

EXAMEN LINUX CARLA GARCÍA PARRA

EJERCICIO MENÚ

1.

```
function opcion1(){
echo Introduce la ruta donde se encuentra el fichero
#cogemos y metemos en la variable ruta la info del usuario
read ruta
echo =====
echo "Apellidos                               Nombre"
echo -----
#utilizamos cut para que salgan solo apellidos y nombre y --output-delimiter para poner espacios entre
#los dos
cat $ruta | cut -d":" -f3,4 --output-delimiter=" "
echo =====
}
```

```
1
=====
Introduce la ruta donde se encuentra el fichero
/home/alumno/Usuarios.txt
=====
Apellidos                               Nombre
-----
BLASCO BURGUEÑO                        DAVID
CIRUELOS GARCIA                        SERGIO
CORDÓN CARMONA                         SANTIAGO
ESPINOSA MARTÍNEZ                      FRANCISCO JAVIER
ESTEVEZ FERNANDEZ                      ALVARO
FERNANDEZ GARCIA                        RUBEN
FLORES MACIAS                          JUAN JOSE
GARCÍA MANDILLO                        DANIEL
GIL JIMÉNEZ                           VICTOR
GONZALEZ CANSECO                       ALVARO
GONZÁLEZ GARCÍA                        RAÚL
JIMENEZ MONARD                         ANGELICA LICETH
LEÓN CÁMARA                            IRENE
LOPEZ REDONDO                          MIGUEL ANGEL
LÓPEZ GONZÁLEZ                         RUBÉN
LOZANO CLEMARES                        LUIS CARLOS
```

2.

```
function opcion2(){
#con el comando groupadd añadimos el usuario (en este caso usuario ASO)
#se tiene que ejecutar en sudo porque si no, nos da permisos denegados
echo "Introduce un grupo"
read grupo
groupadd -f $grupo
}
```

```
sambashare:x:132:alumno
systemd-coredump:x:999:
vboxsf:x:998:
ASO:x:1001:
```

3

```
function opcion3(){  
#se necesita estar en el terminal como sudo para que funcione  
#creo un fichero solo con los nombres para crear los usuarios  
cat Usuarios.txt | cut -d":" -f2 > usuario  
#creamos los usuarios con bucle while leyendo linea a linea para crearlos  
while IFS= read -r line  
do  
    useradd $line -d /home  
done < usuario  
}
```

```
3  
=====  
useradd: el usuario «dblasco» ya existe  
useradd: el usuario «sciruelos» ya existe  
useradd: el usuario «scordon» ya existe  
useradd: el usuario «fjespinosa» ya existe  
useradd: el usuario «aestevez» ya existe  
useradd: el usuario «rfernandez» ya existe  
useradd: el usuario «jjflores» ya existe  
useradd: el usuario «dgarcia» ya existe  
useradd: el usuario «vgil» ya existe  
useradd: el usuario «agonzalez» ya existe  
useradd: el usuario «rgonzalez» ya existe  
useradd: el usuario «ajimenez» ya existe  
useradd: el usuario «ileon» ya existe  
useradd: el usuario «malopez» ya existe  
useradd: el usuario «rlopez» ya existe  
useradd: el usuario «lclozano» ya existe  
useradd: el usuario «jmmartin» ya existe  
useradd: el usuario «mmendoza» ya existe  
useradd: el usuario «amenendez» ya existe  
useradd: el usuario «mcmoral» ya existe  
useradd: el usuario «aosuna» ya existe  
useradd: el usuario «dpardo» ya existe
```

4.

no sabia hacerlo

5.

```
function opcion5(){
systeminfo
}
```

5

```
=====
Arquitectura:                x86_64
Modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes:          Little Endian
Address sizes:                39 bits physical, 48 bits virtual
CPU(s):                       1
Lista de la(s) CPU(s) en línea: 0
Hilo(s) de procesamiento por núcleo: 1
Núcleo(s) por «socket»:      1
«Socket(s)»:                  1
Modo(s) NUMA:                 1
ID de fabricante:             GenuineIntel
Familia de CPU:                6
Modelo:                        165
Nombre del modelo:             Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz
Revisión:                      3
CPU MHz:                       2903.998
BogoMIPS:                      5807.99
Fabricante del hipervisor:     KVM
Tipo de virtualización:       lleno
```

6.

```
6
=====
introduce un fichero para copiar
Usuarios.txt
introduce un lugar donde copiarlo
/home/alumno/prueba.txt
```

```
drwxr-xr-x 2 alumno alumno 4096 oct  4  2020 Plantillas
-rw-r--r-- 1 root  root  1151 ene 10 10:59 prueba.txt
drwxr-xr-x 2 alumno alumno 4096 oct  4  2020 Público
drwx----- 3 alumno alumno 4096 nov  3 09:35 span
```

7

```
function opcion7(){
exit
}
```

7. Salir

```
=====
7
=====
root@Ubuntu20:/home/alumno#
```

EJERCICIO TABLAS MULTIPLICAR

```
#!/bin/bash
function tabla(){
#pedimos al usuario que tabla es el comienzo del bucle
read -p "Ingresa un número: " respuesta
#bucle de tabla
for h in $(seq $respuesta 10)
do
echo "          Tabla del $h          "
echo =====
for tabla in $(seq 1 10)
do
#creamos una variable que guarde el resultado de dicha operación
let resultado=$h*$tabla
echo "          $h * $tabla = $resultado "
done
echo
sleep 2
done
}
```

```

      Tabla del 8
=====
      8 * 1 = 8
      8 * 2 = 16
      8 * 3 = 24
      8 * 4 = 32
      8 * 5 = 40
      8 * 6 = 48
      8 * 7 = 56
      8 * 8 = 64
      8 * 9 = 72
      8 * 10 = 80

      Tabla del 9
=====
      9 * 1 = 9
      9 * 2 = 18
      9 * 3 = 27
      9 * 4 = 36
      9 * 5 = 45
      9 * 6 = 54
      9 * 7 = 63
      9 * 8 = 72
      9 * 9 = 81

```

Me sale un error al comprobar la variable \$respuesta

```

#lo pongo en comentario porque me sale un error todo el rato al hacer la comprobación
#comprobamos con un if
#if [[ $respuesta -ge 10 ]]
#then echo "introduce un valor correcto"
#elif [[ $respuesta -lt 1 ]]
#then echo "introduce un valor correcto"
#si todo esta bien, llamamos a la función de la tabla
#else tabla
#fi

```