

APELLIDOS Y NOMBRE:

NOTA:

PARTE PRACTICA

2) Crear un Shell Script de nombre “examen.bash” que ejecute las siguientes acciones:

(vale 3,5 puntos):

-----El programa tiene que mostrara el siguiente menú de inicio.

```
administrador@administrador-VirtualBox:~$ bash examen.sh
-----MENU-----
1.---PERMISOS DE UN ARCHIVO
2.---CONTENIDO DEL ARCHIVO
3.---GRUPOS SECUNDARIOS DE USUARIOS
4.---SALIR
Selecciona una opción:
```

En el caso de que el usuario introduzca un valor distinto de 1,2,3,4 el mensaje que mostraremos será: “No has introducido una opción válida” y retorna a visualizar el menú anterior.

```
administrador@administrador-VirtualBox:~$ bash examen.sh
-----MENU-----
1.---PERMISOS DE UN ARCHIVO
2.---CONTENIDO DEL ARCHIVO
3.---GRUPOS SECUNDARIOS DE USUARIOS
4.---SALIR
Selecciona una opción:6
No has introducido una opción valida
```

-----Cuando el usuario introduzca la opción 1.

- Se presenta el siguiente mensaje al usuario:
“INTRODUCE EL NOMBRE DE UN ARCHIVO:”
El usuario tiene que proporcionar un nombre de archivo: Tabla.bash
- Si el archivo (Tabla.bash) no existe, presenta un mensaje que diga “Archivo inexistente y vuelve a mostrar el menú.
- Si el archivo si existe el script debe analizar los permisos del archivo, relativos al grupo y presentará los mensajes de acuerdo con los permisos que tenga el archivo:

```
INTRODUCE EL NOMBRE DE UN ARCHIVO: 
```

```
INTRODUCE EL NOMBRE DE UN ARCHIVO:LISTA
El grupo tiene permiso de lectura
El grupo no tiene permiso de escritura
El grupo no tiene permiso de ejecución
-----MENU-----
      1.---PERMISOS DE UN ARCHIVO
      2.---CONTENIDO DEL ARCHIVO
      3.---GRUPOS SECUNDARIOS DE USUARIOS
      4.---SALIR
Selecciona una opción:█
```

Pues al hacer un `ls -l` analizamos los permisos solo del grupo, y si tiene lectura, escritura o ejecución.

Si no tiene permisos de lectura, escritura o ejecución indicamos:” El grupo no tiene permiso de lectura, escritura o ejecución.

Si hay algún error en los permisos “Error de permisos”

Después vuelves a mostrar el menú.

-----Cuando el usuario introduzca la opción 2.

- Presenta el siguiente mensaje al usuario:
“INTRIDUCE ARCHIVO A VISUALIZAR:”
El usuario tiene que proporcionar un nombre de archivo ejemplo: `examen.bash`
- Si el archivo no existe, presenta un mensaje que diga “Archivo inexistente y vuelve a mostrar el menú.
- Si el archivo si existe el script debe mostrar el contenido del archivo, y después volver a mostrar el menú.

```
INTRODUCE ARCHIVO A VISUALIZAR: examen.bash█
```

-----Cuando el usuario introduzca la opción 3.

- Se presenta el siguiente mensaje al usuario:
“USUARIO PARA BUSCAR SUS GRUPOS:”
Tenemos que proporcionar un nombre de usuario, y se validará la existencia en el sistema del usuario. En caso de que el usuario si exista en el sistema, nos listará los grupos a los que pertenece ese usuario. Vamos a considerar que el grupo principal de cada usuario se llama igual que el usuario

Administración de Sistemas Operativos

```
USUARIO PARA BUSCAR SUS GRUPOS:administrador
USUARIO administrador EXISTE EN EL SISTEMA SU GRUPO PRINCIPAL administrador

El usuario administrador tambien pertenece al grupo cdrom
El usuario administrador tambien pertenece al grupo plugdev
El usuario administrador tambien pertenece al grupo lpadmin
-----MENU-----
1.---PERMISOS DE UN ARCHIVO
2.---CONTENIDO DEL ARCHIVO
3.---GRUPOS SECUNDARIOS DE USUARIOS
4.---SALIR
Selecciona una opción:
```

En caso de que el usuario si exista en el sistema, pero no pertenezca a otros grupos, solo nos dirá el grupo principal del usuario y retorna a visualizar el menú.

```
USUARIO PARA BUSCAR SUS GRUPOS:ana
USUARIO ana EXISTE EN EL SISTEMA SU GRUPO PRINCIPAL ana

-----MENU-----
1.---PERMISOS DE UN ARCHIVO
2.---CONTENIDO DEL ARCHIVO
3.---GRUPOS SECUNDARIOS DE USUARIOS
4.---SALIR
Selecciona una opción:█
```

