

LABORATORIO 1

1. GESTIÓN Y MANIPULACIÓN DE ARCHIVOS

1) Crear una carpeta llamada AS

```
nagore@instance-1: mkdir AS
```

2) Entrar dentro de la carpeta y comprobar que el directorio coincide con el contenido de la variable de entorno PWD.

```
nagore@instance-1:~/AS$ echo $PWD
```

```
/home/nagore/AS
```

```
nagore@instance-1:~/AS$ pwd
```

```
/home/nagore/AS
```

3) Instalar cal con el comando apt install ncal. Utilizar cal para mostrar un calendario y redirigir la salida a un fichero de texto. Comprobar que ese fichero se crea correctamente y que su contenido es el esperado.

```
nagore@instance-1:~/AS$ cal > calendario.txt
```

```
nagore@instance-1:~/AS$ cat calendario.txt
```

4) Copiar el fichero recién creado al directorio raíz del usuario.

```
nagore@instance-1:~/AS$ cp calendario.txt /home/nagore
```

5) Listar en formato extendido (parámetro -l) los directorios y archivos presentes. Redirigir esa información a un fichero.

```
nagore@instance-1:~$ ls -l > lista.txt
```

6) Listar los 5 ficheros más nuevos de la carpeta /etc (que no sean carpetas).

```
nagore@instance-1:~$ ls /etc -ltr | grep -v '^d' | tail -n 5
```

- -l: formato largo
- -t: Ordena por fecha de modificacion (mas viejos a mas recientes)
- -r: Inverso
- -v: muestra las lineas que no coinciden
- ^d: comienzo de linea por letra d, de directorio
- -n 5: ultimas 5 lineas

7) Cambiar los permisos del fichero creado con información del calendario para que sólo el usuario propietario tenga capacidad de leer y escribir.

```
nagore@instance-1:~$ ls -l calendario.txt
```

```
-rw-rw-r-- 1 nagore nagore 184 Sep 13 17:34 calendario.txt
```

```
nagore@instance-1:~$ chmod g-r-w calendario.txt
```

```
nagore@instance-1:~$ chmod o-r calendario.txt
```

8) Cambiar permisos para evitar que ningún otro usuario pueda acceder nuestro directorio raíz.

```
nagore@instance-1:/home$ chmod g-r-x /home/nagore
```

9) Comprobar cuántos usuarios hay en el sistema y cual es el nuestro Shell de inicio.

```
nagore@instance-1:~$ wc -l /etc/passwd
```

```
nagore@instance-1:~$ echo $SHELL
```

10) Comprobar cuándo y desde donde accediste la última vez al sistema.

```
nagore@instance-1:~$ last -1
```

11) Comprimir en un archivo .tar.gz los contenidos del directorio \$HOME. Descomprimirlos /tmp y comprobar que se ha hecho correctamente.

```
nagore@instance-1:~$ tar -czvf backup_home.tar.gz $HOME
```

- -c: Crea un nuevo archivo.
- -z: Utiliza la compresión gzip.
- -v: Muestra detalles sobre el proceso de compresión.
- -f: Especifica el nombre del archivo de salida (backup_home.tar.gz en este caso).

```
nagore@instance-1:~$ tar -xzvf backup_home.tar.gz -C /tmp
```

- -x: Extrae el contenido del archivo.
- -z: Utiliza la compresión gzip.
- -v: Muestra detalles sobre el proceso de extracción.
- -f: Especifica el nombre del archivo de entrada (backup_home.tar.gz en este caso).
- -C: Especifica el directorio de destino (/tmp en este caso).

```
nagore@instance-1:~$ ls /tmp
```

12) Como usuario "root", buscar todos los archivos que sean propiedad de tu usuario en el sistema (desde el directorio raíz /) y listarlos en forma extendida (con toda la información).

```
nagore@instance-1:~$ sudo find / -user nagore -exec ls -l {} \;
```

- `-exec ls -l {} \;`: Esto ejecuta el comando `ls -l` en cada archivo que cumple con los criterios de búsqueda. `{}` representa el archivo encontrado y `\;` indica el final del comando `-exec`.

13) Como usuario "root", mostrar las últimas 30 líneas de /var/log/syslog.

```
nagore@instance-1:~$ sudo tail -n 30 /var/log/syslog
```

2. SHELL SCRIPTING

1) Crear un script llamado `lsdirs.sh` que muestre los directorios contenidos en el directorio actual.

```
#!/bin/bash
for dir in */; do
    [ -d "$dir" ] && echo "$dir"
done
```

2) Crear un script llamado `see.sh` que reciba un nombre de fichero/directorio como parámetro. Si el nombre corresponde a un fichero, el script muestra el contenido del fichero con `more`, sino muestra el contenido del directorio con `ls`.

```
#!/bin/bash

if [ -d "$1" ]; then
    ls -l "$1"
elif [ -f "$1" ]; then
    more "$1"
else
    echo "El argumento no es ni un archivo ni un directorio"
```

fi

3) Crear un script que pida al usuario que teclee una palabra y escriba por pantalla el número de caracteres de esa palabra.

```
#!/bin/bash
echo "Teclea una palabra:"
read palabra
caracteres=$(echo -n "$palabra" | wc -m)
echo "La palabra '$palabra' tiene $caracteres caracteres"
```

