UNIVERSIDAD TECNOLÓGICA NACIONAL

TRABAJO PRÁCTICO: 1° ENTREGA -BASH

ASIGNATURA: ARQUITECTURA Y SISTEMAS OPERATIVOS

PROFESOR: BRUSELARIO, SEBASTIÁN
ALUMNO: BARTHELEMY, TOMÁS
TURNO: TURNO TARDE
AÑO: 2024

TABLA DE CONTENIDOS

1. INTRODUCCIÓN..... 3
2. RESOLUCIÓN DEL EJERCICIO 1.....4
3. RESOLUCIÓN DEL EJERCICIO 2.....4
4. RESOLUCIÓN DEL EJERCICIO 3.....5
5. RESOLUCIÓN DEL EJERCICIO 4.....7
6. RESOLUCIÓN DEL EJERCICIO 5.....7
7. RESOLUCIÓN DEL EJERCICIO 6.....8
8. RESOLUCIÓN DEL EJERCICIO 7.....9
9. RESOLUCIÓN DEL EJERCICIO 8.....10
10. RESOLUCIÓN DEL EJERCICIO 9.....11
11. RESOLUCIÓN DEL EJERCICIO 10.....12
12. RESOLUCIÓN DEL EJERCICIO 11.....12
13. RESOLUCIÓN DEL EJERCICIO 12.....13
14. RESOLUCIÓN DEL EJERCICIO 13.....14

1. INTRODUCCIÓN

El Trabajo Práctico consiste en una serie de 13 ejercicios los cuales fueron desarrollados en mi PC virtual con el Sistema Operativo Linux(ubuntu).
Los ejercicios anteriormente mencionados son prácticas de scripting en bash, donde podemos apreciar el uso de condicionales if, bucles, funciones, programas,comandos,etc.
Dichos ejercicios se explicaran detalladamente paso a paso el funcionamiento y el camino que elegí, cabe recalcar que no se explica a fondo el funcionamiento de los comandos mencionados pero se da una idea general de los mismos.

2. RESOLUCIÓN DEL EJERCICIO 1

Mostrar mensaje "Buenas noches" y solicitar que se ingrese un texto. Mostrar el texto ingresado

```
#!/bin/bash
echo Buenas noches
echo Ingrese un texto
read textoingresado
echo $textoingresado
```

Comencé creando un archivo con extensión "**.sh**", agregue el **shebang,** luego mediante **echo** mostre al usuario el texto *"Buenas noches"* e *"Ingrese un texto"* para luego a través del comando **read** y lo almacena en la variable "*textoingresado*",luego lo mostramos con al usuario con la línea *"echo $textoingresado"*

Una vez ejecutado el programa podemos ver lo siguiente:

```
Buenas noches
Ingrese un texto
texto de prueba
texto de prueba
```

3. RESOLUCIÓN DEL EJERCICIO 2

Armar un archivo vacío cuyo nombre consista en una entrada más un texto predeterminado

```
#!/bin/bash
echo Ingrese el nombre del archivo
read nombre
touch "$nombre""test"
```

En los siguientes ejercicios voy a omitir la creación del archivo dado que es el mismo proceso para todos. Para comenzar agregue el **shebang,** en la siguiente línea utilizo el comando **echo** para mostrar al usuario el texto *"Ingrese el nombre del archivo",* luego con el comando **read** solicitamos al usuario que ingrese un "nombre" para el archivo, el cual se asigna a la variable *nombre* luego con el comando **touch** creamos un archivo, este comando nos solicita unos parámetros, en este caso, agregué el valor de la variable *nombre*, anteriormente asignado por el usuario, más un texto predeterminado que en esta ocasión seria *"test"*.

Al ejecutar el script podemos ver la siguiente salida en la terminal:

```
Ingrese el nombre del archivo
esunarchivo
```

Luego en el lugar donde tenemos el script podremos ver la creación del archivo

esunarchivo
test

4. RESOLUCIÓN DEL EJERCICIO 3

Calcular días transcurridos entre dos fechas

Siendo este en mi consideración el ejercicio más complicado del trabajo práctico debido a que no está permitido el uso del comando **date**, se hace muy difícil calcular la fluctuación de los días que contiene cada mes y evaluar si es año bisiesto o no, dicho esto puede ser que tenga una variación entre la cantidad de días por lo anteriormente mencionado.
Dado a que el script quedó extenso lo voy a ir explicando por sectores.
Primer sector: Ingreso de datos

```
#!/bin/bash
echo "Ingrese el  año de la primera fecha "
read year1
echo "ingrese el mes de la primera fecha"
read mes1
echo " ingrese el dia de la primera fecha"
read dia1
echo "Ingrese el año de la segunda fecha"
read year2
echo "Ingrese el mes de la segunda fecha"
read mes2
echo "Ingrese el dia de la segunda fecha"
read dia2
```