En caso negativo nos mostrara un mensaje diciendo “Este usuario no pertenece al sistema” y retorna a visualizar el menú.

```
USUARIO PARA BUSCAR SUS GRUPOS:pepito
ESTE USUARIO pepito NO EXISTE EN EL SISTEMA

-----MENU-----
1.---PERMISOS DE UN ARCHIVO
2.---CONTENIDO DEL ARCHIVO
3.---GRUPOS SECUNDARIOS DE USUARIOS
4.---SALIR
Selecciona una opción:
```

-----**Cuando el usuario introduzca la opción 4.**

Termina el script.

```
#!/bin/bash
opcion=S
numero=0
lectura=0
escritura=0
ejecucion=0
archivo=0
usuario=0
while [ $opcion = S ]
do
    echo " -----MENU-----"
    echo "          1.---PERMISOS DE UN ARCHIVO"
    echo "          2.---CONTENIDO DEL ARCHIVO"
    echo "          3.---GRUPOS SECUNDARIOS DE USUARIOS"
    echo "          4.---SALIR"
    read -n1 -p "Selecciona una opción:" numero
    case $numero in
        1) clear
            read -p "INTRODUCE EL NOMBRE DE UN ARCHIVO:" archivo
            if [ ! -f $archivo ]
            then
                echo " El archivo no existe"
            else
                lectura=$(ls -l $archivo | cut -c5)
                escritura=$(ls -l $archivo | cut -c6)
                ejecucion=$(ls -l $archivo | cut -c7)
                case $lectura in
                    r) echo "El grupo tiene permiso de lectura";;
                    -) echo "El grupo no tiene permiso de lectura";;
                    *) echo " Error de los permisos";;
                esac
                case $escritura in
                    w) echo "El grupo tiene permiso de escritura";;
                    -) echo "El grupo no tiene permiso de escritura";;
                    *) echo "Error de los permisos";;
                esac
            fi
        *)
            echo "Opción no válida"
        esac
    done
done
```

```

        case $ejecucion in
            x) echo "El grupo tiene permiso de ejecución";;
            -) echo "El grupo no tiene permiso de ejecución";;
            *) echo "Error de los permisos";;
        esac
    fi;;
2)clear
    echo
    read -p "INTRODUCE ARCHIVO A VISUALIZAR:" archivo
    if [ ! -f $archivo ]
    then
        echo "El archivo no existe"
    else
        cat $archivo
    fi;;
3) clear
    echo
    read -p "USUARIO PARA BUSCAR SUS GRUPOS:" usuario
    filas=`cut -d ":" -f1 /etc/passwd | grep -i $usuario`
    if [ $? -eq 0 ]
    then
        echo "USUARIO $usuario EXISTE EN EL SISTEMA SU GRUPO PRINCIPAL ES $usuario"
    fi

```

Echo "Usuario \$usuario existe en el sistema su grupo principal es \$usuario"

```

    echo
    grupos=`grep ^[^$usuario] /etc/group | grep $usuario | cut -d ":" -f1`
    for i in $grupos
    do
        echo "    El usuario $usuario tambien pertenece al grupo $i"
    done
    else
        echo
        echo "ESTE USUARIO $usuario NO EXISTE EN EL SISTEMA"
        echo
    fi;;
4)exit;;
*) echo

```

La fila primera no se muestra de forma completa, aquí pongo la fila completa.

Grupos=`grep ^[^\$usuario] /etc/group | grep \$usuario | cut -d ":" -f1`

```

*) echo
    echo "No has introducido una opción valida"
    echo;;
esac
done

```

- 1) Crea un Shell Script llamado factorial.sh que muestre en pantalla el resultado del factorial de los números introducido como parámetro o argumento. (vale 2 puntos)

El factorial de un numero ejemplo $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$

$4! = 4 \cdot 3 \cdot 2 = 24$

Visualizar el resultado igual que el ejemplo siguiente:

Cuando no se le pasan parámetros aparece el siguiente mensaje:

“Error necesito que introduzcas parámetros” y termina el script

Cuando se le pasan parámetros, nos muestra el factorial de los números pasados como argumento.

Solo podemos hacer el factorial de los números comprendidos entre 1 y 9.

Cualquier otro valor pasado como parámetro no se halla el factorial y se muestra el mensaje “El parámetro ‘valor del parámetro’ no es un valor correcto”

```
administrador@administrador-VirtualBox:~$ bash factorialex.sh 4 3 a bc 6
+++++EL FACTORIAL DE 4 +++++
EL FACTORIAL DE 4! ES: 24
+++++EL FACTORIAL DE 3 +++++
EL FACTORIAL DE 3! ES: 6
El parametro a no es un valor correcto
El parametro bc no es un valor correcto
+++++EL FACTORIAL DE 6 +++++
EL FACTORIAL DE 6! ES: 720
```

```
#!/bin/bash

numero=0
factorial=1

if [ $# -eq 0 ]
then
    echo " error necesito que introduzcas parametros"
    exit
fi
while [ $# -ge 1 ]
do
    numero=$1
    case $numero in
        [0-9])
            factorial=1
            echo "+++++EL FACTORIAL DE $1 +++++"
            while [ $numero -gt 1 ]
            do
                factorial=$(( $factorial * $numero ))
                numero=$(( $numero - 1 ))
            done
        ;;
    esac
done
```

```
        echo " EL FACTORIAL DE $1! ES: $factorial";;  
    *) echo " El parametro $1 no es un valor correcto";;  
    esac  
    shift  
done
```

1) Puesto en examen.

