

---

## SISTEMAS OPERATIVOS I

### Actividades - Práctica 1 (con soluciones)

---

## 1. Actividades de Laboratorio

1. Estando situado en el directorio <HOME> utilice el comando `cat` entre otros para realizar las siguientes tareas:

- a) Mostrar el contenido del archivo `/etc/passwd` en la salida estándar
- b) Copiar el contenido del archivo `/etc/passwd` en el archivo `/tmp/usuarios`
- c) Situar en el directorio `/tmp` y copiar el contenido del archivo `/etc/passwd` en un archivo con nombre `usuarios` del directorio <HOME> del usuario actual
- d) Añadir al final del archivo creado en el punto anterior el contenido del archivo `/etc/group`
- e) Concatenar el contenido de los archivos `/etc/shells` y `/etc/services` en el archivo `varios` del directorio actual
- f) Realizar una copia de las líneas recibidas por la entrada estándar en la salida estándar

- a) `cat /etc/passwd`
- b) `cat /etc/passwd > /tmp/usuarios`
- c) `cd /tmp`  
`cat /etc/passwd > ~/usuarios`
- d) `cat /etc/group >> ~/usuarios`
- e) `cat /etc/shells /etc/services > varios`
- f) `cat`

2. Crear la estructura de directorio que se indica en la figura 1 en el directorio *home*. Los diferentes archivos son archivos de texto con una línea de texto con la frase "Este es el archivo <nombre-archivo>". De esa forma se puede reconocer el nombre del archivo original por su contenido.

```
cd
mkdir -p practicas/backup
mkdir -p practicas/comun
mkdir -p practicas/p1
mkdir -p practicas/p2
mkdir -p practicas/p3

echo 'Este es el archivo comun.c' > practicas/comun/comun.c
echo 'Este es el archivo comun.h' > practicas/comun/comun.h
echo 'Este es el archivo Makefile1' > practicas/p1/Makefile1
echo 'Este es el archivo p1.c' > practicas/p1/p1.c
echo 'Este es el archivo p1.h' > practicas/p1/p1.h
echo 'Este es el archivo Makefile2' > practicas/p2/Makefile2
echo 'Este es el archivo README.txt' > practicas/p2/README.txt
echo 'Este es el archivo p2.c' > practicas/p2/p2.c
echo 'Este es el archivo p2.h' > practicas/p2/p2.h
echo 'Este es el archivo Makefile3' > practicas/p3/Makefile3
echo 'Este es el archivo p3.c' > practicas/p3/p3.c
echo 'Este es el archivo p3.h' > practicas/p3/p3.h
```

3. Situar en el directorio <HOME>.

Cambiarse ahora al subdirectorio <HOME>/practicas/p3. Crear desde aquí un subdirectorio del directorio <HOME>/practicas/comun denominado p2.

```
cd
cd practicas/p3
mkdir ~/practicas/comun/p2
```

4. Seguimos situados en <HOME>/practicas/p3.

Copiar el archivo <HOME>/practicas/p1/p1.c con el nombre p1.c en el directorio actual.

```
cp ~/practicas/p1/p1.c .
```

5. Seguimos situados en <HOME>/practicas/p3.

Copiar el directorio <HOME>/practicas/p2 con todo su contenido como un subdirectorio de <HOME>/practicas/p1.

```
cp -r ~/practicas/p2 ~/practicas/p1
```

6. Situar en el directorio <HOME>/practicas/p2.

Mover el archivo <HOME>/practicas/p2/README.txt al directorio <HOME>/practicas/p3 con el nombre texto.txt.

```
cd ~/practicas/p2
mv ~/practicas/p2/README.txt ~/practicas/p3/texto.txt
```

7. Seguimos situados en <HOME>/practicas/p2.

Crear desde aquí un enlace físico al archivo <HOME>/practicas/p3/p3.c que se llame p31.c y esté en el directorio <HOME>/practicas/p1.

