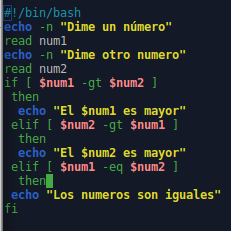
**Ejercicios Linux – Scripts**

**1. Decir cual es el mayor de 2 números.**



#!/bin/bash

echo -n "Dime un número"

read num1

echo -n "Dime otro numero"

read num2

if [ $num1 -gt $num2 ]

then

echo "El $num1 es mayor"

elif [ $num2 -gt $num1 ]

then

echo "El $num2 es mayor"

elif [ $num1 -eq $num2 ]

then

echo "Los numeros son iguales"

fi



*realmente he hecho el 2 xdddd*

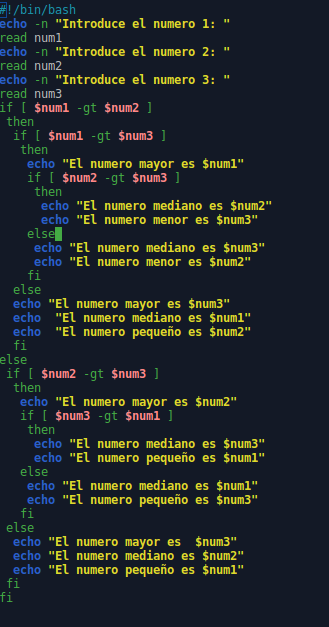
**2. Igual que el ejercicio 1, pero contemplar que sean iguales.**

**3. Ordenar 3 números de menor a mayor.**

**4. Igual que el ejercicio 3, pero de mayor a menor. (LO HICE DE MENOR A MAYOR XDD)**

#!/bin/bash

echo -n "Introduce el numero 1: "

read num1

echo -n "Introduce el numero 2: "

read num2

echo -n "Introduce el numero 3: "

read num3

if [ $num1 -gt $num2 ]

then

if [ $num1 -gt $num3 ]

then

echo "El numero mayor es $num1"

if [ $num2 -gt $num3 ]

then

echo "El numero mediano es $num2"

echo "El numero menor es $num3"

else

echo "El numero mediano es $num3"

echo "El numero menor es $num2"

fi

else

echo "El numero mayor es $num3"

echo "El numero mediano es $num1"

echo "El numero pequeño es $num2"

fi

else

if [ $num2 -gt $num3 ]

then

echo "El numero mayor es $num2"

if [ $num3 -gt $num1 ]

then

echo "El numero mediano es $num3"

echo "El numero pequeño es $num1"

else

echo "El numero mediano es $num1"

echo "El numero pequeño es $num3"

fi

else

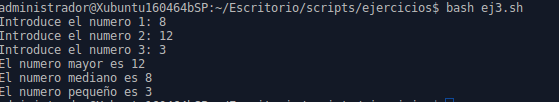
echo "El numero mayor es $num3"

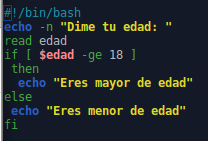
echo "El numero mediano es $num2"

echo "El numero pequeño es $num1"

fi

fi



**5.Preguntar por la edad de una persona y decir si es mayor o menor de edad.** 

#!/bin/bash

echo -n "Dime tu edad: "

read edad

if [ $edad -ge 18 ]

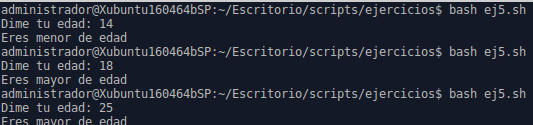
then

echo "Eres mayor de edad"

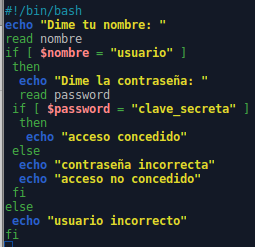
else

echo "Eres menor de edad"

fi

****

**6. Pedir un nombre y una contraseña y permitir el acceso si el nombres es “usuario” y la contraseña “clave\_secreta”.**

#!/bin/bash

echo "Dime tu nombre: "

read nombre

if [ $nombre = "usuario" ]

then

echo "Dime la contraseña: "

read password

if [ $password = "clave\_secreta" ]

then

echo "acceso concedido"

else

echo "contraseña incorrecta"

