

Programación con Shell Scripting: Sesión 3

Máster Universitario en Bioinformática



Universidad
Internacional
de Valencia

Dra. Paula Soler Vila
paula.solerv@professor.universidadviu.com

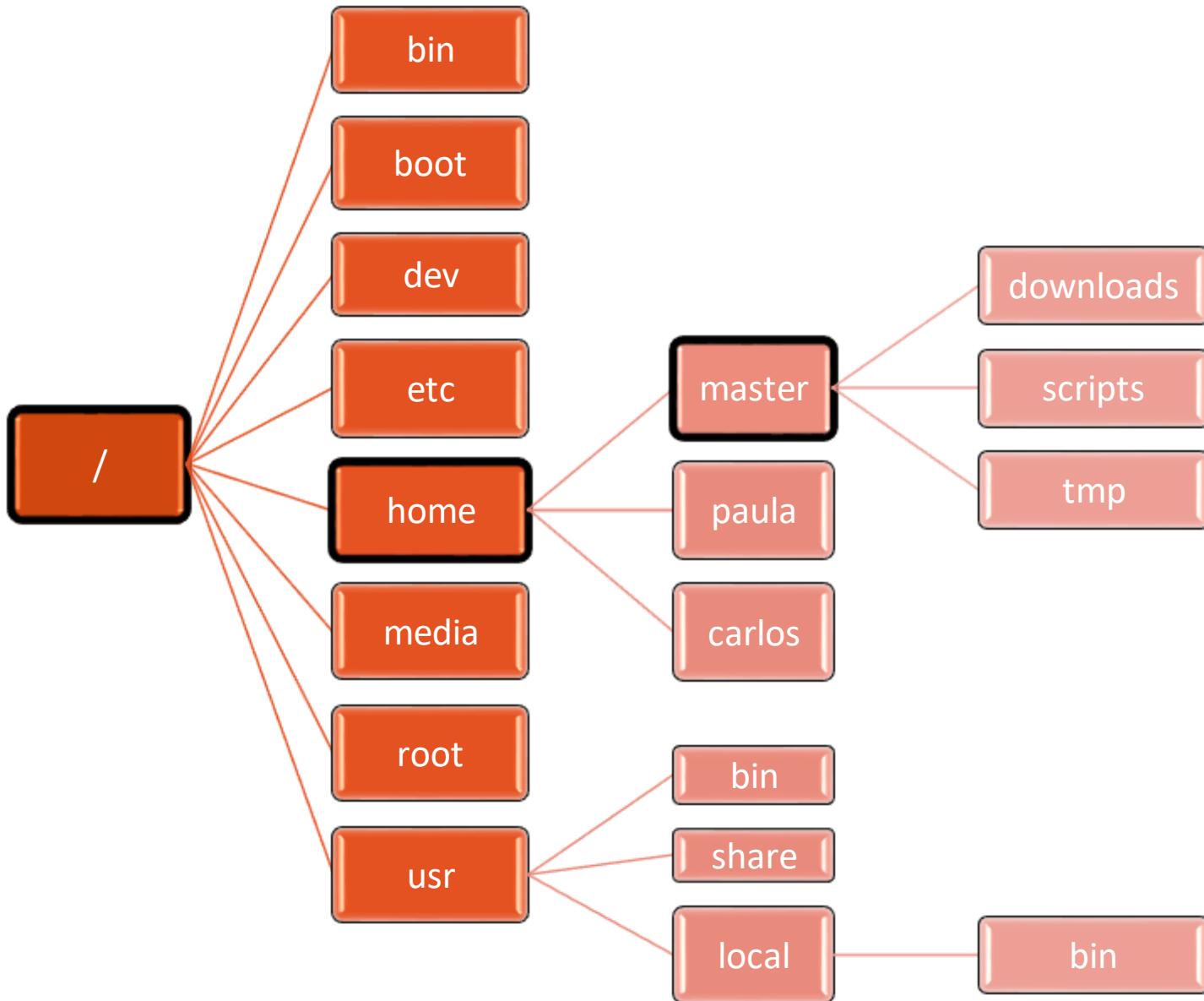
De:
 Planeta Formación y Universidades



Aspectos a tratar

- 1 Ejercicio: Sistema y jerarquía de archivos
Comandos *locate / find*
- 2 Flujo de datos y redirecciones
- 3 Metacaracteres
Búsqueda de archivos y flujo de datos

