```
**************************************************
EJERCICIO1 SE PASA UN DIRECTORIO, LO COMPRIME CON EL NOMBRE yyyy-mm-ddNombreDirec.tar.gz
#|/bin/bash
if [ $# -eq 1 ];then
        if [ -d $@ ];then
                #indicar nombre
                fecha=`date +%F`
                tar -cf $fecha$@.tar $@
                gzip $fecha$@.tar
        else
                echo "El parámetro $@ no es directorio"
        fi
        else
                echo "Debe introducir un parámetro"
fi

**************************************************
EJERCICIO 2 CLACULADORA DE OPERACIONES SOBRE 2 NÚMEROS PASADOS COMO PARÁMETROS
#!/bin/bash
menu (){
        echo "Elige opción del siguiente menú:"
        echo "
        1) suma
        2) resta
        3) producto
        4) división
        5) resto"
}
#perfecto con números enteros. Fallos con reales solucionado con ()
suma (){
        resultado=`echo "($1)+($2)" | bc -l`
        #resultado=$(($1+$2)) #sólo para enteros
}
resta(){
        resultado=`echo "($1)-($2)" | bc -l`
        #resultado=$(($1-$2))
}
producto(){
        resultado=`echo "($1)*($2)" | bc -l`
        #resultado=$(($1*$2))
}
division(){
        if [ "$2" != "0" ];then #para reales -ne es para enteros
                resultado=`echo "scale=2;($1)/($2)" | bc -l`
        else
                echo "División por 0"
        fi
}
resto(){
        if [ $2 -ne 0 ];then
                resultado=$(($1%$2)) #sólo para enteros no para reales
        else
                echo "División por 0"
        fi
}
#PROGRAMA PRINCIPAL
if [ $# -eq 2 ];then
        #comprobación de números enteros y reales
        echo $1 | egrep -q '^\-?[0-9 .,]+$'
        if [ $? -ne 0 ];then
                echo "El parámetro $1 no es número"
                exit
        fi
        echo $2 | egrep -q '^\-?[0-9 .,]+$'
        if [ $? -ne 0 ];then
                echo "El parámetro $2 no es número"
                exit
        fi
        echo "Se procede a realizar las operaciones que elijas del siguiente menú"
```

```
        menu #llamada a funciónread -p "Introduce opción del menú: " resp
        echo $resp | egrep -q '^?[0-9 .,]+$' #nos aseguramos números
        while [ $? -eq 0 ] && [ $resp -ge 1 ];do
                case $resp in
                        1)operacion=" + "; echo "Se va a realizar $operacion ";suma $1
                        2)operacion=" - ";echo "Se va a realizar $operacion ";resta $1
                        3)operacion=" * ";echo "Se va a realizar $operacion ";producto
                        4)operacion=" / ";echo "Se va a realizar $operacion ";division
                        5)operacion=" % ";echo "Se va a realizar $operacion ";resto $1
                        *)echo "opción $resp no válida y se termina";break;; #sale
                esac
                echo "$1 $operacion $2 = $resultado"
                read -p "Introduce opción del menú: " resp
        done
else
        echo "Debe introducir 2 números"
fi
************************************************
EJERCICIO  3. Crea un menú para indicar:
        1. Cuántos usuarios están conectados
        2. Usuarios con directorio home y almacenarlos en un fichero llamado all.users
        3. Lista de cuentas de usuarios ordenados por nombre y por id
        4. Dia mes y año mostradas en pantalla con el formato ddmmYY
        5. Cambiar a mayúsculas los nombres de los ficheros pasados por parámetros.

menu(){
        echo "
                ---------------------------------------------------------------------
                1.CUENTA USUARIOS CONECTADOS
                2.CUENTA USUARIOS CON DIRECTORIO HOME Y LOS ALMACENA EN all.users,
                3.LISTA CUENTAS DE USUARIOS ORDENADOR POR NOMBRE E ID
                4.DIA MES AÑO EN VARIABLE MOSTRADA EN PANTALLA
                5.CAMBIA A MAYÚSCULAS LOS NOMBRES DE FICHEROS PASADOS POR PARÁMETRO
                ---------------------------------------------------------------------
                "
}
uno(){
        echo "El número de usuarios conectados es: ";users | wc -w
}
dos(){
        echo "El número de usuarios con directorio home es: ";ls /home | wc -l
        #almacenamiento de usuarios en allusers
        ls /home > allusers
        echo "comprobación de allusers"
        cat allusers
}

tres(){
        echo "---Cuentas de usuarios ordenadas por nombre:--- "
        cat /etc/passwd | cut -d: -f1,3 | sort -t: -k1
        echo "---Cuentas de usuarios ordenadas por id: "
        cat /etc/passwd | cut -d: -f1,3 | sort -t: -k2 -n
}
cuatro(){
        echo "Dia-més-año: "`date +%d-%m-%Y`
}
cinco(){
        #comprobación de parámetros
        if [ $# -ne 0 ];then
                for i in $*;do
                        if [ -f $i ];then # los ficheros son del directorio de trabajo
                                #traducción del nombre
                                nuevo=`echo "$i" | tr [:lower:] [:upper:]`
                                #sustitución del archivo
                                mv $i $nuevo
                                echo "Comprobación de cambio de nombres de ficheros"
                                ls | grep $nuevo
                        else
                                echo "El parámetro $i no es fichero"
```

```
                        fi
                done
        else
                echo "No ha introducido nombres de ficheros a cambiar el nombre"
        fi
}
#PROGRAMA PRINCIPAL
#llamadas a funciones usando menú y eligiendo opciones de menú
while true;do #repite menú mientras introduce opción válida
        menu
        read -p "Introduce opción del menú anterior: " opc
        case $opc in
                1)clear;uno;;
                2)clear;dos;;
                3)clear;tres;;
                4)clear;cuatro;;
                5)clear;cinco $@;;
                *)echo "Ha introducido una opción no válida";break;;
        esac
done

**********************************************************************
EJERCICIO  4 Crear un menú para:
        1. Visualizar las particiones de todos los discos
        2. Visualizar la memoria libre
        3. Visualizar el espacio ocupado por directorios
        4. Visualiza la versión completa del sistema

#!/bin/bash
menu(){
        echo "
        ------------------------------------------------
        1)Visualización de particiones
        2)Visualización de memoria libre
        3)Visualización de espacio ocupado por directorios
        4)Visualización de versión completa del sistema
        ------------------------------------------------
        "
}
uno(){
        #para las particiones debe ser root
        usuario=`id -u`
        if [ $usuario -eq 0 ];then
                echo "Las particiones del sistema son:"
                fdisk -l
        else
                echo "Debe ser root para conocer las particiones"
        fi
}
dos(){
        echo "La memoria libre es:"
        free -h
}
tres(){
        echo "El espacio ocupado por directorios es:"
        du -a
}
cuatro(){
        echo "La versión completa del sistema es:"
        uname -a
}
#programa principal
while true;do
        menu
        read -p "Introduce una opción del menú anterior: " opc
        case $opc in
                1)clear;uno;;
                2)clear;dos;;
                3)clear;tres;;
```

```
                4)clear;cuatro;;
                *)echo "Debe introducir una opción válida";break;;
        esac
done
```

