

Introducción: Sistema de ficheros y gestión de aplicaciones /

## 1. Sistema de ficheros y gestión de aplicaciones I

### 1.1 Sabe crear carpetas y ficheros y gestionarlos usando los comandos básicos:

- \* Crea la carpeta "prueba" en tu carpeta personal.  
mkdir prueba
- \* Dentro de la carpeta "prueba" crea otra carpeta "trabajo".  
cd prueba/  
mkdir trabajo
- \* Crea los siguientes ficheros vacíos dentro de la carpeta "trabajo":  
cd trabajo/  
touch gato.txt gata.txt gatita.txt gatito.txt perro.txt perra.txt gallo.txt gallina.txt
- \* Crea una carpeta "alumnos" en tu carpeta personal. Dentro de la carpeta "alumnos" crea 2 ficheros con la siguiente información de cada alumno. Por ejemplo "nano bego" que incluya la siguiente información:  
Nombre: Bego  
Calle: Nieves Cano  
Número: 3  
Teléfono: 945945945  
cd ~  
mkdir alumnos  
cd alumnos  
nano bego  
nano eli
- \* Cambia el nombre de los ficheros a mayúsculas.  
mv bego BEGO  
mv eli ELI
- \* Crea un enlace simbólico para un alumno y un enlace real para el otro. ¿Qué diferencias hay entre los diferentes tipos de enlaces?  
ln -s BEGO begosim  
ln ELI elireal
- \* Lista el contenido de este directorio, con información de inodo, permisos ,propietario, grupo, tamaño, fecha de actualización, nombre, etc...  
808159 4 -rw-rw-r-- 1 xabierland xabierland 64 feb 10 17:13 BEGO  
786452 0 lrwxrwxrwx 1 xabierland xabierland 4 feb 10 17:45 begosim -> BEGO  
786437 4 -rw-rw-r-- 2 xabierland xabierland 60 feb 10 17:42 ELI  
786437 4 -rw-rw-r-- 2 xabierland xabierland 60 feb 10 17:42 elireal
- \* Listar los ficheros de la carpeta personal (tambien los ocultos).  
ls -a  
.  
snap .bash\_history .cache Documents .local .pam\_environment .profile  
Templates  
.. .bash\_logout .config Downloads .mozilla Pictures  
prueba .ssh Videos  
alumnos .bashrc Desktop .gnupg Music .pki  
Public .sudo\_as\_admin\_successful .vscode
- \* Busca los ficheros modificados en menos de 5 minutos de la carpeta personal.  
find -mmin -5
- \* Busca los ficheros cuyo tamaño es superior a 100 Kb de la carpeta personal.  
find -size +500k

---

### 1.2 Sabe manejarse usando rutas absolutas y relativas:

- \* Copia un fichero de uno de los alumnos a la carpeta padre usando la ruta absoluta.  
cp /home/xabierland/alumnos/BEGO /home/xabierland/
- \* Copia un fichero de uno de los alumnos a la carpeta padre usando la ruta relativa.  
cp BEGO ..
- \* Cambia al directorio "prueba", usando la ruta absoluta.  
cd /home/xabierland/prueba
- \* Cambia al directorio "prueba", usando la ruta relativa sabiendo que estamos en el directorio personal.  
cd prueba
- \* Cambia al directorio "prueba", usando la ruta relativa sabiendo que estamos en el directorio "trabajo".  
cd ..
- \* Cambia al directorio "prueba", usando la ruta relativa sabiendo que estamos en el directorio tmp

```
cd ../home/xabierland/prueba
```

1.3 Sabe utilizar las expresiones regulares utilizando la sintaxis globbing utilizando el comando ls:

\* Cambia a la carpeta "trabajo" donde tienes creados los siguientes ficheros vacíos: gato.txt, gata.txt, gatita.txt, gatito.txt, perro.txt, perra.txt, gallo.txt y gallina.txt para realiza los siguientes ejercicios utilizando la sintaxis globbing:

```
cd /home/xabierland/prueba/trabajo
```

\* Lista los archivos cuyo nombre coincida con las palabras:gato.txt, gata.txt, gatito.txt y gatita.txt

```
ls gat*.txt
```

\* Lista los archivos cuyo nombre coincida con las palabras:gato.txt y gata.txt

```
ls gat?.txt
```

\* Lista los archivos cuyo quinto carácter sea "o": gallo.txt y perro.txt

```
ls ???o.txt
```

\* Lista los archivos cuyo segundo y cuarto carácter sea una vocal: gato.txt, gata.txt,gatito.txt y gatita.txt

```
ls ?[aeiou]?[aeiou]*.txt
```

1.4 Sabe utilizar las expresiones regulares utilizando la sintaxis regex utilizando el comando grep:

\* Buscar las líneas en las que aparece la palabra bash en el archivo /etc/passwd.

```
grep 'bash' /etc/passwd
```