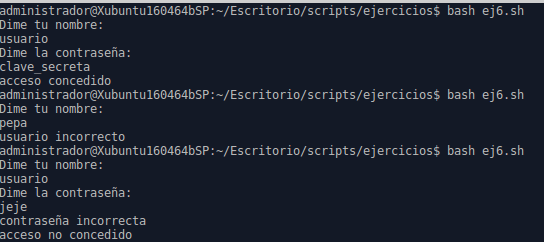
echo "acceso no concedido"

fi

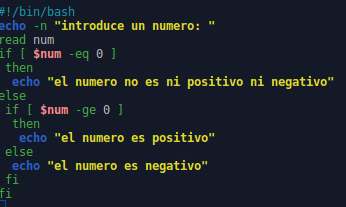
else

echo "usuario incorrecto"

fi



**7. Decir si un número es negativo o positivo.**

#!/bin/bash

echo -n "introduce un numero: "

read num

if [ $num -eq 0 ]

then

echo "el numero no es ni positivo ni negativo"

else

if [ $num -ge 0 ]

then

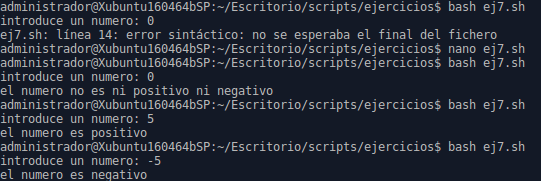
echo "el numero es positivo"

else

echo "el numero es negativo"

fi

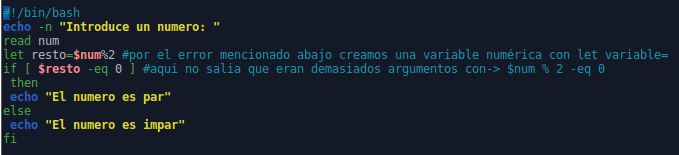
fi



**8. Decir si un número es par o impar. let: es para cálculos numéricos como= expr**

#!/bin/bash

echo -n "Introduce un numero: "

read num

let resto=$num%2

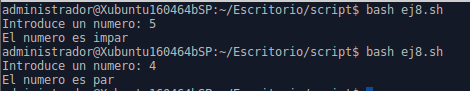
if [ $resto -eq 0 ] then

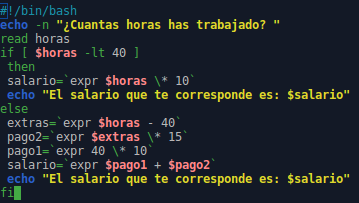
echo "El numero es par"

else

echo "El numero es impar"

fi

****

**9. Preguntar las horas que ha trabajado un empleado y calcular su sueldo (cada hora son 10€, y las horas extras a 15€. Son horas extra a partir de 40)** 

#!/bin/bash

echo -n "¿Cuantas horas has trabajado? "

read horas

if [ $horas -lt 40 ]

then

salario=`expr $horas \\* 10`

echo "El salario que te corresponde es: $salario"

else

extras=`expr $horas - 40`

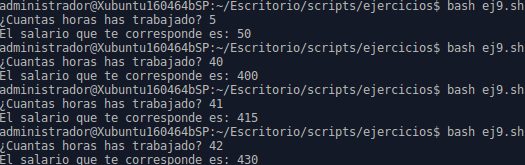
pago2=`expr $extras \\* 15`

pago1=`expr 40 \\* 10`

salario=`expr $pago1 + $pago2`

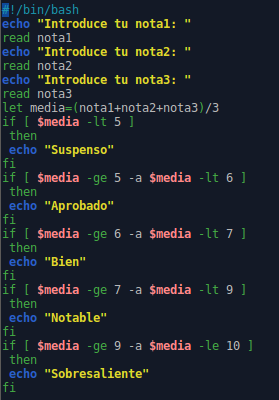
echo "El salario que te corresponde es: $salario"

fi

****

**10. Calcular la media de 3 notas y escribir la calificación literal (suspenso, aprobado, bien, notable, sobresaliente)**

#!/bin/bash

echo "Introduce tu nota1: "

read nota1

echo "Introduce tu nota2: "

read nota2

echo "Introduce tu nota3: "

read nota3

let media=(nota1+nota2+nota3)/3

if [ $media -lt 5 ]

then

echo "Suspenso"

fi

if [ $media -ge 5 -a $media -lt 6 ]

then

echo "Aprobado"

fi

if [ $media -ge 6 -a $media -lt 7 ]

then

echo "Bien"

fi

if [ $media -ge 7 -a $media -lt 9 ]

then

echo "Notable"

fi

if [ $media -ge 9 -a $media -le 10 ]

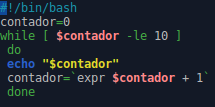
then

echo "Sobresaliente"

fi

**“SI usasemos expr tendríamos que hacer primero el cálculo de la suma de las notas y volver a poner la misma variable dividida entre 3”**

**11. Mostrar los números del 1 al 10.**



#!/bin/bash

contador=0

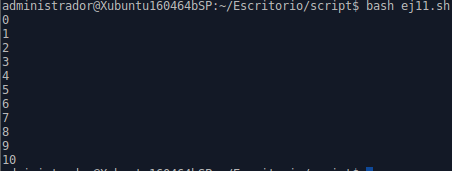
while [ $contador -le 10 ]