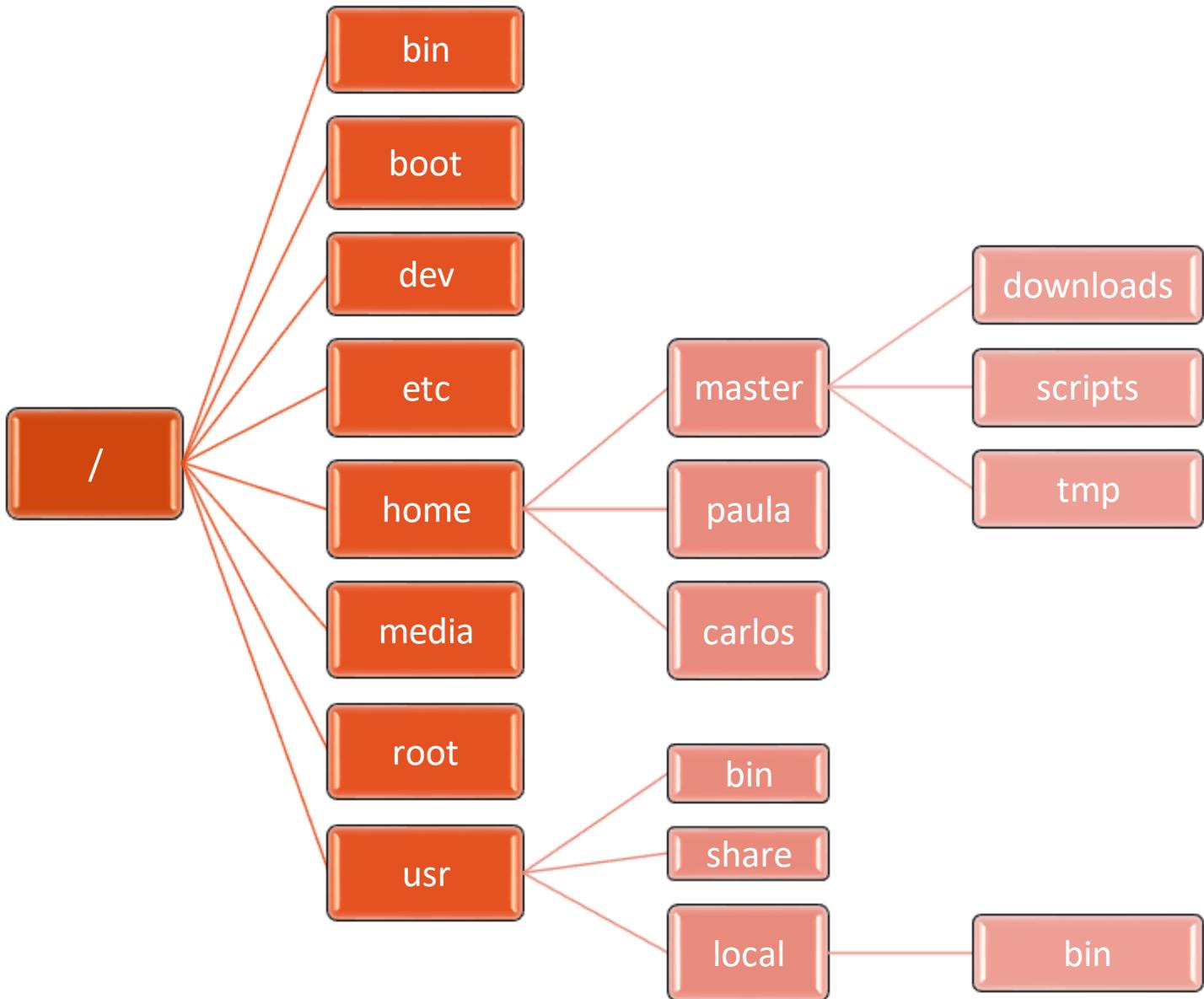
Sistema y jerarquía de archivos -> Ejercicio I



MATE Terminal

```
> pwd  
/home/master  
  
> cd ../../  
> pwd  
>/
```

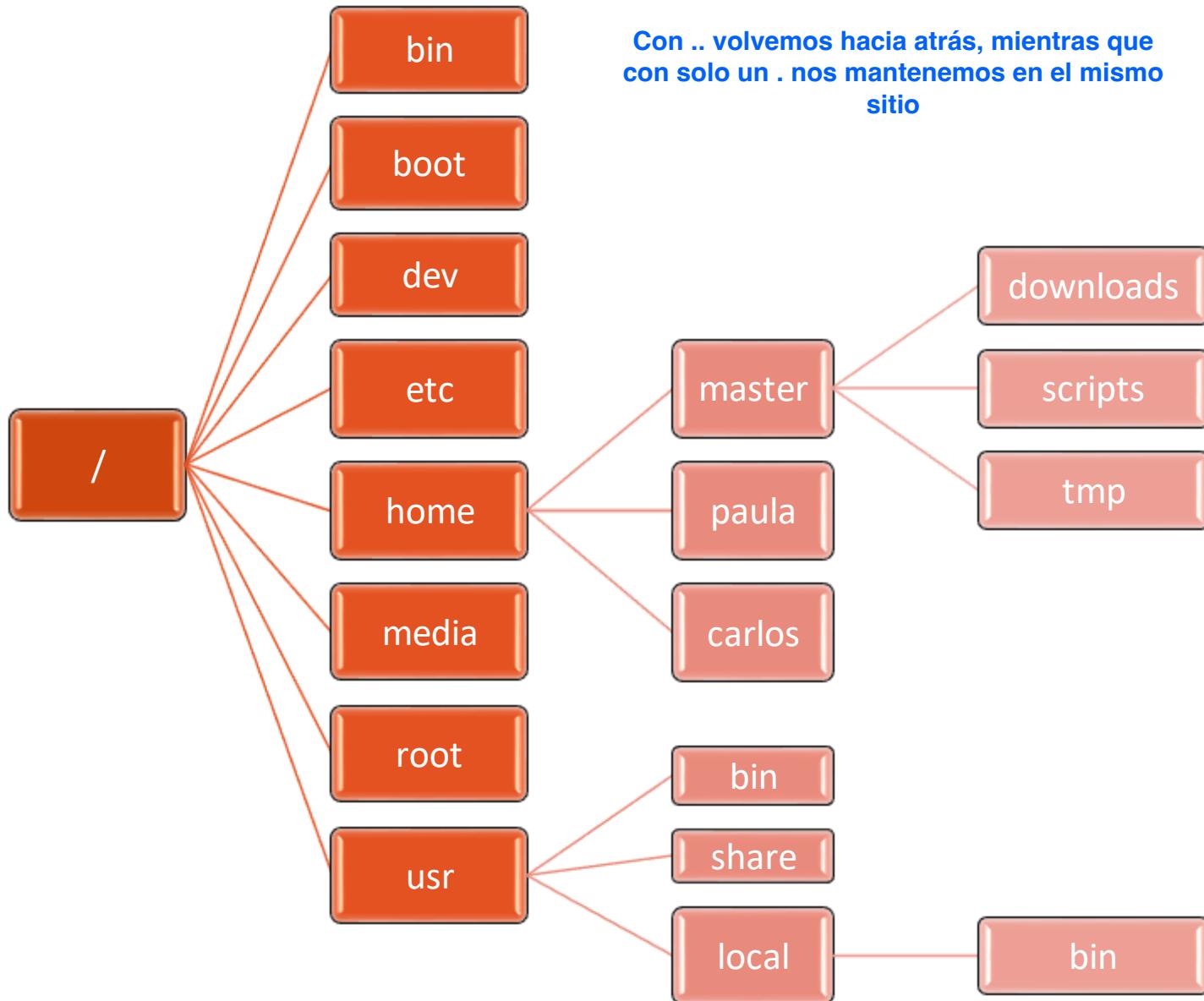
Sistema y jerarquía de archivos -> Ejercicio I



