

Bucles I

Wuke Zhang

2-ASIR

Para poner en práctica el uso de los bucles dentro del Shell vamos a crear una serie de scripts que resuelvan los siguientes enunciados:

1- Un script que lee una cantidad de números (escrito como parámetro en la ejecución) y retorne la suma de todos:

```
$> ./suma_todo.sh 1 3 6 2 8
```

```
$> Es resultado es 20
```

```
root@DESKTOP-DC3KKHF: /home/wuke123
```

```
GNU nano 6.2 suma_todo.sh *
#!/bin/bash

# Verificar si hay parámetros
if [ "$#" -lt 1 ]; then
    echo "Por favor ingresa al menos un número."
    exit 1
fi

# Sumar los números
suma=0
for num in "$@"; do
    suma=$((suma + num))
done

# Mostrar el resultado
echo "El resultado es: $suma"
```

```
root@DESKTOP-DC3KKHF:/home/wuke123# ./suma_todo.sh 1 3 6 2 8
El resultado es: 20
```

2- Un script que escriba un texto tantas veces como se le indique:

```
$> ./repite_mensaje.sh
```

```
$> Escribe el mensaje a repetir:
```

```
$> Hola
```

```
$> ¿Cuántas veces?
```

```
$> 5
$> Hola
$> Hola
$> Hola
$> Hola
$> Hola
```

```
root@DESKTOP-DC3KKHF: /home/wuke123
GNU nano 6.2 repite_mensaje.sh *
#!/bin/bash

# Pedir el mensaje y las veces
read -p "Escribe el mensaje a repetir: " mensaje
read -p "¿Cuántas veces? " veces

# Repetir el mensaje
for ((i = 1; i <= veces; i++)); do
    echo "$mensaje"
done

root@DESKTOP-DC3KKHF:/home/wuke123# ./repite_mensaje.sh
Escribe el mensaje a repetir: hola
¿Cuántas veces? 5
hola
hola
hola
hola
hola
hola
```

3- Un script que ponga todos los números pares hasta el número que se escriba como parámetro:

```
$> ./números_pares.sh 10
$> 2, 4, 6, 8
```

```
root@DESKTOP-DC3KKHF: /home/wuke123
GNU nano 6.2 numeros_pares.sh *
#!/bin/bash

# Verificar si se ha dado un parámetro
if [ -z "$1" ]; then
    echo "Por favor ingresa un número."
    exit 1
fi

# Mostrar los números pares
for ((i = 2; i <= $1; i+=2)); do
    echo -n "$i"
    if [ $i -lt $1 ]; then
        echo -n ", "
    fi
done
echo
```

```

root@DESKTOP-DC3KKHF:/home/wuke123# ./numeros_pares.sh 10
2, 4, 6, 8, 10
root@DESKTOP-DC3KKHF:/home/wuke123#

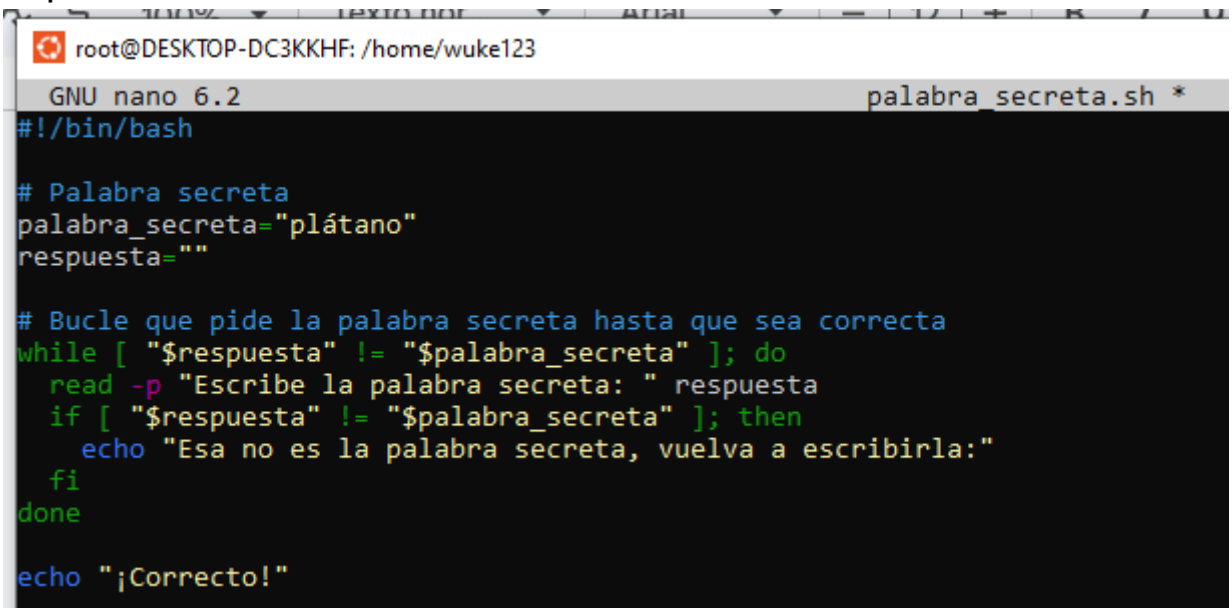
```