Añadir al final del nodo p31.c la frase **Esta es la parte nueva**.

Revisar con un paginador el contenido del archivo <HOME>/practicas/p3/p3.c.

Ahora borrar el archivo <HOME>/practicas/p3/p3.c.

Revisar con un paginador el contenido del archivo <HOME>/practicas/p1/p31.c.

Explique lo que ha sucedido.

```
ln ~/practicas/p3/p3.c ~/practicas/p1/p31.c
echo Esta es la parte nueva >> ~/practicas/p1/p31.c
less ~/practicas/p3/p3.c o more ~/practicas/p3/p3.c
rm ~/practicas/p3/p3.c
less ~/practicas/p1/p31.c o more ~/practicas/p1/p31.c
# Siempre hemos tenido un sólo archivo con 2 enlaces y con el comando rm se eliminan enlaces
# y sólo se elimina un archivo cuando se queda sin ningún enlace ese archivo
```

8. Seguimos situados en <HOME>/practicas/p2.

Crear dos enlaces simbólicos denominados lcomun.c y lcomun.h en el directorio <HOME>/practicas/p1 apuntando respectivamente a los archivos comun.c comun.h del directorio <HOME>/practicas/comun.

Comprobar el resultado mediante ls.

Mediante less explorar el contenido de <HOME>/practicas/p1/lcomun.c.

Borrar ahora <HOME>/practicas/comun/comun.c.

Volver a explorar ahora <HOME>/practicas/p1/lcomun.c.

¿Qué sucede?

```
ln -s ~/practicacomun/comun.c ~/practicasp1/lcomun.c
ln -s ~/practicacomun/comun.h ~/practicasp1/lcomun.h
less ~/practicasp1/lcomun.c
rm ~/practicacomun/comun.c
less ~/practicasp1/lcomun.c
# No se puede leer el archivo lcomun.c pues el único enlace físico del archivo (comun.c) se
# ha eliminado y lcomun.c es sólo un enlace simbólico que apunta al enlace físico
```

9. Seguimos situados en <HOME>/practicasp2.

Obtener el día y la fecha actual y añadirlo al final del archivo <HOME>/practicacomun/comun.h y el archivo fecha del directorio <HOME>

```
date | tee -a ~/practicacomun/comun.h ~/fecha
```

10. Seguimos situados en <HOME>/practicasp2.

Obtener ahora el número de líneas del listado en formato largo, recursivo e incluyendo los nodos ocultos del directorio <HOME>/practicacomun.

```
ls -alR ~/practicacomun | wc -l
```

11. Seguimos situados en <HOME>/practicasp2.

Obtener ahora las tres primeras líneas del archivo <HOME>/practicasp1/lcomun.h y después las tres últimas.

```
head -n3 ~/practicasp1/lcomun.h
tail -n3 ~/practicasp1/lcomun.h
```

12. Obtener el número de palabras del archivo autoconf-2.68-noarch-1 del directorio /var/log/packages. Escribir el número de palabras y la orden del shell utilizada para su cálculo directo.

```
wc -w /var/log/packages/autoconf-2.68-noarch-1
```

13. Supongamos que estamos situados en nuestro directorio *home*, el cual es ahora nuestro directorio de trabajo actual. Crear en el directorio *home* un subdirectorio denominado *work*. Establecer (sin moverse del directorio *home*) un enlace simbólico (blando) denominado *bison.log* en nuestro directorio *work* al archivo *bison-2.4.3-i486-2* del directorio /var/log/packages.

```
cd
mkdir work
ln -s /var/log/packages/bison-2.4.3-i486-2 work/bison.log
```

14. Situémonos en el directorio *home*. Copiar los archivos del directorio /var/log/packages cuyo nombre contenga en el segundo carácter cualquier letra comprendida entre la "a" y la "d" ambas inclusive en nuestro directorio *work*.

```
cp /var/log/packages/[a-d]* work
```