MATE Terminal

```
> cd ../../home/paula/../../master  
> pwd  
> /home/master
```

Sistema y jerarquía de archivos -> Ejercicio I



MATE Terminal

```
>cd /home/master/downloads/..../downloads  
/home/master/downloads/  
  
> cd /home/master/..../master/..../usr/..../bin  
/bin  
  
> cd /home/./master/.tmp/..../tmp/././.  
/home/master/tmp  
  
> cd /home/././master/././downloads  
/home/master/downloads
```

¿Dónde está mi archivo?

¿Dónde esta mi archivo? → Comando *locate*

No va a encontrar archivos recientes, hay que actualizar

```
$ sudo yum install mlocate  
$ sudo updatedb
```

```
$ locate GRCh38_latest_genomic.fna  
/home/paula.soler/Asignaturas/Programación_con_Shell_Scripting/Sesiones/Datasets/GRCh38_latest_genomic.fna.gz  
/home/paula.soler/human_genome_hg38/GRCh38_latest_genomic.fna /home/paula.soler/test/GRCh38_latest_genomic.fna  
/volumes/user/home/paula.soler/Asignaturas/Programación_con_Shell_Scripting/Sesiones/Datasets/GRCh38_latest_genomic.fna.gz  
/volumes/user/home/paula.soler/human_genome_hg38/GRCh38_latest_genomic.fna  
/volumes/user/home/paula.soler/test/GRCh38_latest_genomic.fna
```

```
$ touch testeando_locate.txt  
$ ls  
Asignaturas Desktop Documents Downloads human_genome_hg38 miniconda3 MultiQC Music Pictures Public R  
Templates test testeando_locate.txt Videos  
$ locate testeando_locate.txt  
$
```

¿Dónde esta mi archivo? → Comando *find*

```
find <inicio_búsqueda><opciones><término_busqueda>
```

- / → busca en todo el sistema.
- . → busca en la carpeta en la que estás trabajando actualmente (directorio actual).
- ~ → para buscar desde tu directorio home.
/home/paula.soler/Documents -> Adicionar una ruta específica

- name / -iname
- type
- time
- size
- user
- perm

- **find . -type f -empty**

Sirve para buscar, y luego puedes filtrar como por ejemplo aquí con -empty para archivos vacíos

PRACTIQUEMOS



Ejemplos de find (history I)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ wget sanchezcorbalan.es/ficheros.tar.gz  
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ tar -xzf ficheros.tar.gz  
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cd ficheros
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ tree -L 2
```

```
.  
├── documentos  
│   ├── borradores  
│   ├── carta-al-presidente.txt -> cartas/carta-al-presidente.txt  
│   ├── cartas  
│   ├── informe-2009.txt  
│   └── informes  
├── fotos  
│   ├── escritorio-ubuntu.jpeg  
│   ├── logo-ubuntu.gif  
│   ├── personales  
│   ├── shell-commands.gif  
│   ├── tux-clasico.jpeg  
│   └── tux-moderno.png  
└── musica  
    ├── 2009  
    └── 2010
```

9 directories, 7 files

Tar para extraer archivos comprimidos en formato tar, que se utiliza para combinar varios archivos en un solo archivo

- x para indicar que queremos extraerlo de un archivo comprimido

- z para decir que se trata de un archivo en formato zip y con la opción -f se le dice que archivos extraer (ficheros)

Si se quiere por colores hay que poner -LC

Ejemplos de find (history II)

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -iname "angel.mp3"  
./musica/2009/internacional/The Corrs/Angel.mp3
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -name "angel.mp3"
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -iname "*.mp3"
```

```
./musica/2009/internacional/The Corrs/Hideaway.mp3
```

```
./musica/2009/internacional/The Corrs/Summer.mp3
```

```
./musica/2009/internacional/The Corrs/Angel.mp3
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -iname "*.mp3" -o -iname "*.jpg"
```

```
./fotos/personales/pescando.jpg
```

```
./fotos/personales/mi_movil.jpg
```

```
./fotos/personales/torre-de-pisa.jpg
```

```
./musica/2009/internacional/The Corrs/Hideaway.mp3
```

```
./musica/2009/internacional/The Corrs/Summer.mp3
```

```
./musica/2009/internacional/The Corrs/Angel.mp3
```

```
./musica/2009/nacional/sabina/peces de ciudad.mp3
```

En vez de -o también se puede poner &&

Con la partícula -i antes del name
le estás diciendo que no sabes si
está en mayúsculas o
minúsculas el archivo que
quieres encontrar

El * se usa como un comodín

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -type d -iname "e*"  
./documentos/informes/estadisticas
```

aquí le dice que busque todos los directorios que empiecen por E

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -size +10k
```

```
./fotos/shell-commands.gif
```

```
./fotos/personales/pescando.jpg
```

```
./fotos/personales/mi_movil.jpg
```

```
./fotos/personales/torre-de-pisa.jpg
```

```
./fotos/personales/con mi portatil.jpg
```

```
./fotos/personales/en-el-monte.jpg
```

```
./fotos/tux-moderno.png
```

cualquier comando dentro de una ayuda, por ejemplo, man find puedes buscar luego dentro
de esa ayuda poniendo / y escribiendo la palabra, y para moverte por donde aparezcan todas
esas palabras, pulsando n minuscula

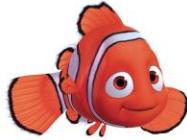
Ejemplos de find (history III)