****

EJERCICIO 5. Crear un menú para:

1. Mostrar la fecha del sistema
2. Mostrar la información sobre qué usuarios han iniciado sesión y qué están haciendo.
3. Mostrar los 10 procesos que consumen más cpu
4. Mostrar los 5 procesos que consumen más memoria
5. Mostrar el estado de la red
6. Salir del menú

```
#!/bin/bash
menu(){
        echo "
        ------------------------------------------------
        1)MOSTRAR FECHA
        2)MOSTRAR USUARIOS CONECTADOS Y QUÉ HACEN
        3)MOSTRAR 10 PROCESOS QUE CONSUMEN MÁS MEMORIA
        4)MOSTRAR 10 PROCESOS QUE CONSUMEN MÁS CPU
        5)MOSTRAR EL ESTADO DE LA RED
        6)SALIR
        ------------------------------------------------
        "
}
uno(){
        echo "LA FECHA DEL SISTEMA ES: "`date +%d-%m-%Y`
}
dos(){
        echo "LOS USUARIOS CONECTADOS Y LO QUE HACEN SON:"; w
}
tres(){
        echo "LOS 10 PROCESOS QUE CONSUMEN MÁS MEMORIA"
        ps -auxf | sort -r -k4 | head -10
}
cuatro(){
        echo "LOS 10 PROCESOS QUE CONSUMEN MÁS CPU"
        ps -auxf | sort -r -k3 | head -10
}
cinco(){
        echo "EL ESTADO DE LA RED ES:";netstat -s
}
#PROGRAMA PRINCIPAL
while true;do
        menu
        read -p "INTRODUCE OPCIÓN DEL MENÚ ANTERIOR: " opc
        case $opc in
                1)clear;uno;;
                2)clear;dos;;
                3)clear;tres;;
                4)clear;cuatro;;
                5)clear;cinco;;
                6)exit;;
                *)echo "OPCIÓN NO VÁLIDA";break;;
        esac
done
```

****

EJERCICIO 6 Menú con 4 opciones que actúe sobre un fichero que se le pase como argumento.

