

Práctica 1 del Laboratorio de Sistemas Operativos

Departamento de Automática
Universidad de Alcalá



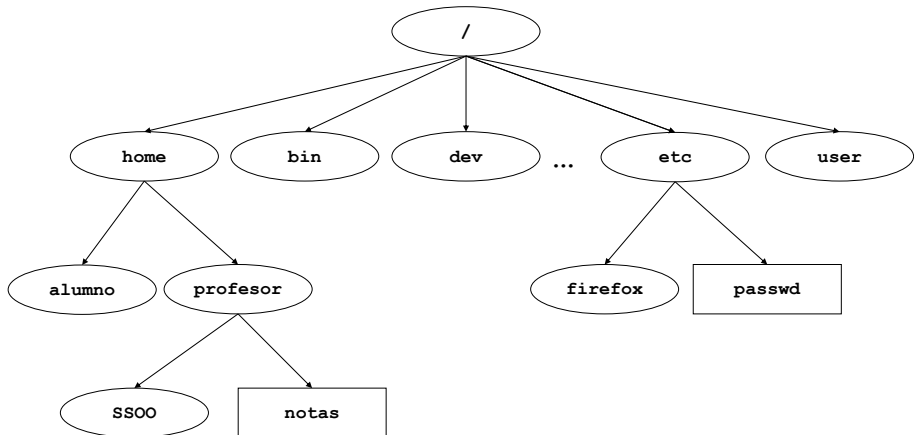
/gso>

Índice

1 Práctica 1

- Rutas absolutas vs relativas
- Repaso preguntas frecuentes semana anterior
- Comandos chmod y umask
- Enlaces

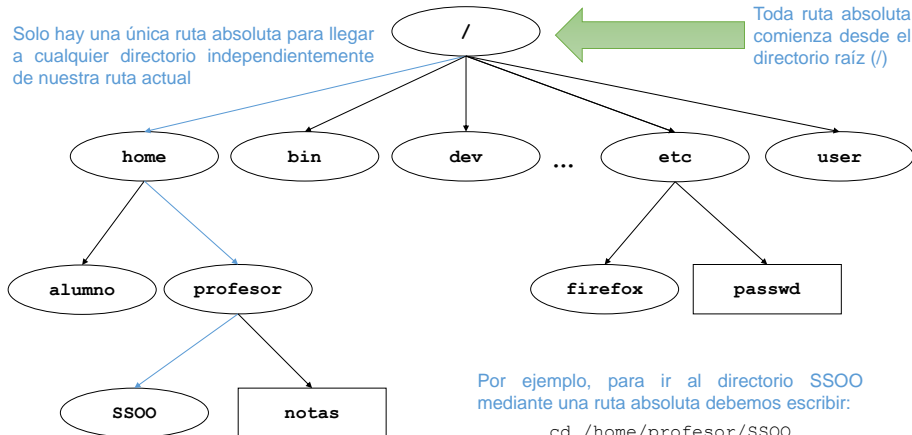
Estructura jerárquica de Linux



Rutas absolutas

Solo hay una única ruta absoluta para llegar a cualquier directorio independientemente de nuestra ruta actual

Toda ruta absoluta comienza desde el directorio raíz (/)

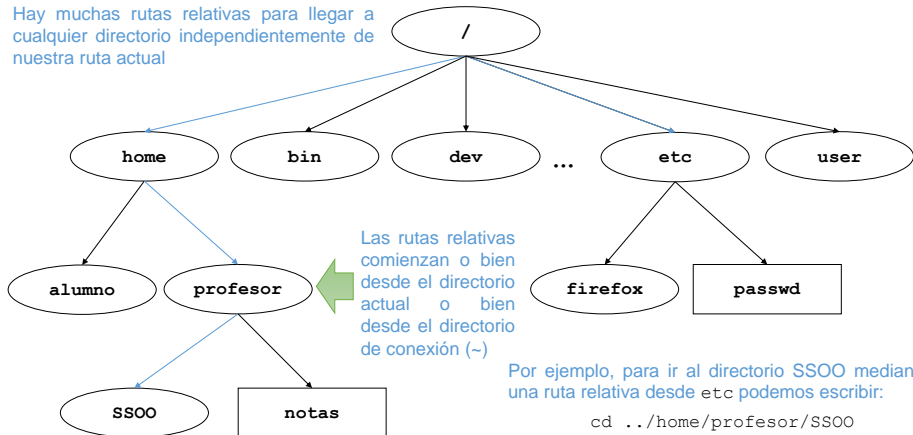


Por ejemplo, para ir al directorio SSOO mediante una ruta absoluta debemos escribir:

```
cd /home/profesor/SSOO
```

Rutas relativas

Hay muchas rutas relativas para llegar a cualquier directorio independientemente de nuestra ruta actual



Por ejemplo, para ir al directorio SSOO mediante una ruta relativa desde etc podemos escribir:

```
cd ../home/profesor/SSOO
cd ~/SSOO
cd ../../home/profesor/SSOO
```

Comando `echo -e`

- 1 El carácter *backslash* (`\`) es un carácter de escape.
- 2 Para poder emplear `\` en cualquiera de los comandos es necesario escribir `\\`.
- 3 Probad las distintas secuencias de escape para ver cuál es el resultado: `\n`, `\r`, `\t`, `\\`, etc.

