

## Ejercicios de Shell Script. Repaso 1.

1. Crea un programa que reciba por parámetro un nombre de proceso y mate a todos los procesos que se llamen igual.

```
for linea in `ps -ef | tr -s ' ' '#' | cut -d '#' -f 2,8`
do
  nom=`echo $linea | cut -d '#' -f2`
  pid=`echo $linea | cut -d '#' -f1`

  if [ $nom = $1 ]
  then
    kill -9 $pid
  fi
done
```

2. Crea un programa que reciba por parámetro un nombre de usuario y muestre por pantalla la suma del tamaño de todos los ficheros que tiene en el sistema.

```
tot=0
for f in `find /home/$1 -user $1`
do
  if [ ! -d "$f" ] && [ -a "$f" ]
  then
    tam=`ls -l "$f" | tr -s ' ' | cut -d ' ' -f5`

    tot=`expr $tot + $tam`
  fi
done

echo $1 - $tot
```

3. Crea un programa que reciba por pantalla un nombre de directorio y calcule el porcentaje de espacio que ocupa cada uno de los ficheros que contiene, en relación con el tamaño total del directorio. Ejemplo.- Si el directorio es /home/alumno/D1 y este tiene tres ficheros (f1,f2 y f3), la salida podría ser:

```
f1 ---- 5%
f2 ---- 20%
f3 ---- 75%
```

NOTA: El programa no debe tener en cuenta los subdirectorios.

```
tot=0
for t in `ls -l $1 | tr -s ' ' | cut -d ' ' -f5`
do
  tot=`expr $tot + $t`
done

for linea in `ls -l $1 | tr -s ' '@' | cut -d '@' -f 5,9`
do
  tam=`echo $linea | cut -d '@' -f1`
  nom=`echo $linea | cut -d '@' -f2`

  por=`expr $tam \* 100`
  por=`expr $por / $tot`

  echo $nom --- $por %'
done
```

4. Crea un script que guarde en el fichero /root/procesos, el nombre, PID y usuario propietario de todos los procesos que NO SEAN del usuario root y lleven más de 10 segundos de tiempo de ejecución.

```
for linea in `ps -ef | tr -s ' ' '@' | cut -d '@' -f1,2,7,8`
do
    usu=`echo $linea | cut -d '@' -f1`
    pid=`echo $linea | cut -d '@' -f2`
    t=`echo $linea | cut -d '@' -f3`
    p=`echo $linea | cut -d '@' -f4`

    if [ $usu != root ]
    then
        h=`echo $t | cut -d ':' -f1`
        m=`echo $t | cut -d ':' -f2`
        s=`echo $t | cut -d ':' -f3`
        if [ $h -gt 0 ] || [ $m -gt 0 ] || [ $s -gt 10 ]
        then
            echo $pid $usu $p
        fi
    fi
done
```

5. Haz un programa que reciba por parámetro una serie de nombres de fichero y muestre el nombre del más antiguo.

```
echo > aux
for linea in `ls -l --full-time | tr -s ' ' '|' | cut -d '|' -f7,10`
do
    echo $linea >> aux
done

cat aux | sort | tail -n1
```