15. Supongamos que ahora estamos situados en nuestro directorio *work*, el cual es ahora nuestro directorio de trabajo actual. Copiar los archivos del directorio */var/log/packages* cuyo nombre no empieza por "a" ni por "z" en nuestro directorio actual.

```
cd ~/work
cp /var/log/packages/[!az]* .
```

16. Situémonos de nuevo en el directorio *home*. Haga ahora una copia recursiva del directorio */var/log/packages* en nuestro directorio *work*.

```
cd
cp -r /var/log/packages work
```

17. Situar en el directorio *<HOME>/practicas/p3*.

Determinar los permisos del archivo *<HOME>/practicas/p1/p2/README.txt*. ¿Qué permisos tiene el administrador del sistema para usar este archivo?

Cambiar los permisos de ese archivo para que todos los usuarios puedan escribir en él, utilizando el modo absoluto. Utilizar el modo simbólico para hacer que cualquier usuario tenga sólo permiso de lectura en el archivo.

```
cd ~/practicas/p3
ls -l ~/practicas/p1/p2/README.txt
# El administrador sólo tiene permiso de lectura
chmod 666 ~/practicas/p1/p2/README.txt
chmod ugo=r ~/practicas/p1/p2/README.txt
```

18. Seguimos situados en el directorio *<HOME>/practicas/p3*.

Cambiar el propietario del directorio *<HOME>/practicas/p1* para que sea el administrador. Retirar los permisos de ejecución al directorio anterior para todos los usuarios menos para el administrador.

Actuando como el usuario *alumno* ver el contenido del archivo *<HOME>/practicas/p1/p1.c*. ¿Qué sucede? (¡Pero si teníamos permiso de lectura en ese archivo...!?).

```
su
chown root /home/alumno/practicas/p1
chmod go-x /home/alumno/practicas/p1
exit
less ~/practicas/p1/p1.c
# No podemos leer el archivo pues no tenemos permiso de ejecución
# en un subdirector de acceso al archivo
```

19. Seguimos situados en el directorio *<HOME>/practicas/p3*.

¿Cuál es el grupo de los usuarios *root* y *alumno*?

Conceder permiso de ejecución en el directorio */home/alumno/practicas/p1* a los usuarios del grupo del nodo.

Actuando como usuario *alumno* ver el contenido del archivo *<HOME>/practicas/p1/p1.c*. ¿Qué ha sucedido ahora?

```

less /etc/passwd
less /etc/group
# El grupo de root es 0 (root) y de alumno es 100 (users)
su
chmod g+x /home/alumno/practicas/p1
exit
less ~/practicas/p1/p1.c
# Se puede acceder pues el grupo del nodo tiene permiso de lectura
# y el usuario alumno pertenece al grupo del nodo

```

- 20.Cuál es el contenido de las variables PATH, HOME y PWD del shell.

Obtener los valores de todas las variables de entorno.

```

echo $PATH
echo $HOME
echo $PWD
env
# Con el comando set se obtiene todas las variables locales del shell
# pero no necesariamente todas son variables del entorno de ejecución

```

21. Seguimos situados en el directorio <HOME>/practicas/p3.

Concatenar los archivos <HOME>/practicas/p2/p2.c y <HOME>/practicas/p1/p31.c de modo que el resultado quede en un archivo denominado concatenado.c situado en el directorio actual.

```
cat ~/practicas/p2/p2.c ~/practicas/p1/p31.c > concatenado.c
```

22. Seguimos situados en el directorio <HOME>/practicas/p3.

Listar recursivamente los nombres de los archivos del directorio <HOME>/practicas/p1 que sean propiedad del usuario *alumno*.

```
find ~/practicas/p1 -type f -user alumno
```

23. Seguimos situados en el directorio <HOME>/practicas/p3.

Obtener un listado en formato largo de los nodos del directorio <HOME>/practicas/p1 recursivamente, que sean propiedad del administrador, y almacenarlo en un archivo denominado listado.txt situado en el directorio actual.

```
find ~/practicas/p1 -user root -ls > listado.txt
```