4- Un script que pida una palabra determinada (código) para terminar la ejecución y mientras no se le indique, no termine de ejecutarse:

```

$> ./palabra_secreta.sh
$> Escribe la palabra secreta:
$> plátano
$> Esa no es la palabra secreta, vuelva a escribirla:
$> manzano
$> Esa no es la palabra secreta, vuelva a escribirla:
$> mandarina
$> ¡Correcto!

```



```

root@DESKTOP-DC3KKHF:/home/wuke123
GNU nano 6.2 palabra_secreta.sh *
#!/bin/bash

# Palabra secreta
palabra_secreta="plátano"
respuesta=""

# Bucle que pide la palabra secreta hasta que sea correcta
while [ "$respuesta" != "$palabra_secreta" ]; do
    read -p "Escribe la palabra secreta: " respuesta
    if [ "$respuesta" != "$palabra_secreta" ]; then
        echo "Esa no es la palabra secreta, vuelva a escribirla:"
    fi
done

echo "¡Correcto!"

```

```

root@DESKTOP-DC3KKHF:/home/wuke123# ./palabra_secreta.sh
Escribe la palabra secreta: platano
Esa no es la palabra secreta, vuelva a escribirla:
Escribe la palabra secreta: manzano
Esa no es la palabra secreta, vuelva a escribirla:
Escribe la palabra secreta: mandarina
Esa no es la palabra secreta, vuelva a escribirla:
Escribe la palabra secreta:
Esa no es la palabra secreta, vuelva a escribirla:
Escribe la palabra secreta: plátano
¡Correcto!
root@DESKTOP-DC3KKHF:/home/wuke123#

```



5- Un script a partir del anterior, pero contando la cantidad de intentos fallidos. Si supera la cantidad, que serán 3, deberá terminar la ejecución indicando el siguiente mensaje:

```

$> Excedido los intentos permitidos.

```

Además, por cada error cometido deberá indicarse cuantos intentos quedan:

\$> Esa no es la palabra secreta, quedan 2 intentos, vuelva a escribirla:

```
root@DESKTOP-DC3KKHF: /home/wuke123
GNU nano 6.2 palabra_secreta2.sh *
#!/bin/bash

# Palabra secreta
palabra_secreta="plátano"
respuesta=""
intentos=3

# Bucle con conteo de intentos
while [ "$respuesta" != "$palabra_secreta" ]; do
    read -p "Escribe la palabra secreta: " respuesta
    if [ "$respuesta" != "$palabra_secreta" ]; then
        ((intentos--))
        if [ $intentos -gt 0 ]; then
            echo "Esa no es la palabra secreta, quedan $intentos intentos, vuelva a escribirla:"
        else
            echo "Excedido los intentos permitidos."
            exit 1
        fi
    fi
done

echo "¡Correcto!"

root@DESKTOP-DC3KKHF:/home/wuke123# ./palabra_secreta2.sh
Escribe la palabra secreta: pepe
Esa no es la palabra secreta, quedan 2 intentos, vuelva a escribirla:
Escribe la palabra secreta: juan
Esa no es la palabra secreta, quedan 1 intentos, vuelva a escribirla:
Escribe la palabra secreta: Anton
Excedido los intentos permitidos.
root@DESKTOP-DC3KKHF:/home/wuke123#
```



La entrega de esta tarea se realizará vía Moodle y mediante un documento en pdf con sus respectivas capturas de pantalla mostrando su perfecto funcionamiento y con el siguiente formato de nombre:

Actividad13_Nombre_Apellido1_Apellido2.pdf

Ejemplo:

Actividad13_Hadriel_Fernando_Quintero_Ceballos.pdf