredelarchivo.gz` se le asigna dichos permisos y pero también hay que descomprimir el archivo con el comando `tar zxvf nombredelarchivo.gz`, como se muestra la Fig3.

```
root@localhost:~# cd /usr/local/java
root@localhost:~/local/java# ls
ARCHIVOS
root@localhost:~/local/java# cd ARCHIVOS
root@localhost:~/local/java/ARCHIVOS# ls
jdk-7u45-linux-64.gz  jdk-7u45-linux-64-gz-install.tar.gz
root@localhost:~/local/java/ARCHIVOS# cd ..
root@localhost:~/local/java# ls
binanda-ig.cfg  binanda-ig-install.log  binanda-ig-install.log  binanda-ig-install.log  binanda-ig-install.log  binanda-ig-install.log
binanda-ig.cfg  binanda-ig-install.log  binanda-ig-install.log  binanda-ig-install.log  binanda-ig-install.log  binanda-ig-install.log
root@localhost:~/local/java# cd /usr/local/java
root@localhost:~/local/java# ls
jdk-7u45-linux-64.gz
root@localhost:~/local/java# chmod 777 jdk-7u45-linux-64.gz
root@localhost:~/local/java# tar zxvf jdk-7u45-linux-64.gz
```

Fig3. Se le asignan permisos y se descomprime el archivo

Se inicia el proceso de descompresión del archivo como se muestra en la Fig4, esperamos a que termine.

```
root@localhost:~/local/java# tar zxvf jdk-7u45-linux-64.gz
jdk1.7.0_45/man/man1/jstack.1
jdk1.7.0_45/man/man1/polysync.1
jdk1.7.0_45/man/man1/jstat.1
jdk1.7.0_45/man/man1/hsperfdata.1
jdk1.7.0_45/man/man1/jpsweb.1
jdk1.7.0_45/man/man1/jpswebcrpt.1
jdk1.7.0_45/man/man1/jpsweb.1
jdk1.7.0_45/man/man1/schemagen.1
jdk1.7.0_45/man/man1/jhat.1
jdk1.7.0_45/man/man1/java.1
jdk1.7.0_45/man/man1/jps.1
jdk1.7.0_45/man/man1/xjc.1
jdk1.7.0_45/man/man1/jarsigner.1
jdk1.7.0_45/man/man1/appletviewer.1
jdk1.7.0_45/man/man1/javaFXpackageger.1
jdk1.7.0_45/man/man1/pack200.1
jdk1.7.0_45/man/man1/keytool.1
jdk1.7.0_45/man/man1/javac.1
jdk1.7.0_45/man/man1/jstatd.1
jdk1.7.0_45/man/man1/javadoc.1
jdk1.7.0_45/THIRDPARTYLICENSEREADME.txt
jdk1.7.0_45/COPYRIGHT
root@localhost:~/local/java#
```

Fig4. Proceso de descompresión del archivo jdk

Al terminar el proceso se observan dos archivos, de los cuales solo utilizaremos el que es de color azul y el de color verde (gz) se borra con el comando `rm_nombredelarchivo` para tener mas espacio, como en la Fig5.

```
root@localhost:~/local/java# ls
jdk1.7.0_45/man/man1/javac.1
jdk1.7.0_45/man/man1/schemagen.1
jdk1.7.0_45/man/man1/jhat.1
jdk1.7.0_45/man/man1/java.1
jdk1.7.0_45/man/man1/jps.1
jdk1.7.0_45/man/man1/jarsigner.1
jdk1.7.0_45/man/man1/appletviewer.1
jdk1.7.0_45/man/man1/javaFXpackageger.1
jdk1.7.0_45/man/man1/pack200.1
jdk1.7.0_45/man/man1/keytool.1
jdk1.7.0_45/man/man1/javac.1
jdk1.7.0_45/man/man1/jstatd.1
jdk1.7.0_45/man/man1/javadoc.1
jdk1.7.0_45/THIRDPARTYLICENSEREADME.txt
jdk1.7.0_45/COPYRIGHT
root@localhost:~/local/java# ls
jdk1.7.0_45  jdk-7u45-linux-64.gz
root@localhost:~/local/java# rm jdk-7u45-linux-64.gz
rm: ¿borrar el fichero regular «jdk-7u45-linux-64.gz»? (s/n) y
root@localhost:~/local/java# ls
jdk1.7.0_45
root@localhost:~/local/java#
```