24. Seguimos situados en el directorio <HOME>/practicas/p3.

Copiar los archivos del directorio <HOME>/practicas/p2 y sus descendientes, que tengan permisos *rw-rw-r--* y con una extensión ".h", en el directorio actual.

```
find ~/practicas/p2 -type f -perm 664 -name '*.h' -exec cp {} . \;
```

25. Seguimos situados en el directorio <HOME>/practicas/p3.

Buscar recursivamente en el directorio <HOME>/practicas los archivos que contengan en su interior la cadena "archivo p2", imprimiendo el número de línea en la que aparece.

```
grep -rn 'archivo p2' ~/practicass 2> /dev/null
```

26. Cambiarnos al directorio <HOME>. Construir un archivo serializado y comprimido con el contenido completo del directorio <HOME>/practicass y guardarlo con el nombre `practicass.tgz` en el directorio <HOME>.

Ahora mover el archivo `practicass.tgz` al directorio <HOME>/practicass/p3, cambiarse a ese directorio y descomprimir el archivo `practicass.tgz` en él.

Controlar lo que ha sucedido mediante la orden `ls`.

```
tar cvfz practicass.tgz practicass
mv practicass.tgz practicass/p3
cd practicass/p3
tar xvfz practicass.tgz
```

27. Crear un archivo de texto con el siguiente contenido

```
125 Gutiérrez Martínez José 1785
350 Gutiérrez Prado Enrique 6780
224 Ansúrez Gómez Alberto 234
```

en el directorio <HOME>/practicass/p2 con el nombre de `tabla.txt`.

Ordenar en orden inverso el contenido del archivo `tabla.txt` por el primer apellido (como clave primaria) y por el segundo apellido (como clave secundaria).

```
echo 125 Gutiérrez Martínez José 1785 > ~/practicass/p2/tabla.txt
echo 350 Gutiérrez Prado Enrique 6780 >> ~/practicass/p2/tabla.txt
echo 224 Ansúrez Gómez Alberto 234 >> ~/practicass/p2/tabla.txt
sort -r -k 2,3 ~/practicass/p2/tabla.txt
```

28. Obtener el número de bytes y el espacio de disco ocupado por el archivo `/etc/passwd`.

```
du -b /etc/passwd # 1049 bytes
du -h /etc/passwd # 4 Kbytes
```

29. Haga ahora una copia recursiva del directorio `/var/log/packages` en nuestro directorio `work`. Escribir el comando necesario para cambiar los permisos del directorio `$HOME/work/packages` para que el propietario tenga todos los permisos sobre el directorio y los demás usuarios ninguno. Utilice la forma absoluta de describir los permisos.

```
cp -r /var/log/packages ~/work
chmod 700 ~/work
```

30. Indicar el o los comandos necesarios para cambiar el propietario y el grupo del directorio `$HOME/work/packages`, si hemos iniciado la sesión como el usuario `alumno`. Supongamos que el directorio queremos hacerlo pertenecer al usuario `pop` y al grupo `games` creados en la máquina.

```
su
chown pop:games /home/alumno/work
exit
```

Cree un nuevo subdirectorio de su directorio *home* denominado *work2*. Haga ahora una copia recursiva del directorio */var/log/packages* en nuestro directorio *work2*.

31. Indicar el comando necesario para comprimir todos los archivos del directorio *\$HOME/work2/packages*.

```
gzip ~/work2/packages/*
```

32. ¿Cuántos archivos regulares con extensión *".conf"* contiene el directorio */etc* y sus subdirectorios recorridos recursivamente? Escriba el número de archivos y la orden del shell utilizada para su cálculo directo.

```
find /etc -type f -name '*.conf' 2> /dev/null | wc -l    # 28 archivos
```

33. ¿Cuántos archivos dentro del directorio */var/log/packages* y sus subdirectorios tienen en su contenido la cadena de caracteres "Linux"? Escriba el número de archivos y la orden del shell utilizada para su cálculo directo.

```
grep -rl 'Linux' /var/log/packages | wc -l    # 35 archivos
```