Carácter comodín

- Para expresar reglas complejas se emplean las denominadas **expresiones regulares**.
 - ① número.
 - ② no mayúscula.
 - ③ varias consonantes seguidas.
 - ④ a, D, 4 o p.
- Busca cómo funcionan las expresiones regulares, crea vuestros ejemplos y prueba su funcionamiento.

Permisos

- No tiene el mismo significado el permiso para un directorio que para un archivo.
- Los permisos son clave en tareas como la creación de **enlaces**.
- Los permisos se pueden modificar con `chmod` o `umask` (hay que saber cómo funcionan los dos).

Archivos		Directorios	
Lectura	- Leer o visualizar el contenido	- Visualizar el contenido	
Escritura	- Modificar o eliminar el contenido	- Modificar o eliminar el contenido	(necesita permiso de ejecución)
	- Modificar permisos y propietario		
Ejecución	- Ejecutar un archivo / script	- Entrar en la carpeta	

Comando chmod

\$ chmod [modificadores] permisos fichero/directorio

Existen permisos para tres tipos de usuario: 1º Dueño (u) 2º Grupo (g) y 3º Otros usuarios (o).

Número	Binario	Lectura (r)	Escritura (w)	Ejecución (x)
0	000	×	×	×
1	001	×	×	✓
2	010	×	✓	×
3	011	×	✓	✓
4	100	✓	×	×
5	101	✓	×	✓
6	110	✓	✓	×
7	111	✓	✓	✓

Modo relativo

chmod **u-r** prueba.c

- Se le quita al usuario el permiso de lectura del fichero prueba.c

chmod **go=x** prueba.c

- Se le fija al grupo y al resto de usuarios el permiso de ejecución del fichero prueba.c

chmod **a+w** prueba.c

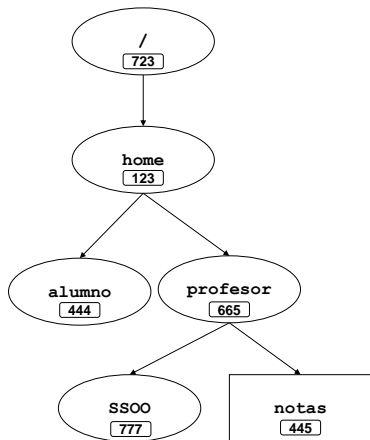
- Se le añade a todos el permiso de escritura del fichero prueba.c

Modo absoluto

chmod **345** prueba.c

- El primer número se corresponde con el dueño del archivo. En el ejemplo tiene el número 3. Es decir el dueño tiene permisos de escritura y ejecución pero no de lectura.
- El segundo número se corresponde con el grupo al que pertenece el fichero. En el ejemplo es el número 4. Es decir el grupo tiene permisos de lectura pero ni de escritura ni de ejecución.
- El tercer número se corresponde con todos los demás usuarios. En el ejemplo representa el número 5. Es decir el resto de usuarios tiene permisos de lectura y ejecución pero no de escritura.

Ejemplo de `chmod`



1. Si me encuentro en el directorio raíz (/), ¿Qué debería cambiar para acceder como dueño al archivo `notas`?
2. Si me encuentro en el directorio `alumno`, ¿Qué permisos debo modificar para modificar el archivo `notas` si soy un usuario del mismo grupo que el dueño?
3. Si me encuentro en el directorio `profesor` y soy un usuario que pertenece al grupo "otro", ¿Qué permisos debo modificar para poder ejecutar el archivo `notas`?

Comando umask (I)

\$ umask [modificadores] permisos

La mayoría de sistemas Unix no permiten crear archivos con permisos de ejecución

Número	Binario	Lectura (r)	Escritura (w)	Ejecución (x)
0	000	×	×	×
1	001	×	×	✓
2	010	×	✓	×
3	011	×	✓	✓
4	100	✓	×	×
5	101	✓	×	✓
6	110	✓	✓	×
7	111	✓	✓	✓

Máscara simbólica

umask **u=r,g=x,o=**

- Los nuevos directorios tendrán permisos de lectura para el dueño y de ejecución para el grupo. Los archivos no tendrán el permiso de ejecución.

umask **u=rwx,g=,o=**

- Los nuevos directorios tendrán todos los permisos para el dueño. Los archivos no tendrán el permiso de ejecución.

umask **u=x,g=x,o=x**

- Los nuevos directorios tendrán permisos de ejecución siempre. Los archivos no tendrán ningún permiso.

