



UNIVERSIDAD DEL BÍO-BÍO

DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN

Sistemas Operativos

Laboratorio 2

Programación Shell Script Linux

Objetivos

1. Introducir al alumno en el uso del lenguaje de programación de Shell-scripts con el intérprete de comandos BASH.
2. Manejar la sintaxis de Shell propuesta por el estándar POSIX IEEE 1003.2
3. Profundizar analizando las funcionalidades adicionales añadidas por bash.

I. Shell Scripts.

Crear un directorio llamado “scripts” dentro de nuestro directorio /home/tu_nombre_usuario, entrar al directorio “scripts”. A partir de aquí crearemos nuestros scripts para trabajar.

1. Ejercicio 1. Primer Script (25 pts)

Crear un script llamado **ejercicio1.sh** que al ejecutarlo nos muestre por pantalla nuestro nombre y apellidos, para esto utilizaremos 2 variables llamadas nombre y apellidos. En adelante para ejecutar cada script lo convertiremos en ejecutable con el siguiente comando, sin comillas: “**chmod +x ejercicio1.sh**” y lo ejecutamos con “**./ejercicio1.sh**”.

```
GNU nano 5.4
1  #!/bin/bash
2  nombre="Dylan"
3  apellidos="Suazo Escobar"
4
5  echo "Tu nombre es: $nombre $apellidos"
6
```

```
root@31ce79996126:/home/dsuazo/scripts# ./ejercicio1.sh
Tu nombre es: Dylan Suazo Escobar
root@31ce79996126:/home/dsuazo/scripts#
```

2. Ejercicio 2. Parámetros (20 pts)

Crear un script que permita recibir 3 parámetros numéricos, sumarlos y mostrar por pantalla el resultado de la suma.

```
GNU nano 5.4                                ejercicio2.sh
1  #!/bin/bash
2  if [ $# -ne 3 ]; then
3      echo "Error: debe ingresar 3 numeros"
4      exit 1
5  fi
6
7  p1=$1
8  p2=$2
9  p3=$3
10 resultado=$((p1 + p2 + p3))
11
12 echo "Los dos primeros parametros que ingresaste fueron: $p1 y $p2"
13 echo "La suma de todos los parametros ingresados: $p1 $p2 $p3 es $resultado"
14
```

```
root@3lce79996126:/home/dsuazo/scripts# ./ejercicio2.sh 5 3 15
Los dos primeros parametros que ingresaste fueron: 5 y 3
La suma de todos los parametros ingresados: 5 3 15 es 23
root@3lce79996126:/home/dsuazo/scripts#
```

3. Ejercicio 3. Operaciones con datos (20 pts)

Crear un script que pida un numero por pantalla y que determine si el número ingresado es par o impar. Linux utiliza el operando % para determinar el resto de una división, por lo tanto, es posible utilizarlo para determinar si es par o impar al dividir por 2, si entrega un resto 0 es par, de lo contrario es impar.

```
GNU nano 5.4
1  #!/bin/bash
2  echo "Introduzca un numero: "
3  read n
4  if [ $((n % 2)) -eq 0 ]; then
5      echo "El numero: $n es par"
6  else
7      echo "El numero: $n es impar"
8  fi
9
```

```

root@3lce79996126:/home/dsuazo/scripts# ./ejercicio3.sh
Introduzca un numero:
2
El numero: 2 es par
root@3lce79996126:/home/dsuazo/scripts# ./ejercicio3.sh
Introduzca un numero:
5
El numero: 5 es impar
root@3lce79996126:/home/dsuazo/scripts# █