1. Buscar el fichero y mostrar su camino absoluto.
2. Cambiar los permisos al fichero.
3. Buscar una cadena (que se solicita) en el fichero.
4. Salir

```
#!/bin/bash
menu(){
        echo "
```

```
        ----------------------------------------
        1)BUSCAR FICHERO Y MOSTRAR CAMINO ABSOLUTO
        2)CAMBIAR PERMISOS AL FICHERO
        3)BUSCAR CADENA EN FICHERO
        4)SALIR
        ----------------------------------------
        "
}
uno(){
        #supongo que está en directorio actual o subdirectorios
        du -a | grep -w "$@" 1>/dev/null
        if [ $? -eq 0 ]; thenecho "EL FICHERO SE HA ENCONTRADO "
                du -a | grep -w "$@" #supongo que está en directorio actual o
subdirectorios
                echo "Y SU CAMINO ABSOLUTO ES: "`readlink -f $@`
        else
                echo "El fichero $@ no se ha encontrado"
        fi
}
dos(){
        echo "LOS PERMISOS ACTUALES DEL FICHERO $@ son "
        ls -lR | grep -w "$@"
        echo "INDICA LOS PERMISOS QUE QUIERES CAMBIAR EN EN VALOR OCTAL"
        read -p "Permiso de usuario " perus
        read -p "Permiso de grupo " pergr
        read -p "Permiso de otros " perot
        chmod $perus$pergr$perot $@
        echo "VERIFICACIÓN DE CAMBIO DE PERMISOS"
        ls -lR | grep -w "$@"
}
tres(){
        read -p "INTRODUCE LA CADENA A BUSCAR EN EL FICHERO " cad
        grep -w "$cad" $@ 1>/dev/null
        if [ $? -eq 0 ];then
                echo "LA CADENA $cad SE HA ENCONTRADO en $@"
        else
                echo "LA CADENA $cad NO SE HA ENCONTRADO EN $@ "
        fi
}
#PROGRAMA PRINCIPAL
#comprobación de parámetro
if [ $# -eq 1 ];then
        if [ -f $@ ];then #sólo funciona para ficheros del directorio actual
                while true;do
                        menu
                        read -p "Introduce un opción del menú " opc
                        case $opc in
                                1)clear;uno $@;;
                                2)clear;dos $@;;
                                3)clear;tres $@;;
                                4)exit;;
                                *)echo "Ha introducido un opción no válida";break
                        esac
                done
        else
                echo "El parámetro $@ no es fichero"
        fi
else
        echo "Debe introducir un parámetro"
fi

*********************************************************************************
EJERCICIO 7 : MENÚ REDES: 1) Hacer ping 2) Consultar IP 3) Añadir ip 4) Cambiar máscara
0) Salir
menu(){
  echo "-------------------------------------
  1) Hacer ping
  2) Consultar IP
  3) Añadir IP
```

```
  4) Cambiar máscara
  0) Salir
  ----------------------------------------
  "
}
uno(){
  clear;echo "Hacer ping"
  read -p "Introduce la ip a la que quieres conectar " direc
  ping -c3 $direc
}

dos(){
  clear; echo "Consultar IP"
  read -p "Introduce la ip a la que quieres consultar " direc
  encontrado=`ip address show | grep $direc`
  if [ "$encontrado" != "" ];then
    echo $encontrado
  else
    echo "$direc no se ha encontrado"
  fi
}
tres(){
  clear;echo "Añadir IP"
  #supongo que no existe
  #faltan las comprobaciones en una función externa
  #si se desea cambiar habría que eliminar y después añadir con los mismos datos menos el
que se desea modificar
  read -p "Introduce la interfaz " disp
  read -p "Introduce la ip a la que quieres añadir " direc
  read -p "Introduce la máscara para la ip en formato abreviado sin /" mas
  read -p "Introduce el broadcast  " bro
  sudo ip address add $direc/$mas broadcast $bro dev $disp
}

cuatro(){
  clear; echo "Cambiar máscara"
  read -p "Introduce la interfaz a modificar " disp #dispositivo
  read -p "Introduce la ip a modificar" direc
  read -p "Introduce máscara anterior en formato abreviado /nº " mas1
  sudo ip address del $direc/$mas1 dev $disp
  read -p "Introduce nueva máscara en formato abreviado /nº " mas2
  read -p "Introduce el nuevo broadcast " bro
  sudo ip address add $direc/$mas2  broadcast $bro dev $disp
}


while true;do
  menu
  read -p "Elige opción del menú anterior " resp
  case $resp in
    1) uno;;
    2) dos;;
    3) tres;;
    4) cuatro;;
    0) exit;;
    *)echo "$resp no es opción válida";exit;;
  esac
done
```