6. Crea un programa que pida por pantalla el nombre de dos grupos y pase a todos los usuarios del primer grupo al segundo.

```
echo Introduzca el grupo origen
read g1
echo Introduzca el grupo destino
read g2

# Modificar los grupos primarios
gid_origen=`cat /etc/group | cut -d ':' -f1,3 | grep -w $g1 | cut -d ':' -f2`
gid_destino=`cat /etc/group | cut -d ':' -f1,3 | grep -w $g2 | cut -d ':' -f2`

#echo $g1 - $gid_g1
#echo $g2 - $gid_g2
```

```

# Modifico /etc/passwd
echo > aux

for linea in `cat /etc/passwd | tr -s ' ' '\n`
do
    gid=`echo $linea | cut -d ':' -f4`

    if [ $gid != $gid_origen ]
    then
        echo $linea | tr -s '\n' ' ' >> aux
    else
        p=`echo $linea | cut -d ':' -f1,2,3`
        f=`echo $linea | cut -d ':' -f5,6,7`

        echo $p:$gid_destino:$f | tr -s '\n' ' ' >> aux
    fi
done

# Obtengo los usuarios del grupo origen en /etc/group
for linea in `cat /etc/group`
do
    gr=`echo $linea | cut -d ':' -f1`

    if [ $gr = $g1 ]
    then
        usu=`echo $linea | cut -d ':' -f4`
    fi
done

# Modifico el fichero
echo > aux1
for linea in `cat /etc/group`
do
    gr=`echo $linea | cut -d ':' -f1`

    if [ $gr = g1 ]
    then
        p=`echo $linea | cut -d ':' -f1,2,3`
        echo $p: >> aux1
    fi

    if [ $gr = g2 ]
    then
        p=`echo $linea | cut -d ':' -f1,2,3`
        f=`echo $linea | cut -d ':' -f4`

        if [ -z "$f" ]
        then
            echo $p:$usu >> aux1
        else
            echo $p:$f,$usu >> aux1
        fi
    fi

    if [ $gr != g1 ] && [ $gr != $g2 ]
    then
        echo $linea >> aux1
    fi
done

```

7. Crea un programa que funcione como un traductor entre español e inglés. Para ello, el programa mantendrá un fichero de diccionario, en el cual se almacenará cada palabra en español y su traducción al inglés. Las operaciones que debe permitir este programa son:

- Introducir palabra en el diccionario.
- Traducir palabra del español al inglés.
- Traducir palabra del inglés al español.
- Borrar palabra del diccionario.
- Corregir una palabra del diccionario.

```
while [ 1 -eq 1 ]
do

    echo DICCIONARIO
    echo
    echo 1 - Introducir palabra
    echo 2 - Traducir Ingles - Español
    echo 3 - Traducir Español - Ingles
    echo 4 - Borrar palabra
    echo 5 - Corregir palabra
    echo 6 - salir

    echo
    echo introduzca una opcion
    read op

    # Añadir palabra
    if [ $op -eq 1 ]
    then
        echo Palabra en Ingles
        read i
        echo Palabra en Español
        read e

        echo ${e}-${i} >> dic
    fi

    # Traduccion de Ingles a español
    if [ $op -eq 2 ]
    then
        echo Palabra en Ingles
        read it

        esta=0
        for linea in `cat dic`
        do
            i=`echo $linea | cut -d '-' -f2`
            e=`echo $linea | cut -d '-' -f1`

            if [ $i = $it ]
            then
                echo $e
                esta=1
            fi
        done

        if [ $esta -eq 0 ]
        then
            echo $it no esta en el diccionario
        fi
    fi
```



```

# Traducción del español al inglés
if [ $op -eq 3 ]
then
    echo Palabra en Español
    read es

    esta=0
    for linea in `cat dic`
    do
        i=`echo $linea | cut -d '-' -f2`
        e=`echo $linea | cut -d '-' -f1`

        if [ $e = $es ]
        then
            echo $i
            esta=1
        fi
    done

    if [ $esta -eq 0 ]
    then
        echo $es no esta en el diccionario
    fi
fi

# Borrar palabra
if [ $op -eq 4 ]
then
    echo Introduzca la palabra en español
    read es
    for linea in `cat dic`
    do
        e=`echo $linea | cut -d '-' -f1`

        if [ $e != $es ]
        then
            echo $linea >> aux
        fi
    done

    mv aux dic
fi

# Corregir palabra
if [ $op -eq 5 ]
then
    echo Introduzca la palabra en español a corregir
    read es
    echo Introduzca la palabra corregida en español
    read co
    echo Introduzca la traduccion al ingles
    read it

```

```

for linea in `cat dic`
do
    e=`echo $linea | cut -d '-' -f1`

    if [ $e != $es ]
    then
        echo $linea >> aux
    else
        echo ${co}-${it} >> aux
    fi

done

mv aux dic
fi

# Salir
if [ $op -eq 6 ]
then
    exit
fi

done

