

LABORATORIO 1: Administración Linux y Shell Scripting

Gestión y manipulación de archivos

Esta parte está enfocada a explorar diferentes comandos del Shell y ver su efecto.

1. Crear una carpeta llamada AS en vuestro directorio raíz de usuario.
2. Entrar dentro de la carpeta y comprobar que el directorio coincide con el contenido de la variable de entorno PWD.
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.

```
maitane@as2-maitane:~/lab1$ cal
  September 2023
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

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

```
maitane@as2-maitane:~/lab1$ cal > calendario.txt

maitane@as2-maitane:~/lab1$ cat calendario.txt
  September 2023
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

5. Moverse al directorio raíz del usuario y listar en formato extendido (parámetro `-l`) los directorios y archivos presentes. Redirigir esa información a un fichero.

```
maitane@as2-maitane:~/lab1$ ls -l > todo.txt

maitane@as2-maitane:~/lab1$ cat todo.txt
total 52
-rwxrw-r-- 1 maitane maitane 237 Sep 14 10:15 10email.sh
-rwxrw-r-- 1 maitane maitane  59 Sep 13 21:41 11sdirs.sh
```

```
-rwxrw-r-- 1 maitane maitane 102 Sep 13 22:06 2see.sh
-rwxrw-r-- 1 maitane maitane 112 Sep 13 22:10 3largopalabra.sh
-rwxrw-r-- 1 maitane maitane 204 Sep 13 22:20 4escomando.sh
-rwxrw-r-- 1 maitane maitane 116 Sep 13 22:30 5carpetaFicheros.sh
-rwxrw-r-- 1 maitane maitane 183 Sep 14 08:56 6nLineaManualLs.sh
-rwxrw-r-- 1 maitane maitane 235 Sep 14 09:17 7txtat.sh
-rwxrw-r-- 1 maitane maitane 81 Sep 14 09:25 8borra.sh
-rwxrw-r-- 1 maitane maitane 378 Sep 14 09:39 9orden.sh
-rw-rw-r-- 1 maitane maitane 188 Sep 26 20:50 calendario.txt
-rw-rw-r-- 1 maitane maitane 85 Sep 14 10:17 cuerpo.txt
-rw-rw-r-- 1 maitane maitane 33 Sep 14 10:18 nombres.txt
-rw-rw-r-- 1 maitane maitane 0 Sep 26 20:51 todo.txt
```

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

```
maitane@as2-maitane:~/lab1$ ls -ltr /etc | grep -v '^d' | tail -n 5
-rw-r--r-- 1 root root      1951 Sep 12 10:10 passwd
-rw-r----- 1 root shadow    1013 Sep 12 10:10 shadow
-rw-r--r-- 1 root root       84 Sep 20 16:36 fstab.backup
-rw-r--r-- 1 root root       84 Sep 20 17:35 fstab
-rw-r--r-- 1 root root    24229 Sep 26 13:57 ld.so.cache
```

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

```
maitane@as2-maitane:~/lab1$ chmod 700 calendario.txt

maitane@as2-maitane:~/lab1$ chmod u-x calendario.txt

maitane@as2-maitane:~/lab1$ ls -l
total 56
-rwxrw-r-- 1 maitane maitane 237 Sep 14 10:15 10email.sh
-rwxrw-r-- 1 maitane maitane 59 Sep 13 21:41 1lsdirs.sh
-rwxrw-r-- 1 maitane maitane 102 Sep 13 22:06 2see.sh
-rwxrw-r-- 1 maitane maitane 112 Sep 13 22:10 3largopalabra.sh
-rwxrw-r-- 1 maitane maitane 204 Sep 13 22:20 4escomando.sh
-rwxrw-r-- 1 maitane maitane 116 Sep 13 22:30 5carpetaFicheros.sh
-rwxrw-r-- 1 maitane maitane 183 Sep 14 08:56 6nLineaManualLs.sh
-rwxrw-r-- 1 maitane maitane 235 Sep 14 09:17 7txtat.sh
-rwxrw-r-- 1 maitane maitane 81 Sep 14 09:25 8borra.sh
-rwxrw-r-- 1 maitane maitane 378 Sep 14 09:39 9orden.sh
-rw----- 1 maitane maitane 188 Sep 26 20:50 calendario.txt
-rw-rw-r-- 1 maitane maitane 85 Sep 14 10:17 cuerpo.txt
-rw-rw-r-- 1 maitane maitane 33 Sep 14 10:18 nombres.txt
-rw-rw-r-- 1 maitane maitane 830 Sep 26 20:51 todo.txt
```

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

```
maitane@as2-maitane:~/lab1$ ls -l /home
total 8
drwxr-xr-x 10 maitane maitane 4096 Sep 26 14:42 maitane
drwxr-xr-x  3 ubuntu  ubuntu  4096 Sep 12 10:10 ubuntu

maitane@as2-maitane:~/lab1$ chmod o-x /home/maitane

maitane@as2-maitane:~/lab1$ ls -l /home
total 8
drwxr-xr-- 10 maitane maitane 4096 Sep 26 14:42 maitane
drwxr-xr-x  3 ubuntu  ubuntu  4096 Sep 12 10:10 ubuntu
```