Comenzando nuevamente agregando el **shebang**, luego a través del comando **echo** muestro al usuario el texto *"Ingrese el año de la primera fecha",* seguido del comando **read** que solicitará al usuario ingresar el valor de la variable *year1*, luego realizamos los mismos pasos para solicitar los días y los meses.

Si bien podría haberlo realizado de otra manera, como por ejemplo solicitar la fecha a través del comando read de manera que quedara una sintaxis similar a "read year1 mes1 dia1" , la utilizada me pareció la más adecuada en el momento de realizarlo.

Segundo sector: Procesamiento de datos
Este sector es el que se encarga de procesar y operar con los datos obtenidos, el mismo lo podemos dividir en 3 subsectores (**año,mes,día**), el nombre del sector da a entender cuáles datos son los que se están procesando en el mismo.

subsector año:

```
if [[ $year1 -gt $year2 ]];
then
difA=$((year1-year2))
elif [[ $year2 -gt $year1 ]];
then
difA=$((year2-year1))
else
difA=0
fi
difAT=$((difA * 365))
```

Como su nombre lo dice, en este subsector se va a procesar los años, para comenzar compare el año de la primera fecha con el año de la segunda utilizando un **if** donde se pregunta si *year1* es mayor que (**-gt**) *year2,* en caso de que la condición se cumpla se le asignará el valor a la variable *difA,* la cual va a

valer el resultado de la operación de *year1* menos *year2.*
Luego comparó lo mismo que en el **if** anterior pero de forma inversa, para así tener la segunda de las tres posibles variaciones, con un **else** cubrimos la opción de que ambos años sean iguales por lo cual se le asigna el valor de 0 a la variable *difA.*
Por último convierto los años en días multiplicando la variable *difA* por 365 y asignándole ese valor a la variable *difAT*.

subsector mes:

```
if [[ $mes1 -gt $mes2 ]];
then
difM=$((mes1 - mes2))
elif [[ $mes2 -gt $mes1 ]];
then
difM=$((mes2 - mes1))
else
difM=0
fi
difAM=$(( difM * 31 ))
```

En este sector cálculo el los meses,
de la misma manera que calculamos los años pero en este caso a la variable difAM la multiplique por 31 para obtener la cantidad de días.

subsector dia:

```
if [[ $dia1 -gt $dia2 ]];
then
difD=$(( dia1 - dia2 ))
elif [[ $dia2 -gt $dia1 ]];
then
difD=$(( dia2 - dia1))
else
difD=0
fi
```

Para este último subsector realizamos lo mismo anteriormente planteado, pero en este caso no necesitamos pasarlos a días porque el valor ya se encuentre en ese tipo.

Tercer sector: sumar total de días y mostrarlo

```
diferencia_total=$((difD + DifAM + difAT))
echo La diferencia de dias es $diferencia_total
```

Por último cree una nueva variable *diferencia_total* , la cual contiene la suma de todo pasado a días , para luego a través del comando **echo** mostrarle al usuario cuantos días hay de diferencia entre las fechas.
Al ejecutar el script se podrá ver lo siguiente:

```
Ingrese el  año de la primera fecha
1999
ingrese el mes de la primera fecha
10
 ingrese el dia de la primera fecha
10
Ingrese el año de la segunda fecha
2000
Ingrese el mes de la segunda fecha
10
Ingrese el dia de la segunda fecha
10
La diferencia de dias es 365
```

5. RESOLUCIÓN DEL EJERCICIO 4

Contar cantidad de letras en una palabra

Para este ejercicio coloque el **shebang**, luego a través del comando **read** con el parámetro **-p** el cual nos permite escribir un texto ,le pedimos al usuario que ingrese una palabra y esta se guardará en la variable *palabra*, luego creamos una variable llamada letras la cual contendrá la cantidad de caracteres que contiene la variable *palabra*, ya que al escribirlo de la siguiente manera *${#palabras}* nos devolverá la cantidad de caracteres y por último con el comando **echo** le mostramos al usuario la cantidad de caracteres.

```
#!/bin/bash
read -p "ingrese la palabra: " palabra
letras=${#palabra}
echo $letras
```