```

9. Crea un programa que muestre los números primos existentes entre el 1 y el 100.

```

num=1
while [ $num -lt 100 ]
do

    # revisa si un numero es primo
    n=`expr $num - 1`
    primo=1

    while [ $n -gt 1 ]
    do
        resto=`expr $num % $n`

        if [ $resto -eq 0 ]
        then
            primo=0
        fi

        n=`expr $n - 1`
    done

    if [ $primo -eq 1 ]
    then
        echo $num
    fi

    num=`expr $num + 1`
done

```

10. Crea un programa que muestre los sistemas de ficheros con más de un 3% de espacio ocupado.

```
i=0
for linea in `df | tr -s ' ' '@`
do
  if [ $i -gt 0 ]
  then
    nom=`echo $linea | cut -d '@' -f1`
    por=`echo $linea | cut -d '@' -f5 | cut -d '%' -f1`

    if [ $por -gt 3 ]
    then
      echo $nom --- $por
    fi
  fi
  i=1
done
```

11. Crea un programa que permita cambiar el directorio de trabajo de un usuario. NOTA: Este programa debe modificar automáticamente el fichero /etc/passwd.

```
echo Introduzca el nombre de usuario
read usu
```

```
echo Introduzca el nuevo directorio
read dir
```

```
for linea in `cat /etc/passwd`
do
  u=`echo $linea | cut -d ':' -f1`

  if [ $u != $usu ]
  then
    echo $linea >> aux
  else
    ini=`echo $linea | cut -d ':' -f1,2,3,4,5`
    fin=`echo $linea | cut -d ':' -f7`
    echo ${ini}:${dir}:${fin} >> aux
  fi
done
```

```
mv aux /etc/passwd
```

12. Escribe un programa que cuente el número de enlaces simbólicos existentes en la carpeta personal de tu usuario y en sus subcarpetas.

```
tot=0
for f in `find $HOME`
do
  if [ -a $f ] && [ ! -d $f ]
  then
    l=`ls -l $f | cut -c1`

    if [ $l = "l" ]
    then
      tot=`expr $tot + 1`
    fi
  fi
done
```

echo El total de enlaces simbolicos es \$tot

13. Crea un programa que reciba el nombre de un fichero, lo busque en el sistema y muestre la ruta absoluta donde esta dicho archivo. Ej.- Si el fichero fichero f1 esta en /home/alumno/D1, el programa debe mostrar /home/alumno/D1

```
nom=`find $HOME -name $1 2>/dev/null`
```

```
for k in `echo $nom | tr -s '/' ' '`
do
  dir=$aux
  aux=${aux}/${k}
done
```

echo el fichero \$1 esta en \$dir

14. Crea un programa que permita realizar las siguientes operaciones:

- Mostrar el factorial de un número.
- Elevar un número a otro.
- Decir si un número es par o no.
- Calcular la media aritmética de una serie de números.

```
while [ 1 -eq 1 ]
do
  echo 1 - factorial
  echo 2 - elevado
  echo 3 - par impar
  echo 4 - media aritmetica
  echo 5 - salir

  echo Introduzca una opcion
  read op

  if [ $op -eq 1 ]
  then
    echo numero
    read num

    i=$num
    res=1
```



```

while [ $i -gt 1 ]
do
    res=${res*$i}
    i=$((i-1))
done

echo $num factorial es $res
fi

if [ $op -eq 2 ]
then
    echo base
    read base
    echo exponente
    read exp

    i=1
    res=1
    while [ $i -le $exp ]
    do
        res=$((res*base))
        i=$((i+1))
    done

    echo $base elevado a $exp es $res
fi

if [ $op -eq 3 ]
then
    echo numero
    read num

    res=$((num % 2))
    if [ $res -eq 0 ]
    then
        echo $num es par
    else
        echo $num es impar
    fi
fi

if [ $op -eq 4 ]
then
    suma=0
    n=1
    num=0

    while [ $n -gt 0 ]
    do
        echo 'introduzca numero (0-salir)'
        read n
    done

```

```
    suma=`expr $suma + $n`  
    num=`expr $num + 1`  
done  
med=`expr $suma / $num`  
echo La media aritmetica de $lista es $med  
fi  
  
if [ $op -eq 5 ]  
then  
    exit  
fi  
done
```