\* Buscar en el archivo /etc/group todas las líneas que empiezan por r.

```
grep '^r' /etc/group
```

\* Buscar las líneas en las que aparece la palabra bash al final de la línea en el archivo /etc/passwd.

```
grep 'bash$' /etc/passwd
```

\* Buscar las líneas de /etc/passwd de los usuarios cuyo identificador de usuario es menor que 1000.

```
grep -e ':x:[0-9][0-9][0-9]' /etc/passwd
```

\* Buscar las líneas de /etc/passwd de los usuarios cuyo identificador de usuario es mayor o igual a 1000.

```
grep -e ':x:[1-9][0-9][0-9][0-9]' /etc/passwd
```

\* Buscar las líneas que contienen 2 o más apariciones de "o" seguidas en el fichero /etc/passwd.

```
grep -e 'o{2,}' /etc/passwd
```

1.5 Sabe gestionar usuarios

\* Crea un nuevo usuario "lsi1" con contraseña "lsi1" utilizando el comando useradd. (man useradd)

```
sudo useradd lsi1
```

```
sudo passwd lsi1
```

\* Crea un nuevo usuario "lsi2" con contraseña "lsi2" utilizando el comando adduser. (man adduser)

```
sudo adduser lsi2
```

\* Crea el grupo "desarrolladores" y añade a "lsi1" y "lsi2" a dicho grupo.

```
sudo groupadd desarrolladores
```

```
sudo usermod -a -G desarrolladores lsi1
```

```
sudo usermod -a -G desarrolladores lsi2
```

\* Dale el poder de administrar el grupo "desarrolladores" al jefe de proyecto "lsi2"

```
sudo gpasswd -A lsi2 desarrolladores
```

\* Dale a "lsi2" los mismos permisos que el administrador del sistema añadiendo los mismos grupos que el administrador del sistema.

```
sudo usermod -G adm,cdrom,sudo,dip,plugdev,lpadmin,lxd,sambashare lsi2
```

\* Con el comando id observa los grupos a los que pertenece "lsi1" y "lsi2". Explica para qué sirve cada grupo.

```
uid=1001(lsi1) gid=1001(lsi1) grupos=1001(lsi1),1003(desarrolladores)
```

```
uid=1002(lsi2) gid=1002(lsi2) grupos=1002(lsi2),1003(desarrolladores),4(adm),-24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
```

```
adm Monitor system logs
```

```
cdrom Use CD-ROM drives
```

```
lpadmin Configure printers
```

```
sudo administer the system
```

```
sambashare Share files ith the local network
```

```
dip Connect to the Internet using a modem
```

```

plugdev          Access external storage devices
* Finalmente elimina las cuentas creadas (lsi1 y lsi2) y el grupo "desarrolladores"
sudo userdel lsi1
sudo userdel lsi2
sudo groupdel desarrolladores
=====
1.6 Sabe manejarse con los permisos de usuario:
* Crear un fichero llamado "reloj.sh" añádele como contenido el comando "date".
touch reloj.sh
nano reloj.sh
* Darle permisos de ejecución a "reloj.sh"
chmod +x reloj.sh
* Ejecuta "reloj.sh"
./reloj.sh
* Crea un fichero "SoyNuevo" para asignarle usando el modo octal y usando el modo literal
los siguientes permisos:
- El propietario tenga permisos de lectura, escritura y ejecución
- El grupo solo lectura
- El resto u otros solo lectura
touch SoyNuevo
chmod 744 SoyNuevo | chmod u=rwx,g=r,o=r SoyNuevo
* Comprueba la máscara actual con el comando (man umask).
umask
* Asigna una nueva máscara "umask 077", crea ficheros y carpetas para comprobar que la
máscara funciona correctamente.
umask 077
mkdir example
124 4 drwx----- 2 xabierland xabierland 4096 feb 10 22:33 example
* Finalmente deja la máscara como estaba al principio por defecto.
umask 0022
* ¿Cuál es el comando para que las carpetas y ficheros que se creen en dicha sesión
tengan los siguientes permisos:
- El propietario tenga permisos de lectura, escritura y ejecución
- El grupo solo lectura y escritura
- El resto u otros solo lectura
- Finalmente deja la máscara como estaba al principio por defecto.
umask 764
umask 022
* Para siempre (futuras sesiones) modifica el fichero ".bashrc" con la siguiente máscara
"umask 0002".
cd ~
nano .bashrc
* Añádele el bit s al "reloj.sh"
chmod u+s reloj.sh
* Detecta el bit sticky en la carpeta tmp, crea un archivo con un usuario e intenta
borrar con otro usuario. ¿Puedes?
No ya que al ser creado por otro usuario con el sticky bit el resto de usuarios
solo tienen permisos para leerlo

```

Lengoaia eta Sistema Informatikoak Saila