34. Mostrar en la salida un listado en formato largo incluyendo los archivos ocultos del directorio */bin* ordenado alfabéticamente en orden inverso por el nombre del archivo. Escriba la orden del shell utilizada.

```
ls -alr /bin
```

35. Escribir un script bash que muestre el nombre de dicho script, el número de argumentos suministrado en la línea de ordenes y el nombre de todos sus argumentos

```
#!/bin/bash
echo "El nombre del script es $0"
echo "El número de argumentos suministrados es $#"
```

```
echo "Los argumentos son: $@"
```

36. Escribir un script bash que reciba un argumento en la línea de ordenes. Este argumento debe corresponder exclusivamente con el nombre de un archivo regular. El script mostrará en formato largo los atributos del archivo. Si el número de argumentos recibido es distinto a uno debe mostrar un mensaje de error.

```
#!/bin/bash
if [ $# -ne 1 ]
then
    echo "Error. Utilización: $0 <nombre_archivo>"
    exit 1
fi

if [ ! -f "$1" ]
then
    echo "Error. $1 no es archivo regular"
    exit 1
fi

ls -l $1
```

37. Escribir un script bash que admita una lista de argumentos formada por nombres de nodos del sistema de archivos y que muestre si los nodos indicados existen o no.

```
#!/bin/bash
for nodo in "$@"
do
    if [ -e $nodo ]
    then
        echo "El nodo $nodo existe"
    else
        echo "El nodo $nodo no existe"
    fi
done
```

38. Escribir un script bash que intercambie el nombre de dos archivos regulares que se indican como argumentos. El script debe comprobar que el número de argumentos suministrado sea correcto y que existan dichos archivos.

```
#!/bin/bash
if [ $# -ne 2 ]
then
    echo "Error. Utilización: $0 <archivo1> <archivo2>"
    exit 1
fi

for nodo in "$@"
do
    if [ ! -f $nodo ]
    then
        echo "Error. El nodo $nodo no es un archivo regular"
        exit 1
    fi
done

mv $1 "/tmp/$1"
if [ $? -ne 0 ]
then
    echo "Error. No se puede mover $1"
    exit 1
fi

mv $2 $1
if [ $? -ne 0 ]
then
    mv "/tmp/$1" $1
    echo "Error. No se puede mover $2"
    exit 1
fi

mv "/tmp/$1" $2
```

39. Escribir un script bash que elimine todos los archivos regulares de longitud cero presentes en la lista de directorios suministrada en la línea de ordenes. Si no se indica ningún directorio se utilizará el directorio actual.



```
#!/bin/bash
shopt -s dotglob # Se recupera nodos ocultos en la expansión de archivos

# Elimina archivo regulares de longitud cero de un directorio
function BorrarArchivos ()
{
    if [ ! -d "$1" ]
    then
        echo "Error. $1 no es un directorio"
    else
        for nombre in $1/*
        do
            if [ ! -s "$nombre" ]
            then
                rm $nombre
            fi
        done
    fi
}

if [ $# -eq 0 ]
then
    BorrarArchivos .
else
    for dir in $@
    do
        BorrarArchivos $dir;
    done
fi
```

40. Escribir un script bash que trunque a los ocho primeros caracteres el nombre de los nodos del sistema de archivo suministrado en la línea de ordenes.

```
#!/bin/bash
limite=8
for nodo in $@
do
    if [ ${#nodo} -gt $limite ]
    then
        mv $nodo ${nodo:0:$limite}
    fi
done
```