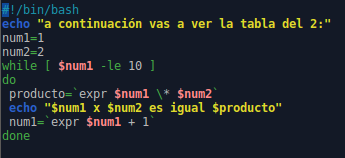
do

echo "$contador"

contador=`expr $contador + 1`

done

****

**12. Mostrar la tabla de multiplicar de un número.** 

#!/bin/bash

echo "a continuación vas a ver la tabla del 2:"

num1=1

num2=2

while [ $num1 -le 10 ]

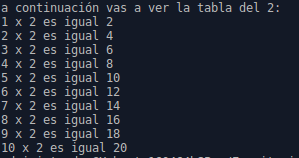
do

producto=`expr $num1 \\* $num2`

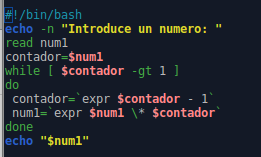
echo "$num1 x $num2 es igual $producto"

num1=`expr $num1 + 1`

done

****

**13. Calcular el factorial de un número.**

#!/bin/bash

echo -n "Introduce un numero: "

read num1

contador=$num1

while [ $contador -gt 1 ]

do

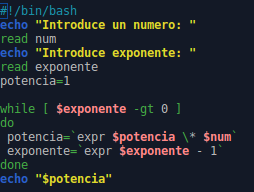
contador=`expr $contador - 1`

num1=`expr $num1 \\* $contador`

done

echo "$num1"

**14. Calcular mediante multiplicaciones la potencia de un número.**

#!/bin/bash

echo "Introduce un numero: "

read num

echo "Introduce exponente: "

read exponente

potencia=1

while [ $exponente -gt 0 ]

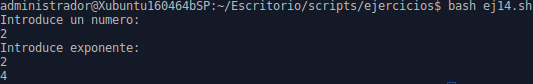
do

potencia=`expr $potencia \\* $num`

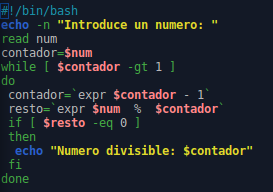
exponente=`expr $exponente - 1`

done

echo "$potencia"



**15. Mostrar los divisores de un número.**



#!/bin/bash

echo -n "Introduce un numero: "

read num

contador=$num

while [ $contador -gt 1 ]

do

contador=`expr $contador - 1`

resto=`expr $num % $contador`

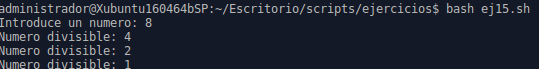
if [ $resto -eq 0 ]

then

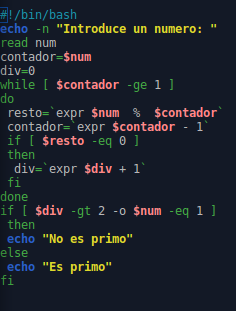
echo "Numero divisible: $contador"

fi

done

****

**16. Decir si un número es primo.**



#!/bin/bash

echo -n "Introduce un numero: "

read num

contador=$num

div=0

while [ $contador -ge 1 ]

do

resto=`expr $num % $contador`

contador=`expr $contador - 1`

if [ $resto -eq 0 ]

then

div=`expr $div + 1`

fi

done

if [ $div -gt 2 -o $num -eq 1 ]

then

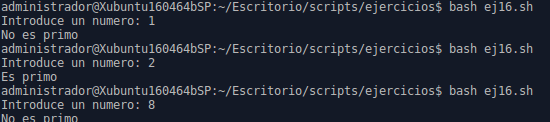
echo "No es primo"

else

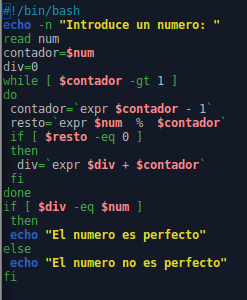
echo "Es primo"

fi

1

****

**17. Decir si un número es perfecto (un número es perfecto si es igual a la suma de sus divisores verdaderos)**

#!/bin/bash

echo -n "Introduce un numero: "

read num

contador=$num

div=0

while [ $contador -gt 1 ]

do

contador=`expr $contador - 1`

resto=`expr $num % $contador`

if [ $resto -eq 0 ]

then

div=`expr $div + $contador`

fi

done

if [ $div -eq $num ]

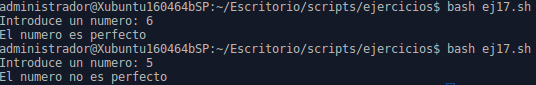
then

echo "El numero es perfecto"

