

## Clase de shell scripts

### Programas del *shell*: *scripts*

- Un *script* es un archivo de texto regular que contiene órdenes del *shell*, llamadas al sistema, llamadas a utilidades e incluso llamadas a programas binarios para posterior ejecución.
- Antes de correr un *script*, es necesario asignarle permisos de ejecución a través de la utilidad **chmod**. Luego para ejecutarlo sólo es necesario escribir su nombre en el *shell*.
- Son útiles para almacenar secuencias de órdenes comunmente usadas. Ahorran el tener que escribir las órdenes por separado.
- Soportan un ambiente sencillo de programación que permite incorporar estructuras de control básicas (for, while, if), definición de variables, manejo de parámetros que le añaden poder a los *scripts* al poder realizar una amplia variedad de tareas.
- Los administradores de sistemas encuentran particular utilidad en los *scripts* al poder automatizar tareas repetitivas de administración tales como avisar a los usuarios cuando el espacio en disco usado está cercano al límite permitido.
- Cuando un *script* se ejecuta, el sistema determina bajo cual *shell* fue escrito y lo ejecuta usando el *script* como su entrada estándar. El sistema descubre cual *shell* fue usado examinando la primera línea del *script*:
  - Si la primera línea tiene sólo un signo #, significa que el *script* se interpretará con el *shell* desde el cual se ejecutó.
  - Si la primera línea tiene la forma **#!pathName**, el archivo ejecutable **pathName** es usado para interpretar el *script*. (ejemplo: **#!/bin/bash**, interpretará en *script* usando el shell bash).
  - Si no hay ninguna especificación en la primera línea, el *script* se interpretará usando Bourne shell.

El símbolo # en cualquier otro lugar del script indica comentario.

- Es recomendable usar la forma **#!pathName** para evitar ambigüedades.

- Los nombres de los scripts pueden ser absolutamente cualquier cosa y no requieren ninguna extensión. Sin embargo es recomendable colocar una extensión que describa el shell que se usó (por ejemplo: myscript.csh, myscript.ksh o myscript.bash)

Algunos ejemplos sencillos de uso de shell scripts:

- Para matar procesos al hacer log out. Muchas veces, incluso después de cerrar una cuenta, quedan procesos ejecutándose (threads de netscape, por ejemplo) o procesos zombies que mantienen recursos de manera innecesaria.

```
salida_rapida:
#!/bin/bash
#Uso: salida_rapida : mata todos los procesos del
#                               usuario 'user_name'
pkill -u $USER

#fin del script

matar_procesos:

#!/bin/bash
#Uso 1: matar_procesos user_name process: mata los procesos
# 'process' del usuario 'user_name'
#Ejemplo matar_procesos yudith java
#Uso 2: matar_procesos user_name : mata todos los procesos del
#usuario 'user_name'

if [ $2 ]
then
ps -u $1 | grep $2|awk '{print $1}'|xargs kill -9
else
ps -u $1|awk '{print $1}'|xargs kill -9
fi
#fin del script
```

Nota; Con este ejemplo también se puede explicar el pase de parámetros.

- Para revisar el tamaño de todos los directorios

```
du_extendido :
#!/bin/bash
# USO: du_extendido
for i in *;do du -sh $i;done
#fin del script
```

Nota: Con este ejemplo también se explica lo que hace du

- Para borrar todos los archivos temporales que ocupan espacio innecesario.

```
borrar:
#!/bin/bash
find . -name *~ | xargs rm
#fin del script
```

Aun cuando este script parece muy sencillo (sólo una línea que la puedo ejecutar siempre), imaginen el desastre que sería si alguna vez olvido colocar el ~ , así al tenerlo en un script, no hay cabida para esa terrible posibilidad.

- Para definir variables de ambientes frecuentemente usadas.

```
mis_variables:
```

```
#!/bin/bash
JDK=/usr/local/jdk1.2
```

```
# Instalacion local de java
JAVA1_2=/usr/local/jdk1.2
```

```
# Instalacion de la libreria de comunicacion
MPIJAVAHOME=/net/y/ayaly/ldc/yudith/mpiJava-lam-1.0
```

```
#Definicion del classpath
```

```
CLASSPATH=$MPIJAVAHOME:/usr/local/plapackJava-lam-1.0:$IAIK_HOME/lib/iaik_jce
```

```
#Incorporacion del . al PATH

PATH=$PATH:.

export JDK JAVA1_2 MPIJAVAHOME PATH CLASSPATH

#fin del script
```

Este tipo de scripts se pueden incorporar en el .vars para que se ejecute automáticamente cada vez que se entre a la cuenta. Para este ejemplo, sólo es necesario agregar al final del .vars la siguiente línea:

```
. /home/yudith/mis_variables
```