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ficheros]$ find . -iname "e*" -exec file {} \;
./documentos/informes/estadisticas: directory
./documentos/informes/estadisticas/estadisticas de compras.txt: ASCII text
./documentos/informes/estadisticas/estadisticas de ventas.txt: ASCII text
./fotos/personales/en-el-monte.jpg: JPEG image data, JFIF standard 1.01, comment: "CREATOR: gd-jpeg v1.0 (using IJG JPEG v62), quality = 100"
./fotos/escritorio-ubuntu.jpeg: JPEG image data, JFIF standard 1.01
```



- **-exec file {} \;**: Aquí se especifica que, una vez encontrados los archivos que cumplen el criterio, se debe ejecutar el comando file en cada uno de ellos.
 - {} es un marcador que se reemplazará con el nombre de cada archivo encontrado
 - \; indica el final de la ejecución del comando para cada archivo.

Lo hemos encontrado!!! Cambiamos ubicación -> comando **mv**



comando **mv**

Mover archivos (comportamiento distinto al comando **cp**)
Modificar el nombre de archivos

```
mv <opciones><DESTINO>
```

Se recomienda siempre usar la opción **-i** porque si hay otro archivo que se llame igual directamente lo va a sobreescribir, sin embargo, usando **-i**, si encuentra otro archivo que se llame igual, te pregunta si sobreescribir o no