Máscara en octal

umask 0235

Los nuevos directorios tienen los permisos opuestos a la máscara en octal.

2 = (010) 3 = (011) 5 (101)
Opuesto (101 100 010) = 542

- Con la máscara 0235 los nuevos directorios tendrán los permisos R-X R- -W-
- Con la máscara 0235 los nuevos archivos tendrán los permisos R- R- -W- porque los archivos no se crean con permisos de ejecución.

¿Qué máscara en octal habría que poner para que todos los directorios nuevos se creasen con permisos de lectura y escritura?

¿Y que máscara simbólica?

Comando umask (II)

- Los directorios y archivos pueden tener permisos diferentes aun teniendo la misma máscara.
- Máscara por defecto:
 - Si eres usuario: 002
 - Si eres administrador: 022

```
root@user:/home/user/prueba# ls -l
total 8
-rw-rw-r-- 1 user  user    0 feb 15 09:06 archivoDePrueba
-rw-r--r-- 1 root  root    0 feb 15 09:09 archivoDePruebaAdmin
drwxrwxr-x 2 user  user  4096 feb 15 09:06 carpetaDePrueba
drwxr-xr-x 2 root  root  4096 feb 15 09:08 carpetaDePruebaAdmin
```

Enlaces

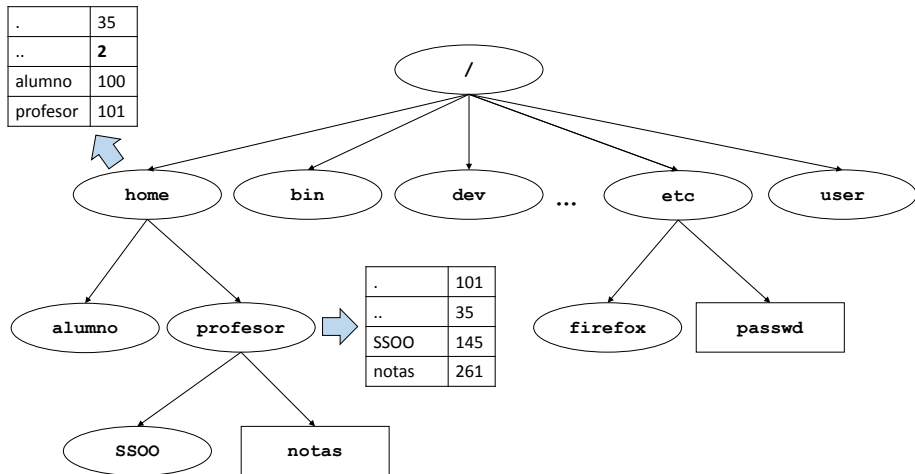
- Referencia a un archivo.
- Existe una relación con el concepto nodo-i.
- Imprescindible conocer el funcionamiento de los permisos de Linux (ver diapositiva 8).
- En Linux existen dos tipos de enlaces:
 - 1 Enlace fuerte, físico o duro.
 - 2 Enlace débil, simbólico o blando.

Nodo-i

- El nodo-i o *inode* es una estructura de datos que contiene las características de un archivo dentro de un sistema Unix.
- Puede identificarse mediante el comando `ls -li`.
- Un directorio es en realidad un archivo que contiene un conjunto de pares (nombre de archivo y el nodo-i).

```
user@user:~/SS00/enlaces$ ls -li
total 4
4204763 -rw-rw-r-- 2 user user 0 feb 22 10:33 archivo1
4204764 -rw-rw-r-- 1 user user 0 feb 22 10:33 archivo2
5244294 drwxrwxr-x 2 user user 4096 feb 22 10:33 directorio1
4204765 lrwxrwxrwx 1 user user 8 feb 22 10:34 enlaceDebilAArchivo1 -> archivo1
4204766 lrwxrwxrwx 1 user user 12 feb 22 10:35 enlaceDebilADirectorio1 -> directorio1/
4204763 -rw-rw-r-- 2 user user 0 feb 22 10:33 enlaceFuerteAArchivo1
```

Representación de directorios



Enlaces fuertes (I) - Definición

Enlace fuerte

En sistemas Unix/Linux, un enlace fuerte o físico es una copia exacta del archivo original con distinto nombre del archivo.

```
ln [Origen] [Destino]
```


Enlaces fuertes (II) - Características

- Comparten el nodo-i con el archivo origen.
- Aumenta el valor del campo destinado a mantener el número de enlaces en el nodo-i.
- **NO** pueden aplicarse a directorios.
- **NO** pueden aplicarse a diferentes sistemas de archivos.
- **NO** duplica el archivo (tamaño dedicado es el mismo).