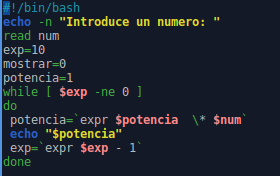
else

echo "El numero no es perfecto"

fi

****

**18. Mostrar las 10 primeras potencias de 2.**

#!/bin/bash

echo -n "Introduce un numero: "

read num

exp=10

mostrar=0

potencia=1

while [ $exp -ne 0 ]

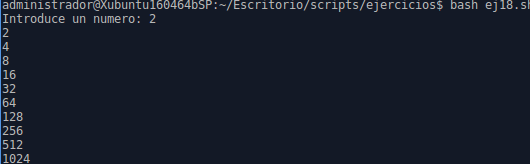
do

potencia=`expr $potencia \\* $num`

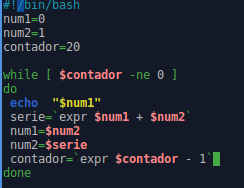
echo "$potencia"

exp=`expr $exp - 1`

done

****

**19. Calcular los 20 primeros números de la serie de Fibonacci (los dos primeros números son 0 y 1, el resto es la suma de los dos anteriores. Ej. 0,1,1,2,3,5,8,13,...)**

#!/bin/bash

num1=0

num2=1

contador=20

while [ $contador -ne 0 ]

do

echo "$num1"

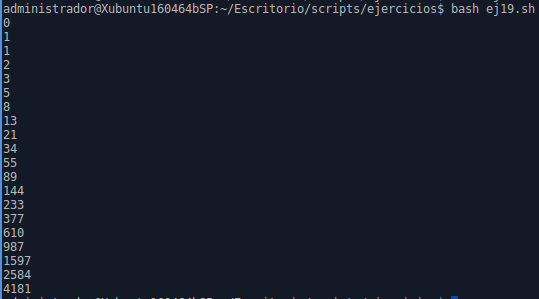
serie=`expr $num1 + $num2`

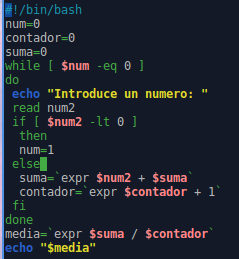
num1=$num2

num2=$serie

contador=`expr $contador - 1`

done



**20. Mostrar la media de todos los números introducidos hasta que se introduzca un número negativo.** 

#!/bin/bash

num=0

contador=0

suma=0

while [ $num -eq 0 ]

do

echo "Introduce un numero: "

read num2

if [ $num2 -lt 0 ]

then

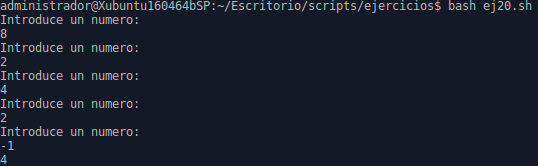
num=1

else

suma=`expr $num2 + $suma`

contador=`expr $contador + 1`

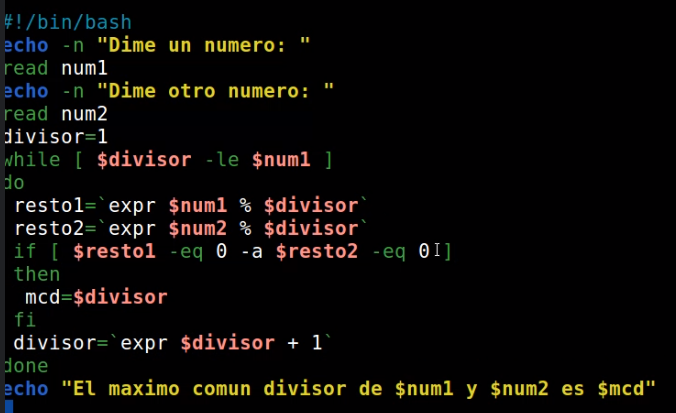
fi

done

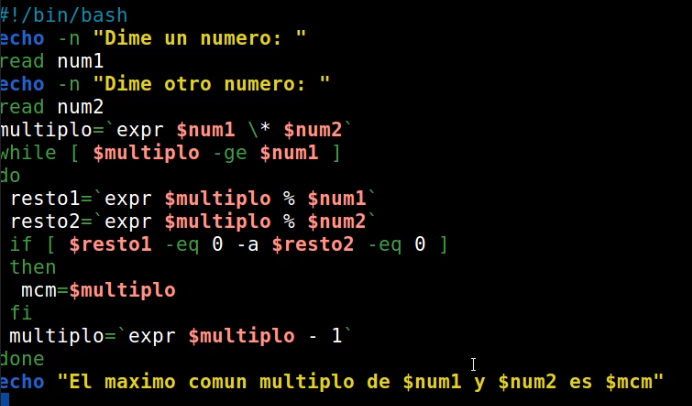
media=`expr $suma / $contador`

echo "$media"