Al ejecutar el script lo que podemos observar en la terminal es:

```
ingrese la palabra: Banfield
8
```

6. RESOLUCIÓN DEL EJERCICIO 5

Definir si un número es o no primo

Inicio colocando el shebang, luego a través del comando **echo** le muestro el texto "Ingrese un numero" al usuario para luego con el comando read solicitarle el ingreso de un *numero* el cual se guardará en la variable numero.

```
#!/bin/bash
echo Ingrese un numero
read numero
primo=1
for ((i=2; i<=numero/2;i++))
do
        if [[ $((numero % i)) -eq 0 ]]
        then
                primo=0
        fi
done
```

Después cree una variable llamada *primo* la cual inicializamos en 1 porque la vamos a utilizar como si fuera una variable lógica.
luego coloco un for e inicializo la variable *i* en 2 porque no es necesario utilizar divisores menores a 2 para verificar si un número es primo.
En la condición del for se establece que el ciclo continua mientras que *i* sea menor a

la variable *numero* dividido 2 , esto es debido a que un número no puede tener un divisor mayor que su mitad. después de cada iteración *i* se incrementa en 1.
Luego en cada iteración utilice un condicional **if,** el cual evalúa si el modulo de *numero* divido *i,* si el módulo da 0 entonces el número es primo, cumplida dicha función cambiamos de valor la variable *primo.*
*Por último realizamos las correspondientes comparaciones con un **if**, preguntando si son iguales (**-eq**)*

```
if [[ $numero -eq 1 ]];
then
echo "es primo"
fi
if [[ $primo -eq 1 ]];
then
echo "es primo"
else
echo "no es primo"
fi
```

Al ejecutar el script en la terminal podremos observar lo siguiente:

```
Ingrese un numero
73
es primo
```

7. RESOLUCIÓN DEL EJERCICIO 6

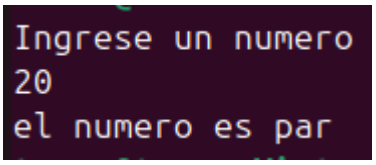
Definir si un número es par o impar

```
#!/bin/bash
echo Ingrese un numero
read numero

if [[ $((numero % 2)) -eq 0 ]];
then
echo el numero es par
else
echo el numero es impar
fi
```

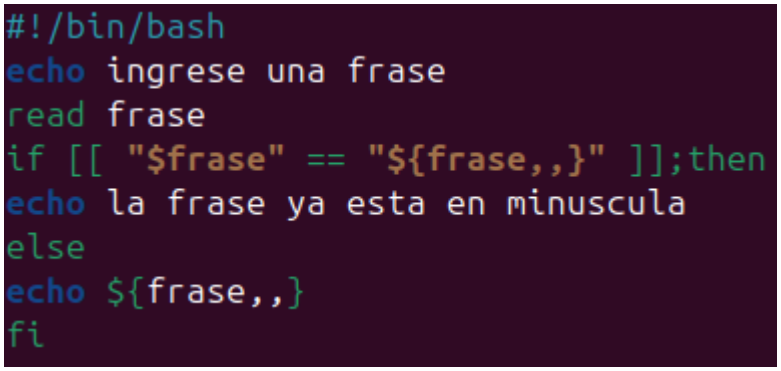
Para este ejercicio colocamos el shebang, con el comando echo muestro el usuario el texto “Ingrese un numero”, para luego con el comando read solicitar al usuario ingresar un valor, el cual se almacenará en la variable *numero.*
luego con un if que evalúa el modulo de numero dividido 2 es igual (-eq) a 0,

si la condición se cumple el número ingresado es par, por lo que con el comando echo le muestro al usuario, en caso contrario el número es impar.
al ejecutar el script lo que podemos ver en la terminal es:

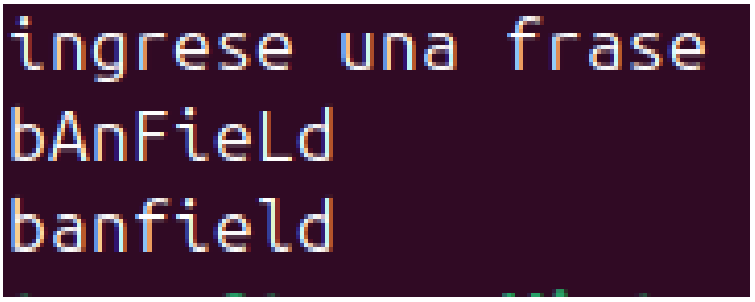


8. RESOLUCIÓN DEL EJERCICIO 7

Convertir una frase de mayúsculas a minúsculas y verificar si esta ingresado todo en minúsculas.