Enlaces fuertes (III) - Ejemplo

Fase 1 – Situación original

Archivo1



Archivo2



Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo1	10000	Archivo ordinario	Hola original	1
Archivo2	10001	Archivo ordinario	Adiós original	1

Fase 2 – Creación enlace fuerte

```
ln /home/Alumno1/apuntes/Archivo1 Enl-Fuerte-A1
```

Archivo1



Archivo2



Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo1	10000	Archivo ordinario	Hola original	2
Archivo2	10001	Archivo ordinario	Adiós original	1
Enl-Fuerte-A1	10000	"Enlace fuerte"	Hola original	2

Fase 3 – Modificación archivo original

Archivo1



Archivo2



Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo1	10000	Archivo ordinario	Hola modificado	2
Archivo2	10001	Archivo ordinario	Adiós original	1
Enl-Fuerte-A1	10000	"Enlace fuerte"	Hola modificado	2

Enlaces fuertes (IV) - Ejemplo

Fase 4 – Eliminación original



Archivo2

Adiós original

Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Enl-Fuerte-A1	10000	"Enlace fuerte"	Hola modificado	1
Archivo2	10001	Archivo ordinario	Adiós original	1

Fase 5 – Creación nuevo enlace fuerte

Archivo2

Adiós original

```
ln /home/Enl-Fuerte-A1 Enl-Fuerte2-A1
```

Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Enl-Fuerte-A1	10000	"Enlace fuerte"	Hola modificado	2
Archivo2	10001	Archivo ordinario	Adiós original	1
Enl-Fuerte2-A1	10000	"Enlace fuerte"	Hola modificado	2

Fase 6 – Eliminación enlaces fuertes

Archivo2

Adiós original

Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo2	10001	Archivo ordinario	Adiós original	1

Enlaces débiles (I) - Definición

Enlace débil

En sistemas Unix/Linux, un enlace débil o simbólico es un archivo especial, el cual, no almacena el contenido del archivo si no la ruta del archivo original.

```
ln -s [Origen] [Destino]
```

Enlaces débiles (II) - Características

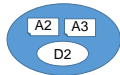
- Crean un nuevo nodo-i.
- Es un nuevo tipo de archivo denominado *link* (l).
- No supone un incremento del número de enlaces en el nodo-i.
- **SÍ** pueden aplicarse a directorios.
- **SÍ** pueden aplicarse a diferentes sistemas de archivos.
- **NO** duplica el archivo.
- Necesita otro nodo-i y dos accesos a disco.

Enlaces débiles (III) - Ejemplo

Fase 1 – Situación original

Archivo1

Directorio1

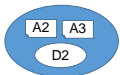


Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo1	10000	Archivo ordinario	Hola original	1
Directorio1	10001	Directorio	A2, A3, D2	2

Fase 2 – Creación enlace débil a archivo

Archivo1

Directorio1



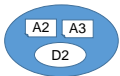
```
ln -s /home/Archivo1 Enl-Debil-A1
```

Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo1	10000	Archivo ordinario	Hola original	1
Directorio1	10001	Directorio	A2, A3, D2	2
Enl-Debil-A1	10002	Link	/home/Archivo1	1

Fase 3 – Modificación archivo original

Archivo1

Directorio1



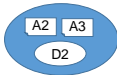
Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Archivo1	10000	Archivo ordinario	Hola modificado	1
Directorio1	10001	Directorio	A2, A3, D2	2
Enl-Debil-A1	10002	Link	/home/Archivo1	1

Enlaces débiles (IV) - Ejemplo

Fase 4 – Eliminación original



Directorio1



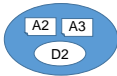
Al eliminar el archivo original, el enlace débil apunta a una dirección inexistente y se convertirá en un enlace roto

Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Directorio1	10001	Directorio	A2, A3, D2	2
Enl-Debil-A1	10002	Link	/home/Archivo1	1

Fase 5 – Creación enlace débil directorio

```
ln -s /home/Directorio1 Enl-Debil-D1
```

Directorio1



Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Directorio1	10001	Directorio	A2, A3, D2	2
Enl-Debil-D1	10002	Link	/home/Directorio1	1

Fase 6 – Eliminación enlace débil

Directorio1



Archivo	Inode	Tipo archivo	Contenido	Nº Enlaces
Directorio1	10001	Directorio	A2, A3, D2	2