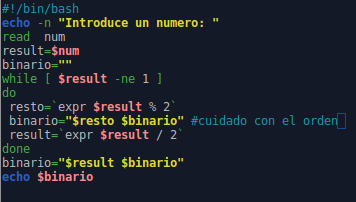
**21. Hallar el máximo común divisor de 2 números.**



**22. Hallar el mínimo común múltiplo de 2 números.**

****

**23. Realiza un script que pase de decimal a binario.**

#!/bin/bash

echo -n "Introduce un numero;"

read num

result=$num

binario=""

while [ $result -ne 1 ]

do

resto=`expr $result % 2`

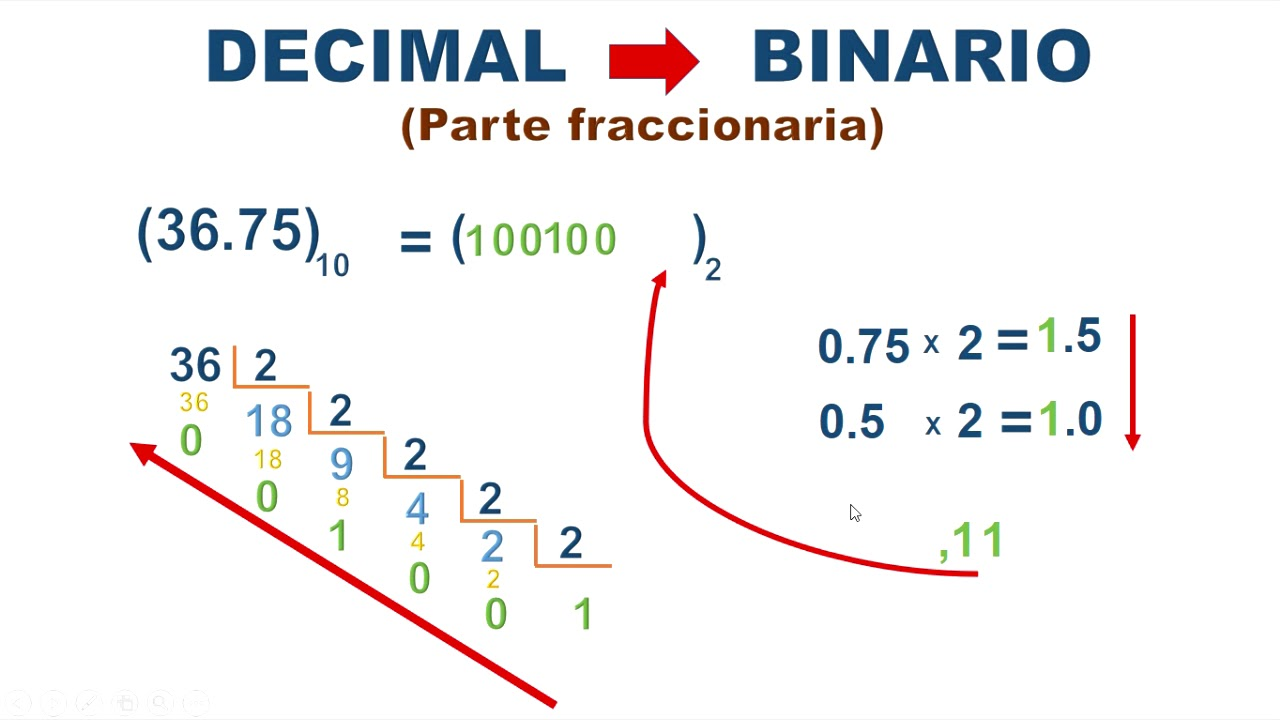
binario="$resto $binario"

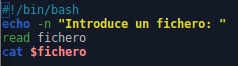
result=`expr $result / 2`

done

binario="$result $binario"

echo $binario



**24. Realiza un script que pida al usuario que introduzca un fichero y a continuación muestre su contenido.** 

#!/bin/bash

echo -n "Introduce un fichero: "

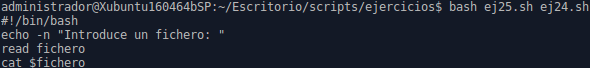
read fichero

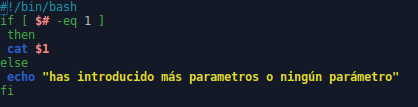
cat $fichero

**25. Igual que el anterior pero indicando el fichero como parámetro del script.** 

#!/bin/bash

cat $1

****

**26. Igual que el anterior pero comprobando que sólo se ha introducido un parámetro, mostrar error en caso contrario.** 

#!/bin/bash

if [ $# -eq 1 ]

