

## Ejercicios de Shell Script

### Relación – 2 SOLUCIONES

1. Crea un programa que muestre un menú para realizar las siguientes operaciones:

- Crear un grupo
- Eliminar un grupo.
- Crear un usuario.
- Eliminar un usuario.
- Salir

```
while [ 1 -eq 1 ]
do
    echo 1 – crear grupo
    echo 2 – eliminar grupo
    echo 3 – crear usuario
    echo 4 – eliminar usuario
    echo 5 – salir
    echo introduzca una opcion
    read op

    if [ $op -eq 1 ]
    then
        echo introduzca el grupo
        read gr
        addgroup $gr
    fi

    if [ $op -eq 2 ]
    then
        echo introduzca el grupo
        read gr
        delgroup $gr
    fi

    if [ $op -eq 3 ]
    then
        echo introduzca el usuario
        read usu
        adduser $usu
    fi

    if [ $op -eq 4 ]
    then
        echo introduzca el usuario
        read usu
        deluser $usu
    fi

    if [ $op -eq 5 ]
    then
        exit 0
    fi
done
```

2.Construye un programa que reciba una serie de numeros y muestre el menor

```
menor=$1
for k in $*
do
    if [ $k -lt $menor ]
    then
        menor=$k
    fi
done
echo $menor
```

3.Construye un programa que reciba como parámetros el nombre de un directorio y un tamaño en bytes. El programa debe mostrar un listado de todas las entradas de ese directorio, cuyo tamaño sea inferior al indicado por parámetro.

```
for k in `ls -l $1 | tr -s ' ' |`
do
    nom=`echo $k | cut -d ' ' -f8`
    tam=`echo $k | cut -d ' ' -f5`

    if [ $tam -lt $2 ]
    then
        echo $nom
    fi
done
```

4.Crea un programa que acepte como parámetros una serie de nombres de ficheros y como último parámetro, el nombre de un directorio. El programa debe mostrar un mensaje indicando cuantos de los ficheros que recibió por parámetro, se encuentran en el directorio indicado.

```
tot=0
# Obtengo el directorio en la variable dir
for dir in $*
do
    s=0
done

for k in $*
do
    if [ -a ${dir}/${k} ]
    then
        tot=`expr $tot + 1`
    fi
done

echo El número de archivos es $tot
```

5.Realizar un script que reciba como parámetros una extensión de archivo (sin punto) y el nombre completo de un directorio. El programa preguntará al usuario si desea copiar o mover los archivos con esa extensión, del directorio actual al especificado como parámetro.

```

echo 1 – copiar
echo 2 – mover
read op
for k in `ls | grep .$1`
do
    if [ $op -eq 1 ]
    then
        cp $k $2
    fi

    if [ $op -eq 2 ]
    then
        mv $k $2
    fi
done

```

6.Hacer un script que busque la presencia del fichero pasado como argumento en alguno de los directorios referenciados en la variable \$PATH, señalando su localización exacta. Nota: No usar el comando whereis.

```

for dir in `echo $PATH | tr -s ':' ' '`
do
    if [ -a ${dir}/${1} ]
    then
        echo el fichero $1 está en $dir
    fi
done

```

7.Crea un programa que reciba por pantalla el nombre de un usuario muestre por pantalla el NOMBRE de su grupo primario.

```

# Leo el identificador del grupo primario
for linea in `cat /etc/passwd`
do
    nom=`echo $linea | cut -d ':' -f1`
    if [ $nom = $1 ]
    then
        gr=`echo $linea | cut -d ':' -f4`
    fi
done

if [ -z $gr ]
then
    echo el usuario $1 no existe
    exit 0
fi

# Leo el nombre del grupo
for linea in `cat /etc/group`
do
    id=`echo $linea | cut -d ':' -f3`
    if [ $id = $gr ]
    then
        nom=`echo $linea | cut -d ':' -f1`
    fi
done

```

```

    echo El grupo primario de $1 es $nom
  fi
done

```

8.Crea un programa que reciba por parámetro el nombre de un usuario y MATE a todos sus procesos.

```

for p in `ps -u $1 | tr -s ' ' | cut -d ' ' -f1`
do
    kill -9 $p
done

```

9.Crea un programa que mire cada 30 segundos los procesos existentes en memoria, y MATE a aquellos procesos cuyo nombre esté en una lista almacenada en el fichero /root/procesos\_capados.

```

while [ 1 -eq 1 ]
do
    for p in `ps -ef | tr -s ' ' | cut -d ' ' -f2,8`
    do
        pid=`echo $p | cut -d ' ' -f1`
        nom=`echo $p | cut -d ' ' -f2`

        b=`cat /root/procesos_capados | grep $nom`
        if ! [ -z $b ]
        then
            kill -9 $pid
        fi
    done

    sleep 30
done

```

10.Crea un programa capaz de gestionar el fichero del ejercicio 9. Las opciones que debe contener son:

- Listar procesos prohibidos.
- Añadir proceso.
- Eliminar proceso.

```

while [ 1 -eq 1 ]
do

    echo 1 – Listar procesos
    echo 2 – Añadir proceso
    echo 3 – Eliminar proceso
    echo 4 – salir
    read op

    if [ $op -eq 1 ]
    then
        cat /root/procesos_capados
    fi

    if [ $op -eq 2 ]

```

```
then
    read proceso
    echo $proceso >> /root/procesos_capados
fi

if [ $Sop -eq 3 ]
then
    read proceso
    for k in `cat /root/procesos_capados`
    do
        if [ $k != $proceso ]
        then
            echo $k >> /root/procesos_capados
        fi
    done
    mv aux /root/procesos_capados
fi

if [ $Sop -eq 4 ]
then
    exit 0
fi
done
```