

Ejercicios de Shell Script

Relación – 1

SOLUCIONES

1. Crea un programa que reciba una ruta absoluta por parámetro y devuelva un mensaje indicando si es un fichero, un directorio o no existe en el sistema.

```
if [ -a $1 ]
then
  if [ -d $1 ]
  then
    echo es un directorio
  else
    echo es un fichero
  fi
else
  echo No existe.
fi
```

2. Crea un programa que reciba un número indeterminado de parámetros y los imprima por pantalla con el siguiente formato:
El parámetro 1 es
El parámetro 2 es

```
i=1
for k in $*
do
  echo El parámetro $i es $k
  i=`expr $i + 1`
done
```

3. Crea un guión shell que reciba una serie de números por parámetro y muestre los menores de 5.

```
for k in $*
do
  if [ $k -lt 5 ]
  then
    echo $k
  fi
done
```

4. Crea un guión shell que reciba un número por parámetro y muestre por la salida estándar todos los números menores que el, hasta el 0. NOTA: Utiliza un bucle while

```
n=$1
while [ $n -lt 0 ]
do
  echo $n
  n=`expr $n - 1`
done
```

5. Crea un guión shell que funcione como el anterior, pero sin usar un bucle while. NOTA:

Utiliza una solución recursiva.

NO SE HACE.

6. Crea un programa que reciba como parámetro el nombre de un usuario del sistema e indique si está conectado o no.

```
c=`who | grep -w $1 2>/dev/null`  
if [ -z $c ]  
then  
    echo $1 no está conectado  
else  
    echo $1 está conectado  
fi
```

7. Crea un guión shell que reciba como parámetro un número e indique si es primo o no.

```
k=`expr $1 - 1`  
while [ k -gt 1 ]  
do  
    resto=`expr $1 % $k`  
    if [ $resto -eq 0 ]  
    then  
        echo El número $1 no es primo  
        exit 0  
    fi  
  
    k=`expr $k - 1`  
done  
  
echo El número $1 no es primo.
```

8. Crea un programa que reciba varios valores por parámetro y los sume.

```
tot=0  
for k in $*  
do  
    tot=`expr $tot + $k`  
done  
  
echo La suma total es $tot
```

9. Crea un guión shell que reciba como parámetro la ruta completa de un directorio y muestre por pantalla la suma del tamaño de todos los ficheros que contiene.

```
tot=0  
for k in `ls -l $1 | tr -s ' ' | cut -d ' ' -f5`  
do  
    tot=`expr $tot + $k`  
done  
  
echo El tamaño de los ficheros de $1 es $tot
```

10. Crea muestra el siguiente menú e implemente sus operaciones:

- Sumar dos números.
- Restar dos números.
- Multiplicar dos números.
- Dividir dos números.
- Salir

```
while [ 1 -eq 1 ]  
do
```

```
    echo 1 – Sumar  
    echo 2 – Restar  
    echo 3 – Multiplicar  
    echo 4 – Dividir  
    echo 5 – Salir
```

```
    echo Introduzca una opción  
    read op
```

```
    # Suma  
    if [ $op -eq 1 ]  
    then  
        echo Introduzca el primer número  
        read n1  
        echo Introduzca el segundo número  
        read n2  
  
        res=`expr $n1 + $n2`  
        echo $n1 + $n2 = $res  
    fi
```

```
    # Resta  
    if [ $op -eq 2 ]  
    then  
        echo Introduzca el primer número  
        read n1  
        echo Introduzca el segundo número  
        read n2  
  
        res=`expr $n1 - $n2`  
        echo $n1 - $n2 = $res  
    fi
```

```
    # Multiplicacion  
    if [ $op -eq 3 ]  
    then  
        echo Introduzca el primer número  
        read n1  
        echo Introduzca el segundo número  
        read n2  
  
        res=`expr $n1 \* $n2`
```

```
    echo $n1 \* $n2 = $res
fi

# División
if [ $Sop -eq 4 ]
then
    echo Introduzca el primer número
    read n1
    echo Introduzca el segundo número
    read n2

    res=`expr $n1 / $n2`
    echo $n1 / $n2 = $res
fi

# Salir
if [ $Sop -eq 5 ]
then
    exit 0
fi
done
```