



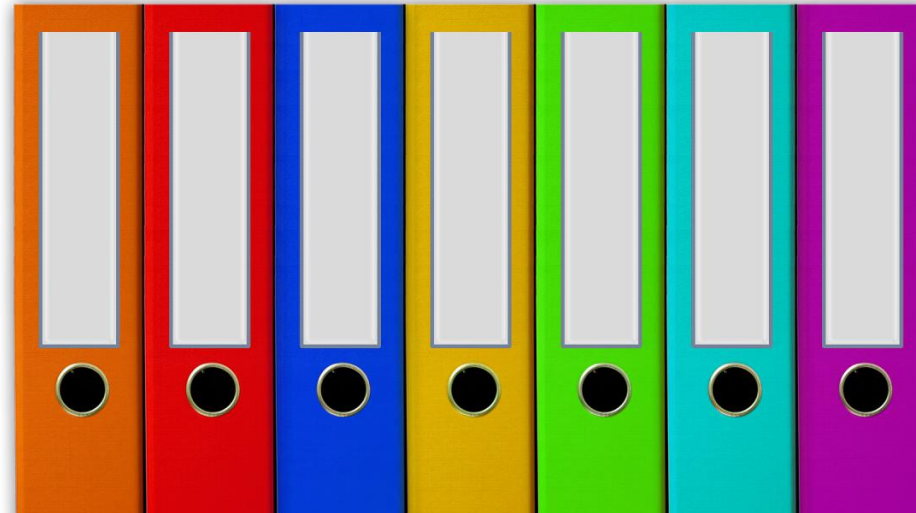
1º DAM/DAW Sistemas Informáticos

U2. Comandos en Linux.

Comandos. Contenido de ficheros

Contenido de ficheros

- Anteriormente, hemos visto comandos que nos permiten movernos por la estructura de directorios, así como gestionar el contenido de éstos (crear, copiar, mover y borrar). En este documento vamos a centrarnos en el uso de comandos para gestionar el **contenido de ficheros**.



Redirección >

- Los comandos tienen, por defecto, una **entrada** estándar (teclado) y una **salida** estándar (pantalla). Sin embargo, es útil y frecuente cambiar esta entrada estándar o esta salida estándar en función de lo que pretendamos obtener.
- Por ejemplo, imaginemos que en vez de mostrar el contenido de un directorio en pantalla, queremos redireccionarlo a un fichero para que se guarde directamente este salida en el fichero. La sintaxis del comando ls se completaría con el carácter ">" seguido del fichero donde queramos mostrar el resultado, por ejemplo salida.txt:

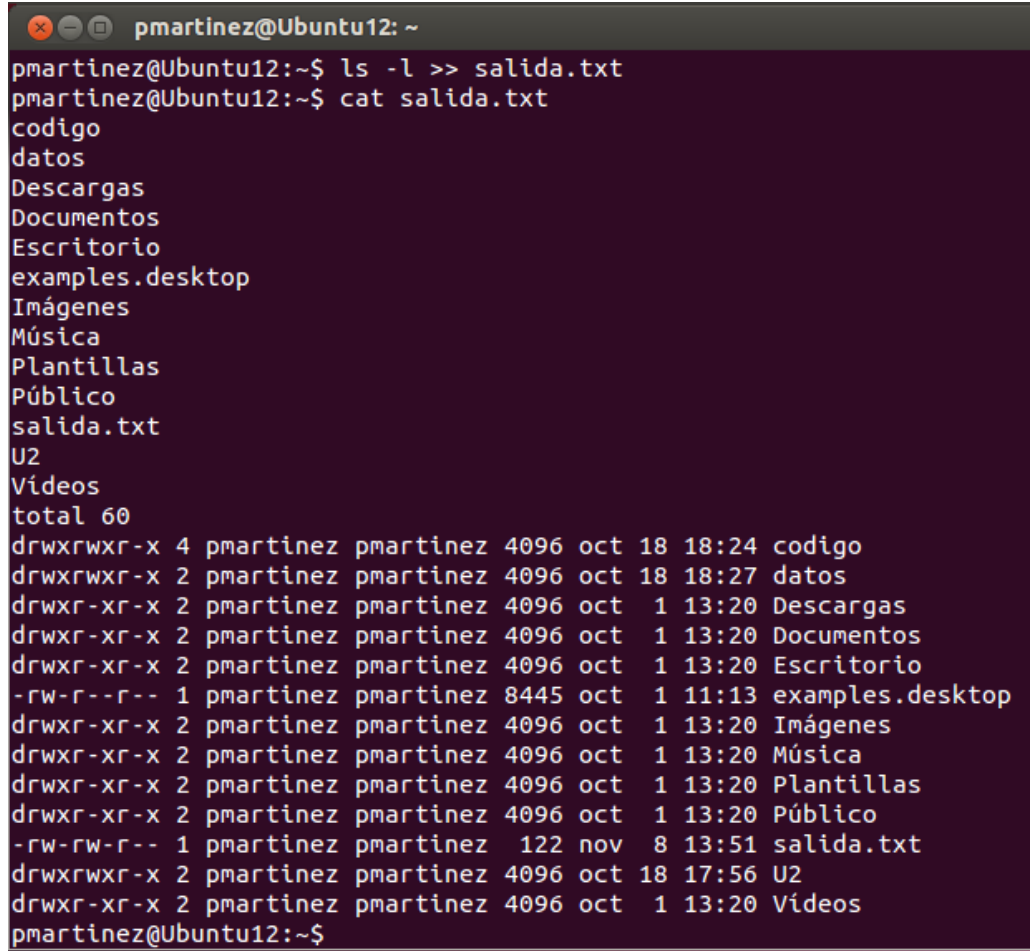
ls > salida.txt

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ ls  
codigo Descargas Escritorio Imágenes Plantillas U2  
datos Documentos examples.desktop Música Público Vídeos  
pmartinez@Ubuntu12:~$ ls > salida.txt  
pmartinez@Ubuntu12:~$ cat salida.txt  
codigo  
datos  
Descargas  
Documentos  
Escritorio  
examples.desktop  
Imágenes  
Música  
Plantillas  
Público  
salida.txt  
U2  
Vídeos  
pmartinez@Ubuntu12:~$
```

Redirección

- Hay que tener en cuenta que la redirección “>” sobrescribe (machaca, destruye, ...) el fichero destino, en caso de existir previamente. Si lo que queremos es añadir contenido al final de un fichero existente debemos utilizar el doble carácter “>>” :

ls >> salida.txt



A terminal window titled 'pmartinez@Ubuntu12: ~' showing the execution of two commands. The first command is 'ls -l >> salida.txt', which appends the output of the 'ls -l' command to the file 'salida.txt'. The second command is 'cat salida.txt', which displays the contents of 'salida.txt'. The output shows a list of files and directories in the home directory, including 'codigo', 'datos', 'Descargas', 'Documentos', 'Escritorio', 'examples.desktop', 'Imágenes', 'Música', 'Plantillas', 'Público', 'salida.txt', 'U2', and 'Videos'. The total size of the files is 60 KB. The permissions and ownership of each file are also listed.

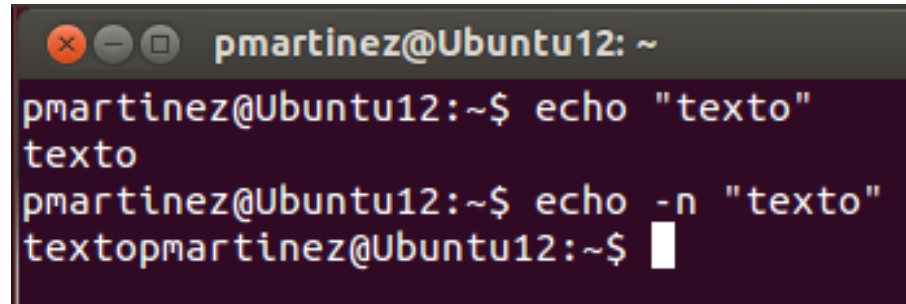
```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ ls -l >> salida.txt  
pmartinez@Ubuntu12:~$ cat salida.txt  
codigo  
datos  
Descargas  
Documentos  
Escritorio  
examples.desktop  
Imágenes  
Música  
Plantillas  
Público  
salida.txt  
U2  
Videos  
total 60  
drwxrwxr-x 4 pmartinez pmartinez 4096 oct 18 18:24 codigo  
drwxrwxr-x 2 pmartinez pmartinez 4096 oct 18 18:27 datos  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Descargas  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Documentos  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Escritorio  
-rw-r--r-- 1 pmartinez pmartinez 8445 oct 1 11:13 examples.desktop  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Imágenes  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Música  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Plantillas  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Público  
-rw-rw-r-- 1 pmartinez pmartinez 122 nov 8 13:51 salida.txt  
drwxrwxr-x 2 pmartinez pmartinez 4096 oct 18 17:56 U2  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Videos  
pmartinez@Ubuntu12:~$
```

Comandos echo, cat y wc

- Comando **echo**: muestra por la salida estándar, la cadena de texto que recibe como parámetro. Cuando pruebes este comando, observarás que introduce un salto de línea después del texto. Si quieres evitar esto en alguna circunstancia, utiliza el parámetro **-n**:

echo "texto"

echo -n "texto"

A terminal window titled 'pmartinez@Ubuntu12: ~' with standard window controls. It shows two commands being executed. The first is 'echo "texto"', which outputs 'texto' on a new line. The second is 'echo -n "texto"', which outputs 'texto' followed by a cursor on the same line.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ echo "texto"  
texto  
pmartinez@Ubuntu12:~$ echo -n "texto"  
textopmartinez@Ubuntu12:~$
```