```
> ls /home/master  
file_1.txt  
  
# Mover un archivo individualmente  
mv file_1.txt /home/master/  
  
mv file_1.txt -i /home/master/  
mv: overwrite '/home/master/file.txt'?
```

```
# Mover múltiples archivos  
mv *.txt /home/master/
```

```
# Renombrar un archivo  
mv file_1.txt FILE_1.txt
```



Aspectos a tratar

1

Ejercicio: Sistema y jerarquía de archivos
Comando *find*

2

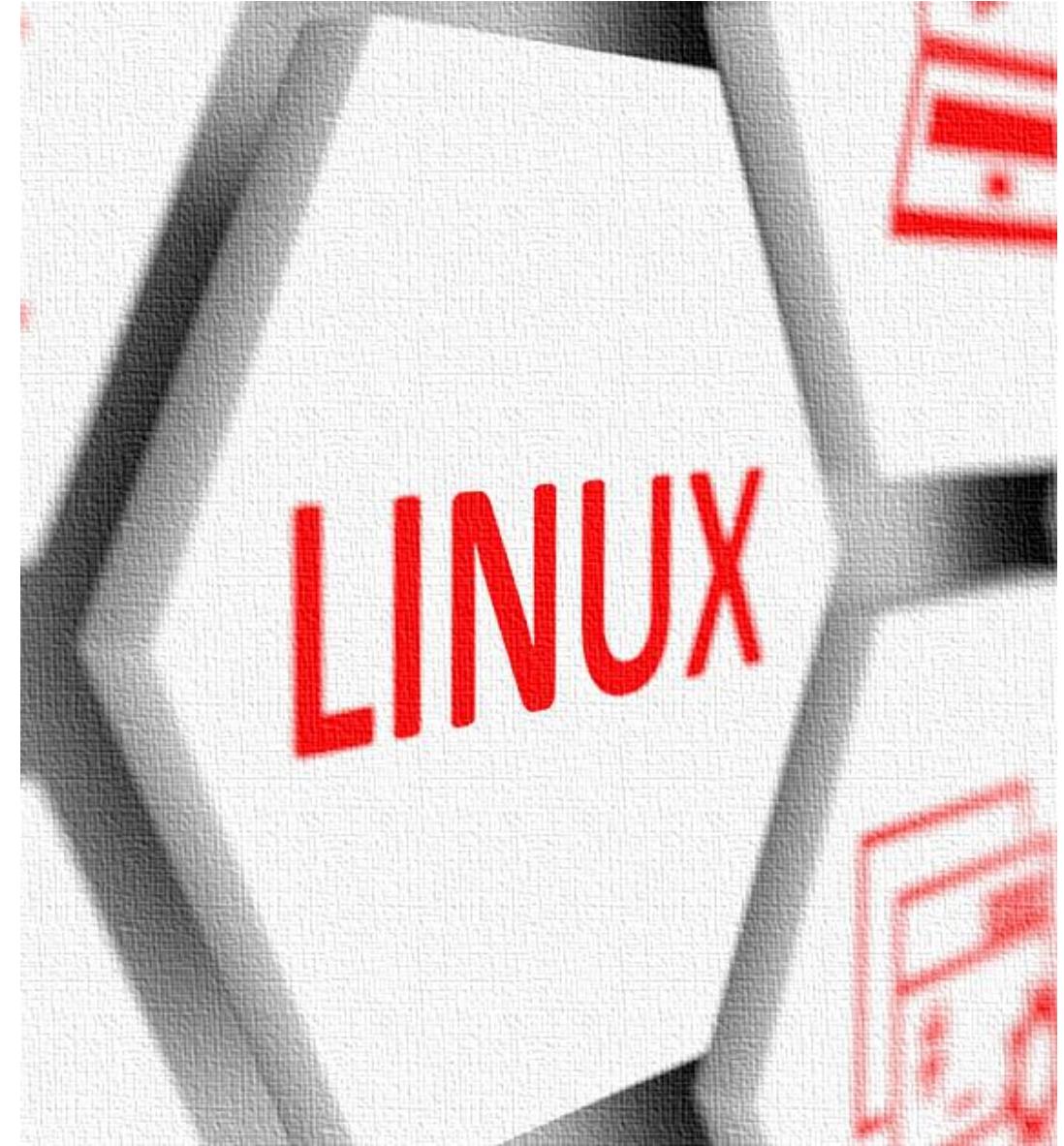
Flujo de datos y redirecciones

3

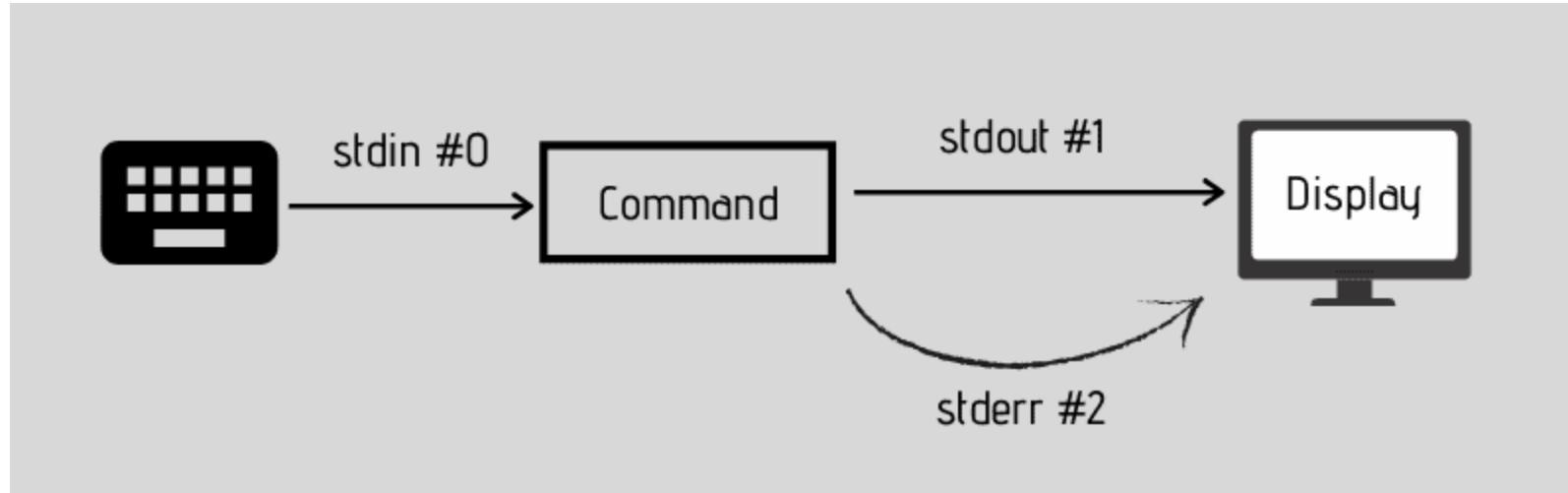
Metacaracteres
Búsqueda de archivos y flujo de datos

Flujo de Datos

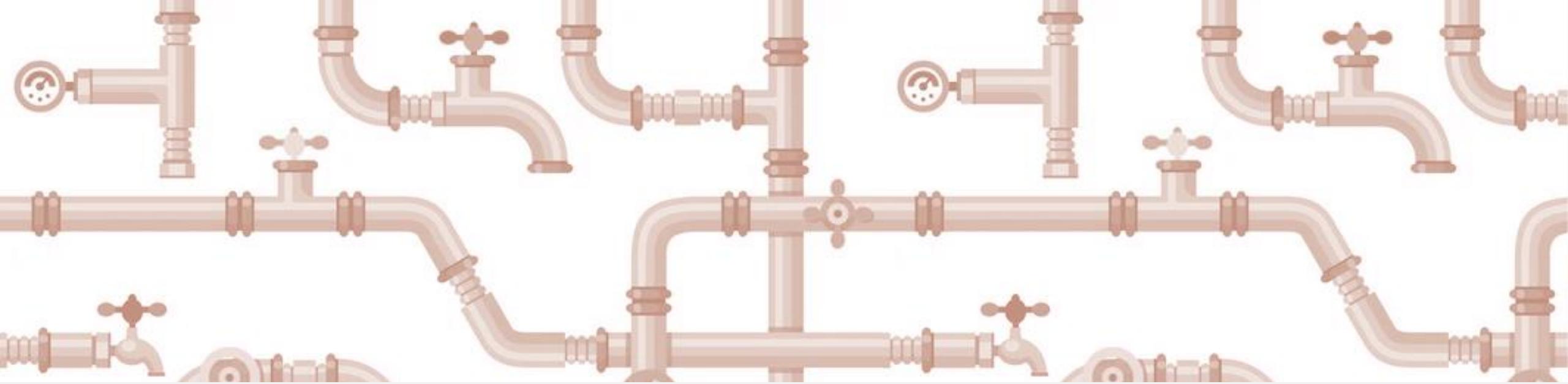
(transmisión de la información)



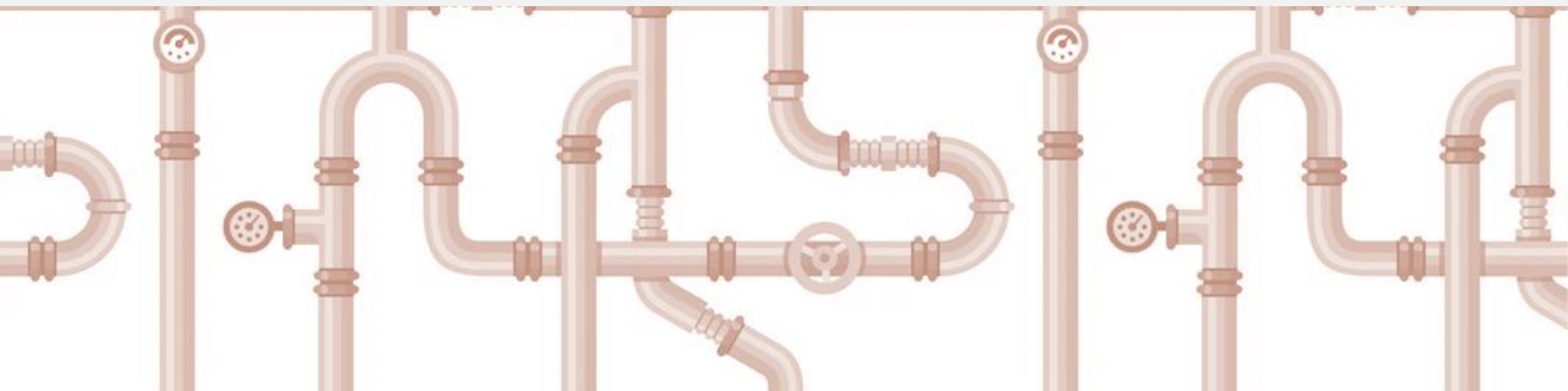
Stdin, stdout & stderr



- La entrada estándar (**standard input - *stdin***) es la fuente de los datos de entrada. Su ID es 0.
- La salida estándar (**standard output - *stdout***) es el resultado del comando. Su ID es 1.
- Error estándar (**standard error - *stderr***) es el mensaje de error (si lo hay) producido por los comandos. Su ID es el 2.



¿Por qué nos interesa el flujo de datos? -> Redirección E/S



Metacaracteres

Un **metacarácter** es un carácter escrito que tiene un significado especial para el *shell* que se usa para conectar comandos, realizar búsquedas específicas, etc.

Símbolo	Significado
>	Redirección de salida: escribe la salida estándar sobre un fichero.
>>	Redirección de salida: añade la salida estándar a un fichero.
<	Redirección de entrada: lee la entrada estándar de un fichero.
<<toc	Redirección de entrada: lee la entrada estándar de un script hasta encontrar la palabra toc .
*	Comodín de sustitución: sustituye a 0 o más caracteres.
?	Comodín de sustitución: sustituye a 1 carácter.
[...]	Comodín de sustitución: sustituye cualquier carácter entre corchetes.