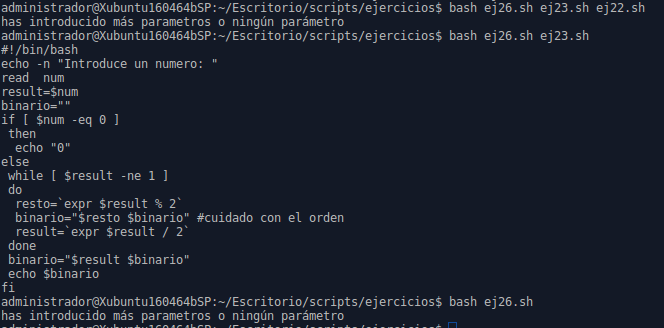
then

cat $1

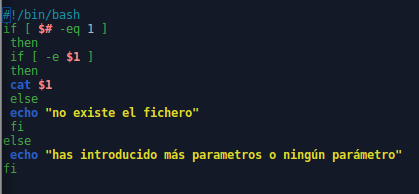
else

echo "has introducido más parametros o ningún parámetro"

fi



**27. Igual que el anterior pero comprobando si el fichero existe, avisar de lo contrario.**

#!/bin/bash

if [ $# -eq 1 ]

then

if [ -e $1 ]

then

cat $1

else

echo "no existe el fichero"

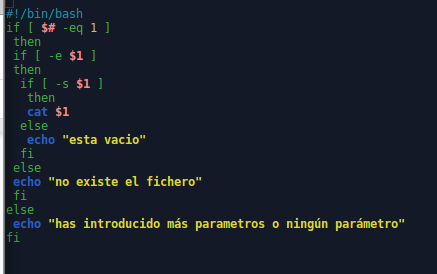
fi

else

echo "has introducido más parametros o ningún parámetro"

fi

**28. Igual que el anterior pero comprobando que el fichero no está vacío.**

#!/bin/bash

if [ $# -eq 1 ]

then

if [ -e $1 ]

then

if [ -s $1 ]

then

cat $1

else

echo "esta vacio"

fi

else

echo "no existe el fichero"

fi

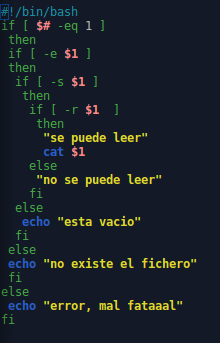
else

echo "has introducido más parametros o ningún parámetro"

fi

****

**29. Igual que el anterior pero comprobando que se puede leer. Dar información en todos los casos en que ocurra un error. Si no existe ningún error entonces se muestra el fichero.**

#!/bin/bash

if [ $# -eq 1 ]

then

if [ -e $1 ]

then

if [ -s $1 ]

then

if [ -r $1 ]

then

"se puede leer"

cat $1

else

"no se puede leer"

fi

else

echo "esta vacio"

fi

else

echo "no existe el fichero"

fi

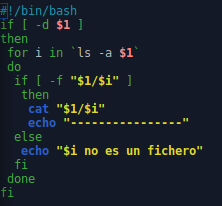
else

echo "error, mal fataaal"

fi

**30. Igual que el anterior Incluyendo que si se ejecuta el script con el parámetro -h muestre ayuda sobre el manejo del script. (ES FÁCIL PERO QUE PEREZA)**

**31. Mostrar el contenido de todos los ficheros de un directorio.**

#!/bin/bash

if [ -d $1 ]

then

for i in `ls -a $1`

do

if [ -f "$1/$i" ]

then

cat "$1/$i"

echo "----------------"

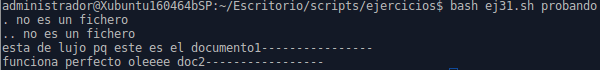
else

echo "$i no es un fichero"

fi

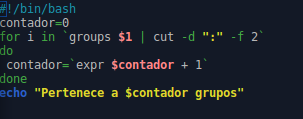
done

fi



**32. Mostrar a cuantos grupos pertenece el usuario.**

#!/bin/bash

contador=0

for i in `groups $1 | cut -d ":" -f 2`

do

contador=`expr $contador + 1`

done

echo "Pertenece a $contador grupos"

*cut para quitarte los dos que cuenta demás por los “:” y “administrador”*

*Sin cut:*



Con cut:



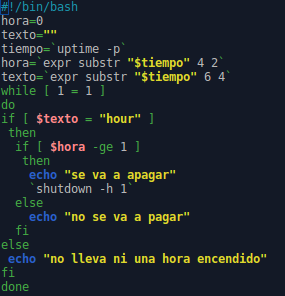
**33. Mostrar cuanto ocupa el directorio del usuario.**

#!/bin/bash

**du -sh** /home/$1/



**34. Crear un demonio que apague el ordenador si este lleva más de 8 horas encendido*. (ESTO ES CON UN GREP PERO ES BASTANTE LIO)***

#!/bin/bash

hora=0

texto=""

tiempo=`uptime -p`

hora=`expr substr "$tiempo" 4 2`

texto=`expr substr "$tiempo" 6 4`

while [ 1 = 1 ]

do

if [ $texto = "hour" ]

then

if [ $hora -ge 1 ]

then

echo "se va a apagar"