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

```
maitane@as2-maitane:~/lab1$ wc -l /etc/passwd
36 /etc/passwd

maitane@as2-maitane:~/lab1$ echo $SHELL
/bin/bash
```

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

```
maitane@as2-maitane:~/lab1$ last | head -n 1
maitane pts/0      88.14.240.110    Tue Sep 26 20:44    still logged in
```

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

```
maitane@as2-maitane:~/lab1$ tar -czvf /tmp/nuevoHome.tar.gz $HOME
tar: Removing leading `/' from member names
/home/maitane/
/home/maitane/.ssh/
/home/maitane/.ssh/authorized_keys
/home/maitane/ubuntu-22.04.3-live-server-amd64.iso

maitane@as2-maitane:~/lab1$ ls /tmp/
nuevoHome.tar.gz

maitane@as2-maitane:~/lab1$ tar -xzvf /tmp/nuevoHome.tar.gz -C /tmp/
home/maitane/
home/maitane/.ssh/
home/maitane/.ssh/authorized_keys
home/maitane/ubuntu-22.04.3-live-server-amd64.iso

maitane@as2-maitane:~/lab1$ ls /tmp/
```

```
home nuevoHome.tar.gz
```

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).

La última parte del comando `-exec ls -l {} \;` se utiliza en combinación con el comando `find` para realizar una acción específica en cada archivo encontrado por `find`.

- `-exec`: le indica a `find` que ejecute un comando en cada archivo o directorio que coincida con los criterios de búsqueda.
- `ls -l`: El comando que se ejecuta en cada archivo.
- `{}`: marcador que `find` reemplaza con la ruta completa de cada archivo antes de ejecutar el comando.
- `\;`: Esta parte indica el final del comando `-exec` para que `find` sepa dónde termina.

```
maitane@as2-maitane:~/lab1$ sudo find / -user root -exec ls -l {} \;
total 64
lrwxrwxrwx 1 root root 7 Sep 7 21:40 bin -> usr/bin
drwxr-xr-x 4 root root 4096 Sep 26 13:57 boot
drwxr-xr-x 16 root root 4000 Sep 26 20:44 dev
drwxr-xr-x 97 root root 4096 Sep 26 20:44 etc
drwxr-xr-x 4 root root 4096 Sep 12 10:10 home
lrwxrwxrwx 1 root root 7 Sep 7 21:40 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Sep 7 21:40 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Sep 7 21:40 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Sep 7 21:40 libx32 -> usr/libx32
drwx----- 2 root root 16384 Sep 7 21:44 lost+found
drwxr-xr-x 2 root root 4096 Sep 7 21:40 media
drwxr-xr-x 3 root root 4096 Sep 20 16:52 mnt
drwxr-xr-x 2 root root 4096 Sep 7 21:40 opt
dr-xr-xr-x 167 root root 0 Sep 26 20:44 proc
drwx----- 5 root root 4096 Sep 26 14:30 root
drwxr-xr-x 25 root root 840 Sep 26 20:44 run
lrwxrwxrwx 1 root root 8 Sep 7 21:40 sbin -> usr/sbin
drwxr-xr-x 12 root root 4096 Sep 13 17:41 snap
drwxr-xr-x 2 root root 4096 Sep 7 21:40 srv
dr-xr-xr-x 13 root root 0 Sep 26 20:44 sys
drwxrwxrwt 12 root root 4096 Sep 26 20:59 tmp
drwxr-xr-x 14 root root 4096 Sep 7 21:42 usr
drwxr-xr-x 13 root root 4096 Sep 7 21:43 var
lrwxrwxrwx 1 root root 7 Sep 7 21:40 /bin -> usr/bin
total 0
lrwxrwxrwx 1 root root 8 Sep 7 21:40 /sbin -> usr/sbin
```