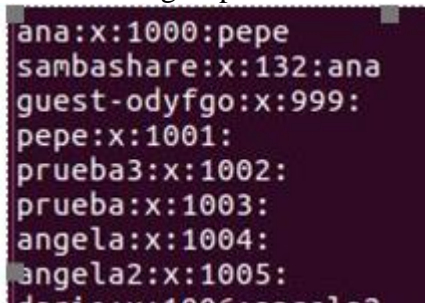
Crea un Shell script que pasándole un número indeterminado de parámetros (que son usuarios del sistema) nos muestre los grupos a los que pertenece cada usuario. Recibirá un parámetro, el usuario en cuestión. Validará la existencia en el sistema del usuario. En caso afirmativo nos listará los grupos a los que pertenece ese usuario.

En caso negativo nos mostrara un mensaje diciendo “Este usuario no pertenece al sistema”.

En caso de que no se le pasen parámetros el script mostrara un mensaje “No me has pasado parámetros” y termina.

Consideramos que el nombre del usuario y el nombre del grupo no tienen por qué ser iguales.

Fichero /etc/group



```
ana:x:1000:pepe  
sambashare:x:132:ana  
guest-odyfgo:x:999:  
pepe:x:1001:  
prueba3:x:1002:  
prueba:x:1003:  
angela:x:1004:  
angela2:x:1005:  
dario:x:1006:angela2
```

La salida por pantalla será como muestra la imagen.


```

ana@ana-VirtualBox:~$ bash examenrecu1 ana ana12 pepe
El usuario ana EXISTE EN EL SISTEMA SU GRUPO principal es ana

    el usuario ana tambien pertenece al grupo cdrom
    el usuario ana tambien pertenece al grupo sudo
    el usuario ana tambien pertenece al grupo dip
    el usuario ana tambien pertenece al grupo plugdev
    el usuario ana tambien pertenece al grupo lpadmin
    el usuario ana tambien pertenece al grupo sambashare

este usuario ana12 NO EXISTE EN EL SISTEMA

El usuario pepe EXISTE EN EL SISTEMA SU GRUPO principal es pepe

    el usuario pepe tambien pertenece al grupo ana

```

```

#!/bin/bash

filas=0
grupo=0
nombreggrupo=0

if [ $# -eq 0 ];then
    echo "No has pasado parametros"
    exit
fi
while [ $# -ne 0 ];do
    # usuario que tengo en el parametro $1 miro si esta en /etc/passwd
    # extraigo el guid
    filas=$(grep -i $1: /etc/passwd | cut -d : -f4)
    # grep -i no distinguir mayusculas y minusculas
    if [ $? -eq 0 ];then
        # busco en /etc/group el nombre del gid anterior y
        #nos quedamos con el nombre

        nombreggrupo=$( grep :$filas: /etc/group | cut -d : -f1)
        echo "El usuario $1 Existe en el sistema su grupo principal es $nombreggrupo "

        grupo=$(grep -v ^$nombreggrupo /etc/group | grep $nombreggrupo |cut -d: -f1)
        # buscamos las lineas que no comiencen por el usuario -v ^$nombreggrupo
        #y de esas lineas buscamos las lineas que
        #contienen el nombre del usuario y despues nos quedamos con
    fi
    shift
done

```

```

    for i in $grupo
    do
        echo " El usuario $1 tambien pertenece al grupo $i"
    done
    else
        echo " Este usuario $1 no Existe en el sistema"
    fi
    shift
done

```


Administración de Sistemas Operativos

grep ^ (empieza)
grep ^[^] (no empieza)
grep -v (busca todo menos
lo que tenemos)
grep -i (no distingue mayus
ni minus)

cut -d: f2 (Para que muestre - datos)
cut -c1 (El nº de caracteres que se va
a extraer)
cut -f2 (Extrae la columna 2)

du -b para el tamaño del fichero

Lo que contiene /etc/group

nombregroup:contraseña:IDgrupo:miembros
Ej:
usuario1:x:1000:user1,user2

Lo que contiene /etc/passwd

nombreusuario:contraseña:IDusuario:IDgrupo:nombreusuariootrasinformaciones:directorioinicio:shell
Ej
user1:x:1000:1000:User One:/home/user1:/bin/bash