

GUIÓN:

Título

AEV6 - Shell Scripts

Objetivos

- Conocer los fundamentos sintácticos de Shell Scripts.
- Utilizar las utilidades de entrada/salida en el desarrollo de Shell Scripts.
- Utilizar estructuras de control en el desarrollo de Shell Scripts
- Desarrollar Shell Scripts para la automatización de tareas
- Desarrollar Shell Scripts para procesar información contenida en ficheros

Temporalización

Se estima una dedicación de **5 horas**, teniendo en cuenta que habrá que revisar los recursos publicados en Florida Oberta para poder realizar la actividad.

Proceso de desarrollo

1. Se va a proponer un ejercicio en el que habrá que cumplir varios objetivos o requerimientos.
2. Todos los requerimientos propuestos deben quedar resueltos en un único fichero de tipo Shell Script.
3. Entregar el fichero de tipo Shell Script, debidamente identificado, y con los comentarios necesarios para que se entienda su funcionamiento. Si se utilizan recursos o herramientas no vistas en el temario, deben quedar debidamente explicadas en los comentarios.

Evaluación

La resolución de todos los requerimientos del ejercicio sumará 10 puntos. Cada error, imprecisión o incoherencia irá restando puntuación. El Script se testeará en plataforma Linux y debe funcionar en dicha plataforma:

- Errores graves: restan entre un 20% y un 100% (si la respuesta no corresponde a lo solicitado, el ejercicio puede llegar a sumar 0 puntos.)
- Errores leves:
 - Ejercicio parcialmente incorrecto, resta entre 10% - 50%.
 - Cada fallo leve resta entre 5% - 20%.

Recursos

Puestos a disposición del alumno en el curso correspondiente del campus virtual Florida Oberta.

Detalle de la actividad

1. Escribir un Shell Script que, a partir de un fichero que contiene grupos y usuarios de un sistema, lo analice y muestre por pantalla cuántos grupos hay en el fichero, cuántos usuarios tiene cada grupo y cuál es el grupo que tiene más usuarios. Para ello, se penalizará el uso del comando “wc” (-2.0 puntos).

Este fichero (puedes encontrar un ejemplo en el apartado de recursos de Florida Oberta, llamado “grupos.txt”) tiene varias líneas o registros, cada línea representa un grupo. El primer valor de cada línea se corresponde con el nombre del grupo, y a partir de los dos puntos, el resto de los valores, separados por espacios, se corresponden con los usuarios que pertenecen a ese grupo.

La ruta al fichero debe pedirse al usuario como parámetro de entrada. Mientras el usuario introduzca una ruta al fichero que no exista, debe mostrar un mensaje de error y volver a pedir al usuario que introduzca la ruta al fichero.

El Script debe funcionar bien con cualquier fichero que introduzca el usuario y que tenga el formato indicado.

Si hay más de un grupo con el mismo número máximo de usuarios, se debe mostrar el nombre de todos ellos.

A continuación, se muestra el contenido del fichero proporcionado:

administradores: yomismo marta

proyecto: marta ana julia sergio andrea jorge ramon yomismo

interfaz: ana julia marta

bbdd: sergio andrea marta

funcionalidad: jorge ramon marta

costes: ana sergio julia andrea marta

tod@s: marta ana julia sergio andrea jorge ramon yomismo

A continuación, se muestra el resultado de la ejecución del Shell Script:

Introduce el nombre del fichero de grupos a procesar: grupos.txt

El fichero proporcionado tiene 7 grupos

administradores: 2 usuarios

proyecto: 8 usuarios

interfaz: 3 usuarios

bbdd: 3 usuarios

funcionalidad: 3 usuarios

costes: 5 usuarios

tod@s: 8 usuarios

Grupo/s con mayor número de usuarios: proyecto tod@s

```
#!/bin/bash
```

```
#Pedimos el fichero al usuario
```

```
read -p "Introduce el nombre del fichero de grupos a procesar: " fichero
```

```
#Comprobamos que el fichero existe, si no lo volvemos a pedir
```

```
while [ ! -f $fichero ]; do
```

```
    read -p "Error. Introduce el nombre del fichero de grupos a procesar: " fichero
```

```
done
```

```
#Cada grupo equivale a una línea del fichero. Contamos los grupos
```

```
cuentaGrupos=0
```

```
for i in `cat $fichero | awk '{print $1}'`; do
```

```
    cuentaGrupos=$((cuentaGrupos+1))
```

```
done
```

```
echo "El fichero proporcionado tiene $cuentaGrupos grupos"
```

```
#Declaramos las variables para guardar el grupo con más usuarios
grupoMax=""
usuariosMax=0

#Recorremos las líneas
cuentaLineas=1

while [ $cuentaLineas -le $cuentaGrupos ]; do

    grupo=`cat $fichero | head -$cuentaLineas | tail -1 | awk '{print $1}'`

    #Contamos los usuarios por cada grupo
    num=2
    usuario=`cat $fichero | head -$cuentaLineas | tail -1 | awk '{print $${num}}'`
    cuentaUsuarios=0

    while [ "$usuario" != "" ]; do
        cuentaUsuarios=$((cuentaUsuarios+1))
        num=$((num+1))
        usuario=`cat $fichero | head -$cuentaLineas | tail -1 | awk '{print $${num}}'`
    done

    #Guardamos en un fichero cada grupo con el número de usuarios
    echo "$grupo $cuentaUsuarios usuarios" >> usuariosPorGrupo.txt

    #Guardamos el máximo número de usuarios de todos los grupos
    if [ $cuentaUsuarios -ge $usuariosMax ]; then
        usuariosMax=$cuentaUsuarios
    fi

    #Siguiente línea o grupo
    cuentaLineas=`expr $cuentaLineas + 1`

done
```

#Recorremos usuariosPorGrupo.txt para buscar todos los grupos con mayor número de usuarios

cuentaUsuarios=0

cuentaLineas=1

echo -n "Grupo/s con mayor número de usuarios:" >> gruposConMasUsuarios.txt

while [\$cuentaLineas -le \$cuentaGrupos]; do

 cuentaUsuarios=`cat usuariosPorGrupo.txt | head -\$cuentaLineas | tail -1 | awk '{print \$2}'`

 if [\$cuentaUsuarios -eq \$usuariosMax]; then

 grupo=`cat usuariosPorGrupo.txt | head -\$cuentaLineas | tail -1 | awk '{print \$1}'`

 #Le quitamos el carácter ":" al grupo. Podemos hacerlo de varias maneras...

 #grupoMax=`echo \$grupo | sed 's/.\$//'`

 grupoMax=`echo \$grupo | awk -F ":" '{print \$1}'`

 echo -n " \$grupoMax" >> gruposConMasUsuarios.txt

 fi

 cuentaLineas=`expr \$cuentaLineas + 1`

done

#Para incluir un salto de línea

echo " " >> gruposConMasUsuarios.txt

#Mostramos resultados

cat usuariosPorGrupo.txt

cat gruposConMasUsuarios.txt

#Borramos los ficheros temporales

if [-f usuariosPorGrupo.txt]; then

 rm usuariosPorGrupo.txt

fi

if [-f gruposConMasUsuarios.txt];then

 rm gruposConMasUsuarios.txt

fi