Fig5. Se muestran los archivos de la carpeta java y el borrado de un archivo

Para corregir el error que se mostró al compilar se debe de editar el archivo hola.java con el comando "gedit_nombredelarchivo.java" como en la pantalla de la Fig15. Esta edición se debe de hacer desde la dirección /usr/local/java/jdk1.7.0_45/bin.



Fig15. Abrir el editor

Ya abierto el archivo aparecerá así como en la Fig16. y lo que se le debe de modificar es la clase es decir, en vez de decir "class Test" se cambia por el nombre del archivo "class hola" como en la Fig17.

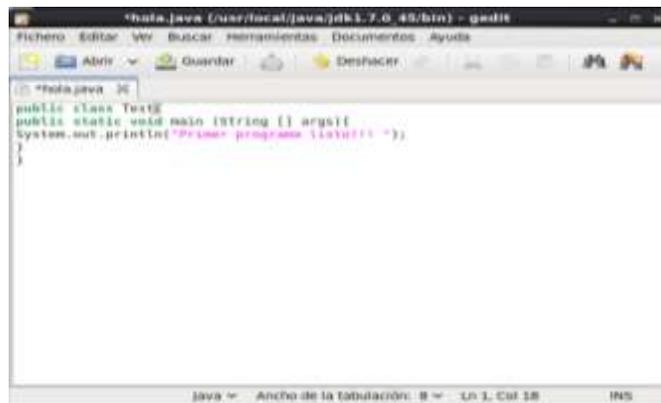


Fig16. Archivo hola.java abierto para modificar



Fig17. Archivo hola.java modificado

Posteriormente nos dirigimos al archivo para a completar la solución del error con la ruta /root/.bashrc



Fig18. Editar las variables de entorno

El archivo que se abre es como el d la Fig19. Y se debe de modificar las variables de entorno anexando dos líneas de código al final de este archivo como se muestra en la Fig20 y se le da click en el boton Guardar-Se reinicia el sistema para que se guarden las modificaciones.