Inicie colocando el **shebang**, luego con el comando **echo** le muestro al usuario el texto *"ingrese una frase"*, luego con el comando **read** se le solicita al usuario ingresar la frase antes mencionadas, la cual se guardará en la variable *frase*. Abro un **if** en el que se evalúa si la frase ingresada ya se encuentra en minúscula con el parámetro (*${frase,,}*) , si la condición se cumple muestra que la frase ya están en minúscula, en caso de que la frase no este en minúscula se le muestra al usuario con el comando ***echo*** la frase ingresa en minúscula.
Al ejecutar el script podemos ver lo siguiente en la terminal:



9. RESOLUCIÓN DEL EJERCICIO 8

Informar entre dos numero cuál es el mayor y cuál es el menor

```
#!/bin/bash
echo Ingrese el primer numero
read numero1
echo Ingrese el segundo numero
read numero2

if [[ $numero1 -lt $numero2 ]];
then
echo $numero2 es mayor y $numero1 es menor
else
echo $numero1 es mayor y $numero2 es menor
fi
```

Como en los ejercicios anteriores comenzamos colocando el **shebang**,luego con el comando **echo** le mostramos al usuario el texto *"Ingrese el primer número"* después con el comando **read** le solicito al usuario ingresar un valor que se va a almacenar en la variable llamada *numero1,* realizamos el mismo proceso para el segundo nú
mero.
Luego con el condicional **if** evalúa si *numero1* es menor(**-lt**) que *numero2*, si la condición se cumple se muestra al usuario con el comando **echo** que el *numero2* es mayor a *numero1*, en caso de que la condición no se cumpla se le muestra al usuario que el *numero1* es mayor que *numero2.*

Al ejecutar el script en la terminal podremos observar lo siguiente:

```
Ingrese el primer numero
29
Ingrese el segundo numero
12
29 es mayor y 12 es menor
```

10. RESOLUCIÓN DEL EJERCICIO 9

Informar el resultado de la suma, resta, división, multiplicación y potencia entre dos números.

```
#!/bin/bash
echo Ingrese el primer numero
read numero1
echo ingrese el segundo numero
read numero2

echo "La suma es $numero1 + $numero2 = " $(( numero1 + numero2))
echo "La resta es $numero1 - $numero2 = " $(( numero1 - numero2))
echo "La division es $numero1 / $numero2 = "$((numero1 / numero2))
echo "La multiplicacion es $numero1 * $numero2 = "$((numero1 * numero2))
echo "La potencia es $numero1 ** $numero2 = "$((numero1 ** numero2))
```

Nuevamente iniciamos colocando el **shebang**, con el comando **echo** le muestro al usuario el texto *"Ingresa el primer número"* , luego con el comando **read** le solicito al usuario que ingrese un valor el cual se asignara en la variable *numero1*, el mismo procedimiento utilizado anteriormente pero esta vez para la variable *numero2*.
Luego de obtener los datos, con el comando **echo** le mostramos al usuarios el resultado de las operaciones solicitadas en el enunciado.
Al ejecutar el script en la terminal podremos observar lo siguiente:

```
Ingrese el primer numero
8
ingrese el segundo numero
8
La suma es 8 + 8 =  16
La resta es 8 - 8 =  0
La division es 8 / 8 = 1
La multiplicacion es 8 * 8 = 64
La potencia es 8 ** 8 = 16777216
```

11. RESOLUCIÓN DEL EJERCICIO 10

Calcular el promedio entre 5 números

```
#!/bin/bash
echo Ingrese las 5 notas separadas por un espacio
read nota1 nota2 nota3 nota4 nota5
promedio=$(((nota1 + nota2 +nota3 + nota4 + nota5)/5))
echo el promedio es de $promedio
```

Para comenzar coloque el el **shabang**, luego a través del comando **echo** le mostrar al usuario el texto *"Ingrese las 5 notas separadas por un espacio"*, luego con el comando read le pedimos al usuario que ingrese 5 valores, los cuales se van a ir almacenando en las variables *nota1,nota2*,etc. Luego creamos una variable llamada *promedio* la que va a contener el valor de las sumas de todas las variables *nota* divido 5, nuevamente con el comando **echo** mostramos al usuario la variable *promedio.*
Al ejecutar el script en la terminal podremos observar:

```
Ingrese las 5 notas separadas por un espacio
8 6 5 9 10
el promedio es de 7
```