'comando'	Sustitución de comando (ver echo).
	Encauzamiento: envía la salida estándar sobre la entrada estándar de otro.
;	Usado en el secuenciamiento de comandos.
	Ejecución condicional: ejecuta un comando si el previo ha fallado.
&&	Ejecución condicional: ejecuta un comando si el previo se ha ejecutado.
(...)	Agrupa comandos.
&	Ejecuta un comando en background.
#	Todos los caracteres siguientes son tomados como comentarios.
\$	Leemos el contenido de una variable.
\	Extensión de comandos que no caben en una sola línea.
,	Posibilita el eco de metacaracteres.

Metacaracteres

Un **metacarácter** es un carácter escrito que tiene un significado especial para el *shell* que se usa para conectar comandos, realizar búsquedas específicas, etc.

- * – Coincide con cualquier carácter cero o muchas veces.
- ? – Coincide con cualquier carácter una sola vez.
- [] – Coincide con un conjunto de caracteres .
- ! – Excluir este rango de caracteres.

Metacaracteres

Comando	file1.txt	file2.txt	file33.txt	fileA.txt	fileC.zip	fileB.txt	Output.pdf
ls file*	X	X	X	X	X	X	
ls file?.txt	X	X		X		X	
ls file[AC].???				X	X		
ls file[A-C].???				X	X	X	
ls *.*[!txt]					X		X

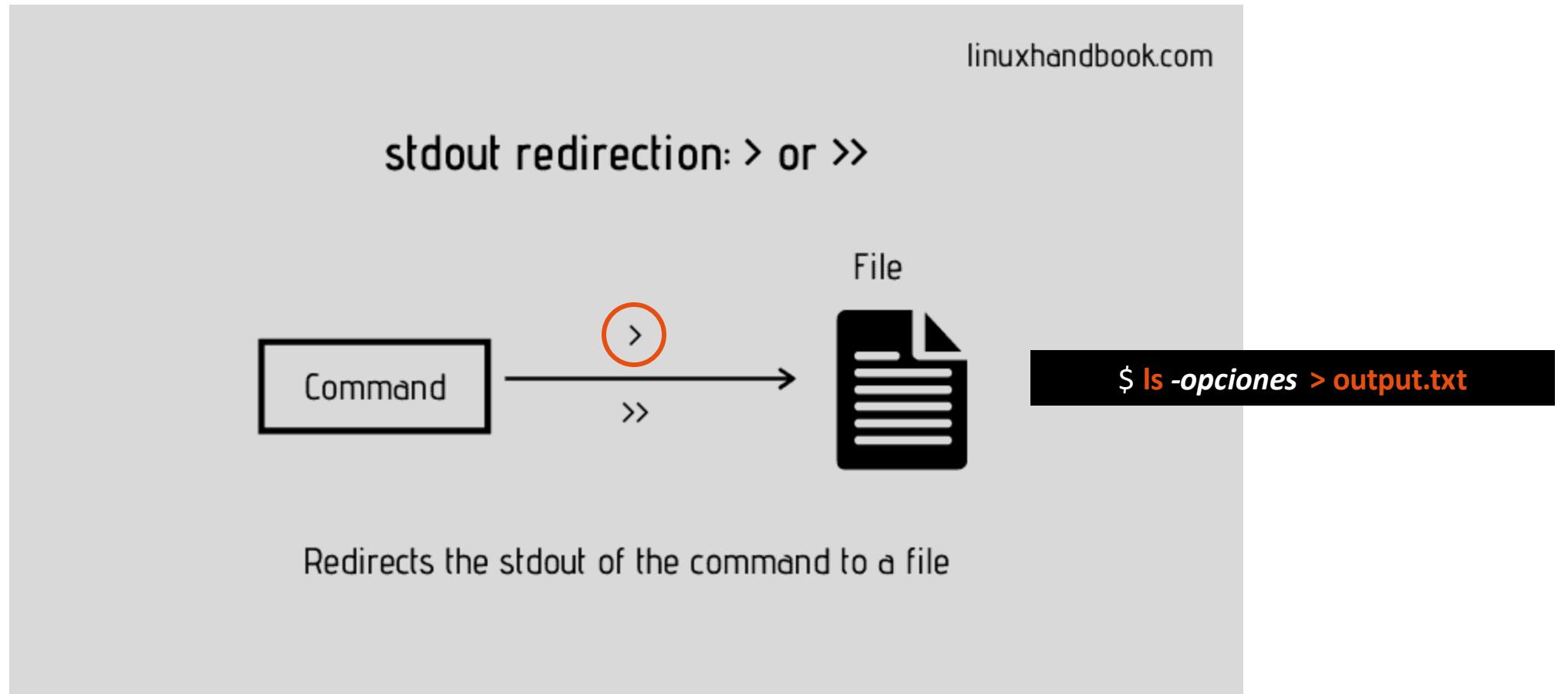
Metacaracteres

Un **metacarácter** es un carácter escrito que tiene un significado especial para el *shell* que se usa para conectar comandos, realizar búsquedas específicas, etc.

- **barra vertical o pipe (|)**
- **ampersand (&)**
- **punto y coma (;)**
- **signo menor que (<) y signo mayor que (>)**

Redirección de la salida estándar (redirección stdout)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ ls  
cat.txt comandos comandos.txt core.31938 core.4554 Desktop Documents Downloads human_genome_hg38 Music Pictures Public Scripts Templates Videos
```