`shutdown -h 1`

else

echo "no se va a pagar"

fi

else

echo "no lleva ni una hora encendido"

fi

done

**FORMA BIEN HECHA ej 34\_2.sh**



while true

do

horas=$(uptime -p | grep -e “hour” | cut -d “ “ -f 2)

if [ -n “$horas” ]

then

if [ $horas -ge 8 ]

then

poweroff

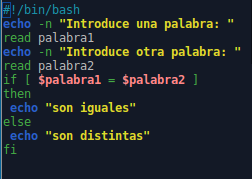
fi

fi

done

**36. Realizar un script que pida dos palabras y muestre si son iguales o distintas.**

#!/bin/bash

echo -n "Introduce una palabra: "

read palabra1

echo -n "Introduce otra palabra: "

read palabra2

if [ $palabra1 = $palabra2 ]

then

echo "son iguales"

else

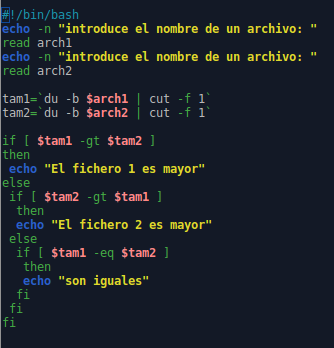
echo "son distintas"

fi

**37. Pedir dos archivos y mostrar cuál es el mayor.**

#!/bin/bash

echo -n "introduce el nombre de un archivo: "

read arch1

echo -n "introduce el nombre de un archivo: "

read arch2

tam1=`du -b $arch1 | cut -f 1`

tam2=`du -b $arch2 | cut -f 1`

if [ $tam1 -gt $tam2 ]

then

echo "El fichero 1 es mayor"

else

if [ $tam2 -gt $tam1 ]

then

echo "El fichero 2 es mayor"

else

if [ $tam1 -eq $tam2 ]

then

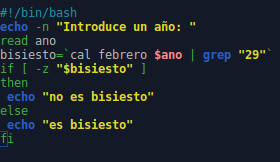
echo "son iguales"

fi

fi

fi

**38. Pedir un año e indicar si es bisiesto.**

#!/bin/bash

echo -n "Introduce un año: "

read ano

bisiesto=`cal febrero $ano | grep "29"`

if [ -z "$bisiesto" ]

then

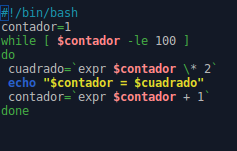
echo "no es bisiesto"

else

echo "es bisiesto"

fi

**39. Mostrar los números del 1 al 100 y sus cuadrados.**



#!/bin/bash

contador=1

while [ $contador -le 100 ]

do

cuadrado=`expr $contador \\* $contador`

echo "$contador = $cuadrado"

contador=`expr $contador + 1`

done

**40. Crear una calculadora con las operaciones básicas.**

#!/bin/bash

echo -n "Introduce un numero: "

read num1

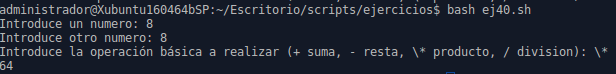
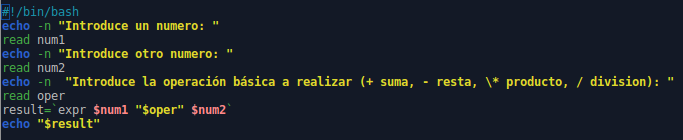
echo -n "Introduce otro numero: "

read num2

echo -n "Introduce la operación básica a realizar (+ suma, - resta, \\* producto, / division): "

read oper

result=`expr $num1 "$oper" $num2`

echo "$result"

*CUIDADO PORQUE AL NO PONER EL $OPER ENTRE COMILLAS NO LEE BIEN EL PRODUCTO \\**

**41. Crear un directorio llamado .papelera en tu carpeta personal.** **42. Realizar un script que simule la papelera de reciclaje.**

#!/bin/bash

echo -n "Introduce 1 si quieres borrar un archivo y cualquier otro numero para recuperar un archivo: "

read elec

if [ $elec -eq 1 ]

then

echo "Introduce el archivo (ruta) a borrar"

read arch

mv $arch /home/administrador/Escritorio/.papelera

echo "archivo movido a la papelera"

else

ls /home/administrador/Escritorio/.papelera

echo "Elige el archivo que quieres recuperar de la papelera: "

read arch2

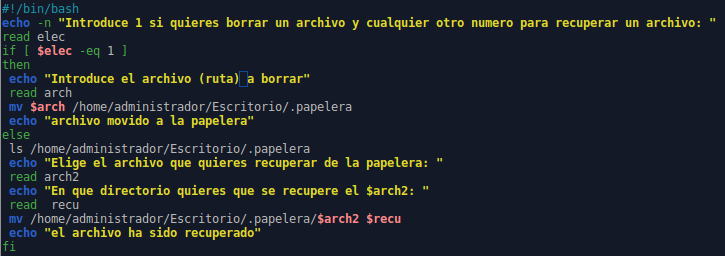
echo "En que directorio quieres que se recupere el $arch2: "