41. Escribir un script bash sin utilizar el comando `find` que muestre en la salida estándar los nombres de los archivos regulares ocultos en el directorio suministrado en la línea de ordenes incluyendo sus subdirectorios, si existen. El formato de salida estándar consistirá en una línea por archivo indicando la ruta de dicho archivo a partir del directorio indicado en la línea de ordenes.

```
#!/bin/bash
shopt -s dotglob # Se recupera nodos ocultos en la expansión de archivos
# Muestra archivo regulares ocultos de un directorio de forma recursiva
function MostrarOcultos ()
{
    if [ ! -d "$1" ]
```

```
then
    echo "Error. $1 no es un directorio"
else
    for ruta in $1/*
    do
        if [ -d "$ruta" ]
        then
            MostrarOcultos "$ruta"
        elif [ -f "$ruta" ]
        then
            nombre=${ruta/#$1/} ;
            if [ ${nombre:1:1} == "." ]
            then
                echo "$ruta"
            fi
        fi
    done
fi
}

if [ ! $# -eq 1 ]
then
    echo "Error. Utilización: $0 <nombre_directorio>"
else
    MostrarOcultos $1
fi
```

## 2. Figuras y Tablas

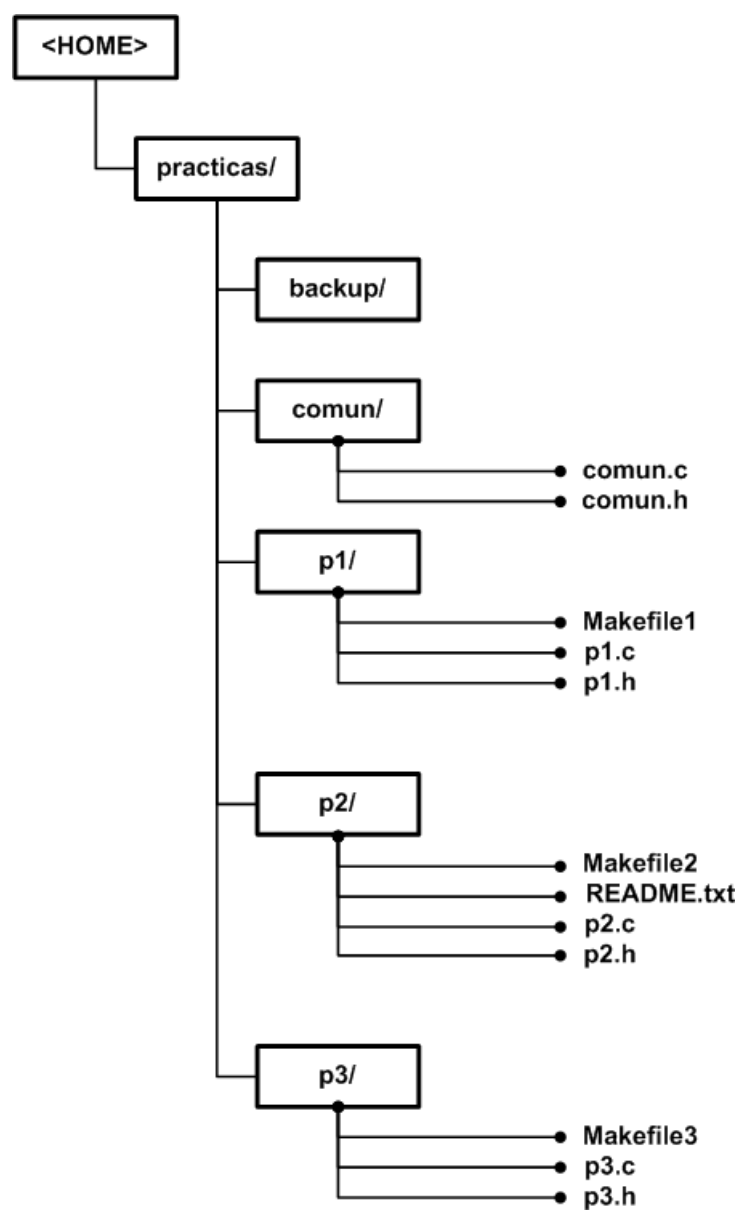


Figura 1: Estructura de directorio