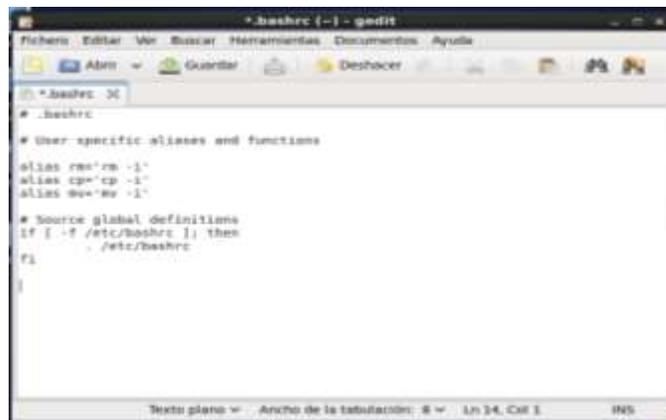


Fig19. Archivo bashrc

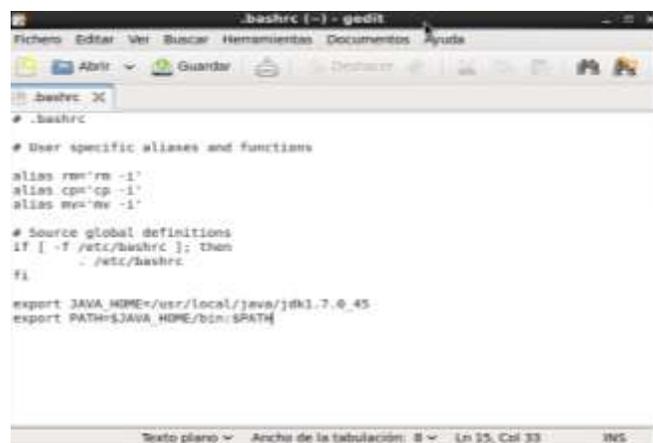


Fig20. Editar el archivo bashrc -Guardar

Para que el archivo corra sin problemas le debemos conceder permisos, en la Fig21. se muestra el comando `chmod 777 nombredelarchivo`.



```
root@localhost:usr/local/java/jdk1.7.0_45/bin
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost ~]# cd /usr/local/java/jdk1.7.0_45/bin
[root@localhost bin]# ls
appletviewer  java          | console      | runscript    | pack200      | unpack200
apt           | javac        | control      | sashbugd     | policytool   | wgen
ControlPanel | javadoc     | db           | stack        | rmic         | winprint
extcheck     | javafpackager | fat         | stat         | rmid        | xjc
hola.java    | javah       | info        | statd        | rmiregistry  |
hola.java-   | javap       | jsp         | visualview   | schwigen     |
idl          | java-rmi.cgi | jsc         | keytool     | serialver    |
jar          | jarman      | jsc.ini     | native2ascii | servertool   |
jarsigner   | jcmd        | jps         | orbd        | tnameserv
[root@localhost bin]# chmod 777 hola.java
```

Fig21. Conceder permisos al archivo

Para finalizar se ejecuta el archivo `hola.java` llamando al compilador `javac` seguido `nombredelarchivo` para compilar el archivo, desde se escribe `java hola` ejecutar el archivo `hola.java` y enseguida aparecerá el texto en pantalla "Primer programa listo!!!" despues se le da `ls` para verificar que se generó el archivo objeto del archivo denominado `hola.class` de color negro, como se muestra en la Fig.22



```
root@localhost:usr/local/java/jdk1.7.0_45/bin
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost ~]# cd /usr/local/java/jdk1.7.0_45/bin
[root@localhost bin]# ls
appletviewer  java          | console      | runscript    | pack200      | unpack200
apt           | javac        | control      | sashbugd     | policytool   | wgen
ControlPanel | javadoc     | db           | stack        | rmic         | winprint
extcheck     | javafpackager | fat         | stat         | rmid        | xjc
hola.java    | javah       | info        | statd        | rmiregistry  |
hola.java-   | javap       | jsp         | visualview   | schwigen     |
idl          | java-rmi.cgi | jsc         | keytool     | serialver    |
jar          | jarman      | jsc.ini     | native2ascii | servertool   |
jarsigner   | jcmd        | jps         | orbd        | tnameserv
[root@localhost bin]# chmod 777 hola.java
[root@localhost bin]# javac hola.java
[root@localhost bin]# java hola
Primer programa listo!!!
[root@localhost bin]# ls
appletviewer  java          | console      | runscript    | pack200      | unpack200
apt           | javac        | control      | sashbugd     | policytool   | wgen
ControlPanel | javadoc     | db           | stack        | rmic         | winprint
extcheck     | javafpackager | fat         | stat         | rmid        | xjc
hola.class    | javah       | info        | statd        | rmiregistry  |
hola.java    | javap       | jsp         | visualview   | schwigen     |
idl          | java-rmi.cgi | jsc         | keytool     | serialver    |
jar          | jarman      | jsc.ini     | native2ascii | servertool   |
jarsigner   | jcmd        | jps         | orbd        | tnameserv
[root@localhost bin]#
```

Fig22. Ejecución del código fuente (Archivo `hola.java`) y generación del código objeto (`hola.class`)

CONCLUSIÓN

La práctica fue un poco estresante y con reto bastante alto ya que la forma de insertar los comandos debe ser tal cual pues el sistema es sensible en la sintaxis, sin embargo si tenemos las herramientas necesarias, ponemos la atención necesaria y tenemos la paciencia se puede lograr el objetivo.

De esta práctica aprendí mucho después de haberlo intentado varias veces porque cada vez que lo hacía aprendía algo nuevo.