Comando rm (remove)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat output.txt  
cat.txt  
comandos  
comandos.txt  
core.31938  
core.4554  
Desktop  
Documents  
Downloads  
human_genome_hg38  
Music  
output.txt  
Pictures  
Public  
Scripts  
Templates  
Videos  
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ rm output.txt
```

Comando rm

- ¿Elimina archivos vacíos o con contenido?
- ¿Podemos utilizarlo para eliminar un directorio?

```
> mkdir test  
> rm test/  
rm: cannot remove 'test/': Is a directory
```

```
> rmdir test/  
rmdir: failed to remove 'test/': Directory not empty
```

```
> rm -r test
```

Comando rm (remove)

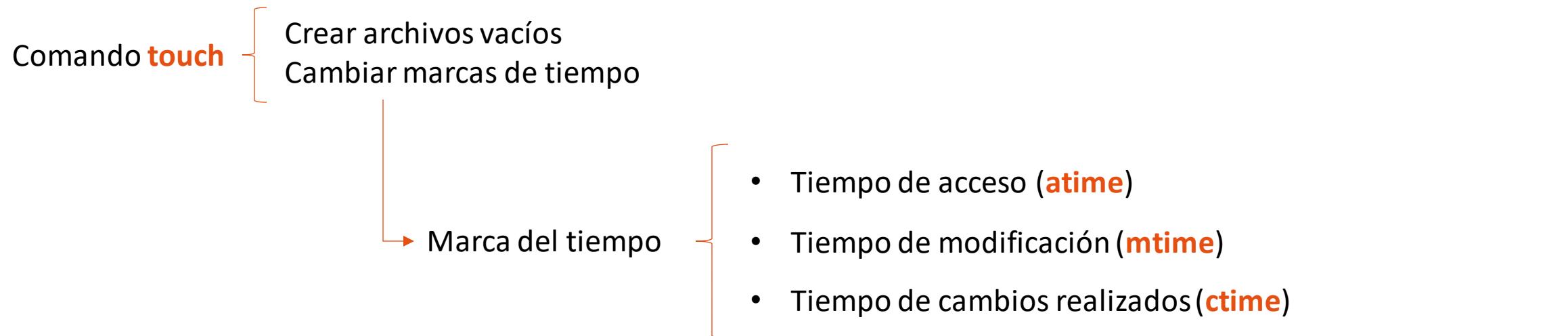
```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat output.txt  
cat.txt  
comandos  
comandos.txt  
core.31938  
core.4554  
Desktop  
Documents  
Downloads  
human_genome_hg38  
Music  
output.txt  
Pictures  
Public  
Scripts  
Templates  
Videos  
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ rm output.txt
```

Comando rm

- rm output1.txt output2.txt ...
- rm -i # confirmación para borrar

```
> rm -i output.txt  
rm: remove regular file 'output.txt'?
```

¿Y podemos crear el archivo de salida de antemano? -> Comando **touch**

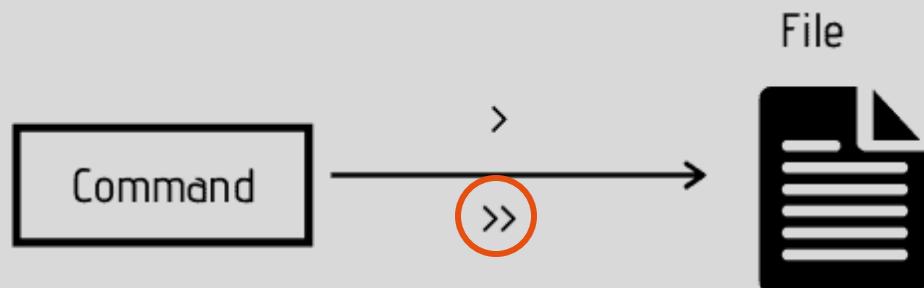


Sintaxis	Descripción
	Modificar hora de acceso y último cambio de un archivo
<code>touch <ruta></code>	<p>Actualiza la hora de acceso y de modificación a la hora actual. Si <ruta> no existe se crea vacío, a menos que se proporcionen las opciones -c o -h.</p> <p>Algunas opciones:</p> <ul style="list-style-type: none">-a cambia solo la hora de acceso-c no crea el archivo-d <s> usa <s> como hora actual-m cambia solo la hora de modificación

```
> touch output.txt  
  
> touch output1.txt output2.txt output3.txt  
> touch output{1..3}.txt  
  
> touch -am output.txt  
  
# CCYYMMDDhhmm.ss  
> touch -t 202210240957.00 output.txt
```

Redirección de la salida estándar (redirección stdout)

stdout redirection: > or >>



Redirects the stdout of the command to a file

Con una sola > estás sobreescibiendo el otro archivo