```

4. Ejercicio 4. Operaciones Aritméticas (20 pts)

Crear un script que pida por pantalla las 3 variables numéricas del ejercicio 2, las sume, las reste, las multiplique y las divida. Mostrar el resultado de cada una y que nos diga si el número es par o impar.

```

GNU nano 5.4                                ejercicio4.sh
1  #!/bin/bash
2  echo "Introduzca un numero: "
3  read n1
4  echo "Introduzca un numero: "
5  read n2
6  echo "Introduzca un numero: "
7  read n3
8
9  suma=$((n1 + n2 + n3))
10 resta=$((n1 - n2 - n3))
11 mul=$((n1 * n2 * n3))
12 div=$((n1 / n2 / n3))
13 restol=$((n1 % n2))
14 resto2=$((restol % n3))
15
16 par_impar(){
17     if [ $(( $1 % 2 )) -eq 0 ]; then
18         echo "par"
19     else
20         echo "impar"
21     fi
22 }
23
24 num1=$(par_impar $suma)
25 num2=$(par_impar $resta)
26 num3=$(par_impar $mul)
27 num4=$(par_impar $div)
28
29 echo "La suma de los numeros: $n1 $n2 $n3 es $suma"
30 echo "La resta de los numeros: $n1 $n2 $n3 es $resta"
31 echo "La multiplicacion de los numeros: $n1 $n2 $n3 es $mul"
32 echo "La division de los numeros: $n1 $n2 $n3 es $div y sobran $resto2"
33 echo "El numero $suma es $num1"
34 echo "El numero $resta es $num2"
35 echo "El numero $mul es $num3"
36 echo "El numero $div es $num4"
37

```

```

root@31ce79996126:/home/dsuazo/scripts# ./ejercicio4.sh
Introduzca un numero:
10
Introduzca un numero:
3
Introduzca un numero:
2
La suma de los numeros: 10 3 2 es 15
La resta de los numeros: 10 3 2 es 5
La multiplicacion de los numeros: 10 3 2 es 60
La division de los numeros: 10 3 2 es 1 y sobran 1
El numero 15 es impar
El numero 5 es impar
El numero 60 es par
El numero 1 es impar
root@31ce79996126:/home/dsuazo/scripts#

```

5. Ejercicio 5. Bucles (20 pts)

Crear un script que adivine el PID (identificador de proceso), que nos responda en cada interacción si el numero ingresado es mayor o menor al PID. Cuando se acierte dicho número, terminará el script y mostrará cuantos intentos se han necesitado.

```

GNU nano 5.4                                ejercicio5.sh
1  #!/bin/bash
2  pid=$$
3  n=0
4
5  pedir_num(){
6      read -p "Ingresa un numero: " num
7      n=$((n+1))
8  }
9
10 pedir_num
11
12 while true; do
13     if [ $num -eq $pid ]; then
14         echo "Haz necesitado $n intentos para adivinar el PID $pid"
15         break
16     elif [ $num -lt $pid ]; then
17         echo "$num es MENOR que PID"
18         pedir_num
19     else
20         echo "$num es MAYOR que PID"
21         pedir_num
22     fi
23 done
24

```

```
root@3lce79996126:/home/dsuazo/scripts# ./ejercicio5.sh
Ingrese un numero: 2900
2900 es MAYOR que PID
Ingrese un numero: 2830
2830 es MENOR que PID
Ingrese un numero: 2832
2832 es MENOR que PID
Ingrese un numero: 2842
2842 es MAYOR que PID
Ingrese un numero: 2833
2833 es MENOR que PID
Ingrese un numero: 2838
2838 es MAYOR que PID
Ingrese un numero: 2825
2825 es MENOR que PID
Ingrese un numero: 2835
2835 es MAYOR que PID
Ingrese un numero: 2834
Haz necesitado 9 intentos para adivinar el PID 2834
root@3lce79996126:/home/dsuazo/scripts#
```

II. Entrega (10 pts)

Documente (screenshots) cada una de las acciones antes señaladas. Es de exclusiva responsabilidad del estudiante respetar el formato de entrega de informe de esta guía **(debajo de cada enunciado su screenshot en donde aparezca de forma clara las sentencias utilizadas)**. El formato de entrega debe ser en PDF. Todas las actividades deben ser entregadas (subidas) a la plataforma digital en las fechas establecidas, por cada hora de atraso, se descontará 1 pto.