4) Crear un script que pida al usuario que teclee una palabra y compruebe si es un comando del sistema o no.

```
#!/bin/bash
echo "Teclea una palabra:"
read palabra
if command -v "$palabra" > /dev/null 2>&1; then
    echo "'$palabra' es un comando del sistema"
else
    echo "'$palabra' no es un comando del sistema"
fi
```

5) Crear un script que cree una carpeta llamada cosas y después cree 100 ficheros vacíos llamados fich<numero>.txt dentro de ella, donde <numero> es un número entre 0 y 99.

6) Extender el script anterior para que cada fichero contenga la N-sima línea del manual de ls (man ls). El fichero fich0.txt tendrá la línea 0 del manual, fich1.txt tendrá la línea 1 del manual, ...

```
#!/bin/bash
carpeta="cosas"
#crear la carpeta si no existe
if [ ! -d "$carpeta" ]; then
    mkdir "$carpeta"
```

```

fi
#cambiar al directorio cosas
cd "$carpeta" || exit
for i in {1..100}; do
    # obtener la nsima linea del manual y guardarla en el nsimo archivo
    man ls | sed -n "${i}p" > "fich${i}.txt"
done

```

7) Crear un script que modifique la extensión de todos los ficheros .txt de un directorio a .t.

```

#!/bin/bash
echo "Inserta un directorio:"
read direc
if [ ! -d "$direc" ]; then
    echo "El directorio '$direc' no existe"
    exit 1
fi
for archivo in "$direc"/*.txt; do
    if [ -f "$archivo" ]; then
        nuevo="${archivo%.txt}.t"
        mv "$archivo" "$nuevo"
        echo "Se cambio de '$archivo' a '$nuevo'"
    fi
done

```

8) Crear un script borra.sh que reciba un número indefinido de parámetros (de 0 a 9) y borre el fichero correspondiente a la suma del valor de los parámetros que reciba. Por ejemplo, borra.sh 1 4 5 9, borraría el fichero fich19.txt (1 + 4 + 5 + 9 = 19).

```

#!/bin/bash
# Hacer la suma de los numeros
suma=0

```

```

for num in "$@"; do
    if [[ "$num" =~ ^[0-9]+$ ]]; then
        suma=$((suma + num))
    else
        echo "Error: '$num' no es un número válido."
        exit 1
    fi
done

# Nombre del archivo a borrar
archivo="fich$suma.txt"

#cambiar al directorio cosas
cd cosas || exit

# Verificar si el archivo existe antes de borrarlo
if [ -e "$archivo" ]; then
    rm "$archivo"

    echo "Se ha borrado el archivo '$archivo'."
else
    echo "El archivo '$archivo' no existe."
fi

```

9) Crear un script orden.sh que muestre el contenido del fichero /etc/passwd ordenado por nombre de usuario, UID o GID. El script recibirá un parámetro que indique cuál de los 3 utilizar como clave de ordenación.

```

#!/bin/bash

case "$1" in
    "nombre")
        sort -t: -k1,1 /etc/passwd
        ;;
    "UID")
        sort -t: -k3,3n /etc/passwd

```

```

;;
"GID")
    sort -t: -k4,4n /etc/passwd
;;
*)
    echo "Clave de orden no válida. Las claves de orden válidas son: nombre, UID, GID"
    exit 1
;;
esac

```

10) Imagina que quieres mandar un e-mail con el mismo cuerpo a varios destinatarios, pero personalizando la primera línea con el nombre de cada uno. Crear un fichero cuerpo.txt con el texto del cuerpo del e-mail y la palabra NOMBRE en cada lugar donde querrías poner el nombre, y un fichero nombres.txt con varios nombres personales. Crear un script que genere varios ficheros con el cuerpo del e-mail personalizado para cada destinatario.

```

#!/bin/bash

# Leer el contenido del cuerpo del correo
cuerpo=$(<cuerpo.txt)

# Iterar a través de la lista de nombres y generar archivos personalizados
while read -r nombre; do
    # Reemplazar "NOMBRE" con el nombre actual
    cuerpo_personalizado="${cuerpo//NOMBRE/$nombre}"

    # Crear un archivo con el cuerpo del correo personalizado
    echo "$cuerpo_personalizado" > "correo_${nombre}.txt"
    echo "Correo personalizado para $nombre creado."
done < nombres.txt

exit 0

```

Exam

```
#!/bin/bash
```

```
echo "Introduce un numero entre 1 y 10"
```

```
read numero
```

```
mkdir $HOME/ficheros
```

```
cd ficheros
```

```
for((i=1; i<=$numero; i++)); do
```

```
    echo $i
```

```
    if [ $numero -le 5 ];then
```

```
        echo "coche">fich_$i.txt
```

```
    else
```

```
        echo "casa">fich_$i.txt
```

```
    fi
```

```
done
```