```
$ echo "Hello World!" > text  
$ cat text  
Hello World!
```

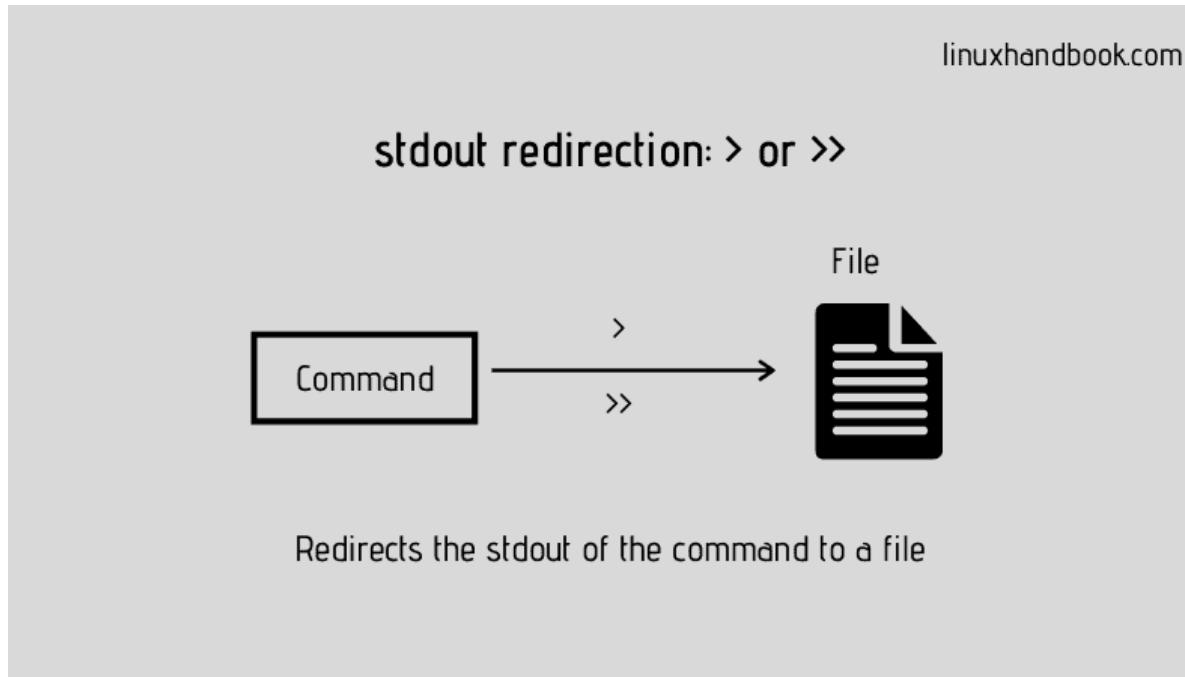
```
$ echo "Hello!" > text  
$ cat text  
Hello!
```

Si pones dos, >>, vas a agregar en lugar de sobreescribir

```
$ echo "Hello to you too!" >> text  
$ cat text  
Hello!  
Hello to you too!
```

Redirección del ERROR estándar (redirección stderr)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat fff  
cat: fff: No such file or directory
```



```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat fff 2> error.txt  
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat error.txt  
cat: fff: No such file or directory
```

Stdout = 1> = >
Stderr = 2>

Redirección del ERROR estándar (redirección stderr)

Concatenar la salida estándar con el error estándar Esto es para guardar los errores junto a su salida

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ cat fff > salida.txt 2> error.txt
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ cat salida.txt
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ cat error.txt
cat: fff: No such file or directory
```

Redirrección conjunta de la salida estándar con el error estándar

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ cat fff &> error.txt
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ cat error.txt
cat: fff: No such file or directory
```

Enviar el error a /dev/null

Esto es para hacer desaparecer el error, es un agujero negro, la info no se puede recuperar

La C del principio indica que es un archivo con carácter especial

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ ls -lh /dev/null
crw-rw-rw- 1 root root 1, 3 Oct 21 04:24 /dev/null
```

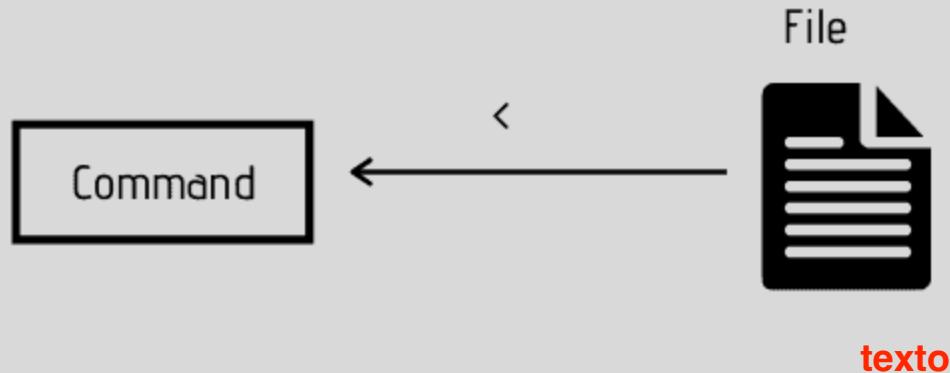
```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr~]$ cat fff 2> /dev/null
```

/dev/null es un dispositivo especial en sistemas operativos tipo Unix (Linux, macOS, etc.) que actúa como un "agujero negro". Todo lo que se envía a este archivo es descartado y desaparece, sin generar ningún error o salida.

Es para suprimir la salida estándar de un comando

Redirección de la entrada estándar (redirección stdin)

stdin redirection: <



Redirects the content of the file to stdin of the command

```
$ cat < text  
Hello!  
Hello to you too!
```

En la mayoría de archivos no se utiliza esta entrada estándar de < porque ya lo va a ejecutar de por si, EXCEPTO con el comando tr (translate), que si hay que usar <

<< (Here documents)

El operador << actúa de manera diferente

Esto es como que te deja escribir hasta que vuelves a poner la palabra que pones con los <<

```
$ cat << hello  
> hey  
> ola  
> hello  
hey  
ola
```



viu

**Universidad
Internacional
de Valencia**

universidadviu.com

De:
 Planeta Formación y Universidades