12. RESOLUCIÓN DEL EJERCICIO 11

Indicar cuál es la palabra de mayor longitud entre 5 palabras

```
#!/bin/bash
echo ingrese 5 palabras separadas por un espacio
read palabra1 palabra2 palabra3 palabra4 palabra5
mayor=$palabra1
if [[ ${#mayor} -lt ${#palabra2} ]];
then
mayor=$palabra2
fi
if [[ ${#mayor} -lt ${#palabra3} ]];
then
mayor=$palabra3
fi
if [[ ${#mayor} -lt ${#palabra4} ]];
then
mayor=$palabra4
fi
if [[ ${#mayor} -lt ${#palabra5} ]];
then
mayor=$palabra5
fi

echo la de mayor logitud es $mayor que tiene ${#mayor} letras
```

Inicio colocando el **shebang,** luego con el comando **echo** le muestro el texto *"ingrese 5 palabras separadas por un espacio"* al usuario, después con el comando **read** solicitamos al usuario que ingrese 5 valores los cuales se van almacenar en las variables *palabra1,palabra2* etc.

Inicializamos como mayor a la primera palabra en la variable llamada *mayor*, luego comparamos con un **if** las siguientes 4 palabra, en caso de que la palabra que esta en la variable mayor sea menor que la variable siguiente con la cual la estoy comparando esta tomará el valor de la otra palabra, en caso de que no sea mayor se mantendrá la misma palabra, luego de realizar todas las comparaciones posible con el comando **echo** le muestro al usuario cual es el resultado final y la cantidad de letras que tiene la palabra.

Si ejecutamos el script podremos ver el siguiente resultado:

```
ingrese 5 palabras separadas por un espacio
banfield messi brus gato pez
la de mayor logitud es banfield que tiene 8 letras
```

13. RESOLUCIÓN DEL EJERCICIO 12

Según una tabla de puntuación (I/B/MB/E), mostrar calificación según nota ingresada.

```
#!/bin/bash
echo Ingrese las nota
read nota

if [[ nota -lt 6 ]];
then
echo "la nota es: I" I
elif [[ nota -lt 8 && nota -gt 5 ]];
then
echo "La nota es: B"
elif [[ nota -lt 10 && 7 ]];
then
echo "La nota es: MB"
elif [[ nota -eq 10 ]];
then
echo "la nota es: E"
elif [[ nota -gt 10 ]];
then
echo La nota ingresada es incorrecta
fi
```

Colocamos el **shebang** a continuación a través del comando **echo** le mostramos al usuario el texto *" ingrese la nota"*, luego con el comando **read** le solicitamos ingresar un valor el cual se va a almacenar en la variable llamada *nota*. Después de obtener los valores los comparamos con un **if** y en base al número que el usuario ingreso se le asigna el valor de dicha tabla a través del comando **echo** seguido del texto correspondiente.

si ejecutamos el script se verá de la siguiente manera:

```
Ingrese las nota
10
la nota es: E
```

14. RESOLUCIÓN DEL EJERCICIO 13

Ingresar usuario y contraseña (solo letras), el usuario debe aceptarse sin importar mayúsculas o minúsculas. La contraseña debe ser exacta.

```
#!/bin/bash
echo Ingrese su usuario
read usuario
echo Ingrese su contraseña
read key

clave=cuenta
user=pepe

if [[ "$usuario" ==  "$user" && "$key" ==  "${clave,,}" ]];
then
echo Iniciaste session
else
echo error verifique sus datos
fi
```

En este último ejercicio, al igual que en los anteriores, comencé colocando el shebang. luego a través del comando **echo** mostré el texto *"Ingrese su usuario"* y seguido de eso con el comando **read** solicite al usuario ingresar un valor que se guardará en la variable llamada *usuario*, repetimos el mismo proceso para la contraseña.
luego asigne el valor de las variables *clave* y *user* para después compararlo con un condicional **if** si los valores ingresados coinciden o no con los valor preestablecidos, en caso de que la condición se cumpla mostra con el comando **echo** al usuario el texto *"iniciaste session"*, en caso de no cumplirla se muestra al usuario un texto que dice *"error verifique sus datos".*
Al ejecutar el script en la terminal veríamos lo siguiente:

```
Ingrese su usuario
pepe
Ingrese su contraseña
cuenta
Iniciaste session
```