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

```
maitane@as2-maitane:~/lab1$ tail -n 30 /var/log/syslog
Sep 26 20:44:51 as2-maitane systemd[1078]: Listening on GnuPG cryptographic agent
```

```
(ssh-agent emulation).
Sep 26 20:44:51 as2-maitane systemd[1078]: Listening on GnuPG cryptographic agent
and passphrase cache.
Sep 26 20:44:51 as2-maitane systemd[1078]: Listening on debconf communication
socket.
Sep 26 20:44:51 as2-maitane systemd[1078]: Listening on REST API socket for snapd
user session agent.
Sep 26 20:44:51 as2-maitane systemd[1078]: Listening on D-Bus User Message Bus
Socket.
Sep 26 20:44:51 as2-maitane systemd[1078]: Reached target Sockets.
Sep 26 20:44:51 as2-maitane systemd[1078]: Reached target Basic System.
Sep 26 20:44:51 as2-maitane systemd[1078]: Reached target Main User Target.
Sep 26 20:44:51 as2-maitane systemd[1078]: Startup finished in 108ms.
Sep 26 20:44:51 as2-maitane systemd[1]: Started User Manager for UID 1001.
Sep 26 20:44:51 as2-maitane systemd[1]: Started Session 1 of user maitane.
Sep 26 20:45:00 as2-maitane systemd[1]: systemd-fsckd.service: Succeeded.
Sep 26 20:45:01 as2-maitane CRON[1186]: (root) CMD (command -v debian-sa1 >
/dev/null && debian-sa1 1 1)
Sep 26 20:45:08 as2-maitane systemd[1]: systemd-hostnamed.service: Succeeded.
Sep 26 20:45:12 as2-maitane systemd[1]: systemd-timedated.service: Succeeded.
Sep 26 20:54:42 as2-maitane systemd[1]: Starting GCE Workload Certificate
refresh...
Sep 26 20:54:42 as2-maitane gce_workload_cert_refresh[1233]: 2023/09/26 20:54:42:
Error getting config status, workload certificates may not be configured: HTTP 404
Sep 26 20:54:42 as2-maitane gce_workload_cert_refresh[1233]: 2023/09/26 20:54:42:
Done
Sep 26 20:54:42 as2-maitane systemd[1]: gce-workload-cert-refresh.service:
Succeeded.
Sep 26 20:54:42 as2-maitane systemd[1]: Finished GCE Workload Certificate refresh.
Sep 26 20:55:01 as2-maitane CRON[1246]: (root) CMD (command -v debian-sa1 >
/dev/null && debian-sa1 1 1)
Sep 26 20:59:42 as2-maitane systemd[1]: Starting Cleanup of Temporary
Directories...
Sep 26 20:59:42 as2-maitane systemd[1]: systemd-tmpfiles-clean.service: Succeeded.
Sep 26 20:59:42 as2-maitane systemd[1]: Finished Cleanup of Temporary Directories.
Sep 26 21:04:42 as2-maitane systemd[1]: Starting GCE Workload Certificate
refresh...
Sep 26 21:04:42 as2-maitane gce_workload_cert_refresh[7530]: 2023/09/26 21:04:42:
Error getting config status, workload certificates may not be configured: HTTP 404
Sep 26 21:04:42 as2-maitane gce_workload_cert_refresh[7530]: 2023/09/26 21:04:42:
Done
Sep 26 21:04:42 as2-maitane systemd[1]: gce-workload-cert-refresh.service:
Succeeded.
Sep 26 21:04:42 as2-maitane systemd[1]: Finished GCE Workload Certificate refresh.
Sep 26 21:05:01 as2-maitane CRON[7540]: (root) CMD (command -v debian-sa1 >
/dev/null && debian-sa1 1 1)
```

Shell Scripting

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

```
#!/bin/bash
for archivo in `ls`; do
    echo $archivo;
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 $1
elif [ -f $1 ]; then
    more $1
else
    echo "prueba de nuevo"
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 "Mete una palabra:"
read palabra
echo "La palabra $palabra tiene una longitud de ${#palabra}"
```

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 "Introduce 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.txt dentro de ella, donde es un número entre 0 y 99.

```
#!/bin/bash
if [ ! -d cosas ]; then
```

```

        mkdir cosas
    fi
    cd cosas || exit
    for i in {0..99}; do
        touch "fich$i.txt"
    done

```

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
if [ ! -d cosas2 ]; then
    mkdir cosas2
fi

i=0

cd cosas2 || exit

man ls | while IFS= read -r linea; do
    touch "fich$i.txt"
    echo $linea > "fich$i.txt"
    ((i++))
done

```

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

```

#!/bin/bash
#for fichero in *.txt; do
#    nombre=$(cut -d "." -f 1 $fichero)
#    mv $fichero "$nombre.t"
#done

for fichero in *.txt; do
    if [ -f "$fichero" ]; then
        nombre="${fichero%.txt}"
        mv "$fichero" "${nombre}.t"
    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
suma=0
for i in "$@"; do

```

```

        suma=$(( $suma+$i ))
done
rm "fich$suma.txt"

```

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
echo "Qué clave de ordenación deseas usar: nombre de usuario (1), UID (2) o GID (3):"
read clave

case $clave in
    1)
        echo "Con usuario"
        sort -t ":" -k1 /etc/passwd
        ;;
    2)
        echo "Con UID"
        sort -t ":" -k3n /etc/passwd
        ;;
    3)
        echo "Con GID"
        sort -t ":" -k4n /etc/passwd
        ;;
    *)
        echo "Opción no válida, prueba de nuevo (elige 1, 2 o 3)"
        ;;
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 querías 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
reemplazar="NOMBRE"
if [[ -f "nombres.txt" && -f "cuerpo.txt" ]]; then
    while read -r nombre; do
        echo "PRIMER EMAIL PARA: $nombre"
        sed "s/$reemplazar/$nombre/g" "cuerpo.txt" > "carta_$nombre.txt"
    done < "nombres.txt"
fi

```