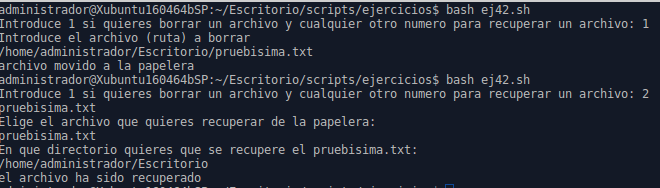
read recu

mv /home/administrador/Escritorio/.papelera/$arch2 $recu

echo "el archivo ha sido recuperado"

fi

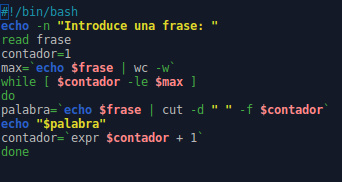




**43. Basándote en el ejercicio anterior realizar un script que permita recuperar un fichero de la papelera de reciclaje.**

**Hecho en el anterior**

**44. Pedir una frase y mostrar cada palabra en una línea distinta.**

!/bin/bash

echo -n "Introduce una frase: "

read frase

contador=1

max=`echo $frase | wc -w`

while [ $contador -le $max ]

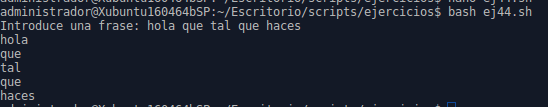
do

palabra=`echo $frase | cut -d " " -f $contador`

echo "$palabra"

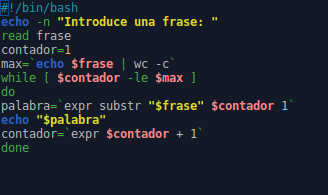
contador=`expr $contador + 1`

done



**45. Pedir una frase y mostrar cada letra en una línea distinta.**

#!/bin/bash

echo -n "Introduce una frase: "

read frase

contador=1

max=`echo $frase | wc -c`

while [ $contador -le $max ]

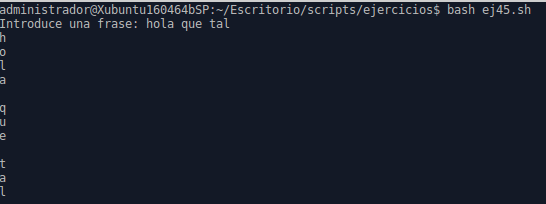
do

palabra=`expr substr "$frase" $contador 1`

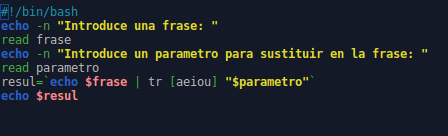
echo "$palabra"

contador=`expr $contador + 1`

done



**46. Pedir una frase y mostrarla sustituyendo todas las vocales por una vocal que te hayan introducido como parámetro.**

****

**35. Crear un script que pida al usuario un porcentaje de color rojo, igual para el verde y para el azul. Generar el número hexadecimal que describe ese color en html.**

#!/bin/bash

hexadecimal() {

case $1 in

10) echo "A" ;;

11) echo "B" ;;

12) echo "C" ;;

13) echo "D" ;;

14) echo "E" ;;

15) echo "F" ;;

\*) echo "$1" ;;

esac

}

echo -n "Introduce el porcentaje de rojo: "

read rojo

echo -n "Introduce el porcentaje de verde: "

read verde

echo -n "Introduce el porcentaje de azul: "

read azul

rojo=`expr $rojo \\* 255 / 100`

verde=`expr $verde \\* 255 / 100`

azul=`expr $azul \\* 255 / 100`

r1=`expr $rojo / 16`

r2=`expr $rojo % 16`

v1=`expr $verde / 16`

v2=`expr $verde % 16`

a1=`expr $azul / 16`

a2=`expr $azul % 16`

r1=$(hexadecimal $r1)

r2=$(hexadecimal $r2)

v1=$(hexadecimal $v1)

v2=$(hexadecimal $v2)

a1=$(hexadecimal $a1)

a2=$(hexadecimal $a2)

#case $r1 in

# 10) r1="A" ;;

# 11) r1="B" ;;

# 12) r1="C" ;;

# 13) r1="D" ;;

# 14) r1="E" ;;

# 15) r1="F" ;;

#esac

color="$r1$r2$v1$v2$a1$a2"

echo "Color Hexadecimal: $color"