Comandos echo, cat y wc

- Comando **cat**: muestra el contenido de uno o más ficheros en el terminal. Puedes probar a volcar una cadena de texto a un fichero mediante el comando “echo” y utilizar el comando “cat” para visualizar este fichero.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ echo "cadena de texto" > texto.txt  
pmartinez@Ubuntu12:~$ cat texto.txt  
cadena de texto  
pmartinez@Ubuntu12:~$
```

- Podemos visualizar el contenido de varios ficheros, poniendo sus nombres seguidos de un espacio.

cat salida.txt texto.txt

Comandos echo, cat y wc

- Comando **wc** (word count): cuenta las líneas, palabras o caracteres de un fichero. Para ello utilizamos los respectivos parámetros “-l”, “-w” y “-c”.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ wc -l salida.txt  
27 salida.txt  
pmartinez@Ubuntu12:~$ wc -w salida.txt  
132 salida.txt  
pmartinez@Ubuntu12:~$ wc -c salida.txt  
916 salida.txt  
pmartinez@Ubuntu12:~$ wc -l -w -c salida.txt  
27 132 916 salida.txt  
pmartinez@Ubuntu12:~$
```

Tuberías

- Una tubería se representa por “|” (AltGr + 1) y nos permite combinar varios comandos para ejecutarlos en un único mandato, donde la salida o resultado del primer comando se envía a la entrada del segundo, y así sucesivamente.
- Por ejemplo, si queremos saber cuántos ficheros (y directorios) tenemos en una carpeta, podemos listar el contenido de la ruta actual, de manera que en cada línea nos aparezca un fichero (ls -l), y después, contar cuántas líneas tiene la salida del primer comando, mediante un segundo comando (wc -l). Utilizando una tubería podríamos concatenar los dos comandos.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ ls -l | wc -l  
15  
pmartinez@Ubuntu12:~$ ls -l  
total 64  
drwxrwxr-x 4 pmartinez pmartinez 4096 oct 18 18:24 código  
drwxrwxr-x 2 pmartinez pmartinez 4096 oct 18 18:27 datos  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Descargas  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Documentos  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Escritorio  
-rw-r--r-- 1 pmartinez pmartinez 8445 oct 1 11:13 examples.desktop  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Imágenes  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Música  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Plantillas  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Público  
-rw-rw-r-- 1 pmartinez pmartinez 916 nov 8 14:02 salida.txt  
-rw-rw-r-- 1 pmartinez pmartinez 16 nov 8 17:38 texto.txt  
drwxrwxr-x 2 pmartinez pmartinez 4096 oct 18 17:56 U2  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Vídeos  
pmartinez@Ubuntu12:~$
```

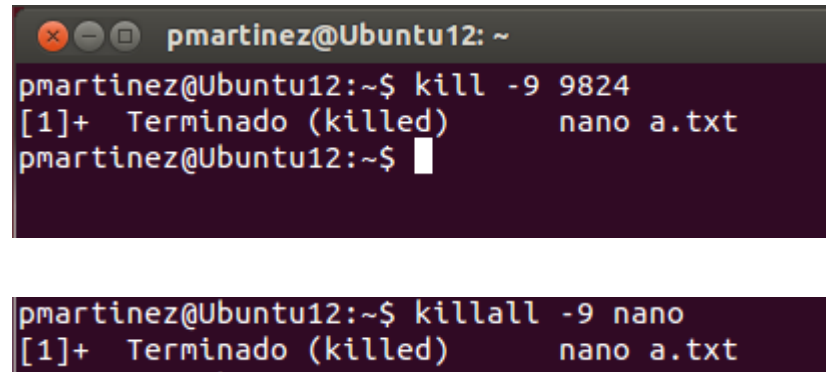

Comando ps aux

- Comando **ps**: un programa es un elemento estático representado por unas líneas de código que describen lo que hace. Por otro lado, un **proceso** es un elemento dinámico que representa una instancia de un programa en ejecución. En este sentido, el comando **ps** con la opción **aux**, nos muestra todos los procesos que hay ejecutándose en el sistema. Por ejemplo, si ejecutamos el editor “nano” y listamos los procesos, vemos que corresponde con el proceso con PID 9824.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ ps aux  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root         1  0.0  0.0   3664   2012 ?        Ss   13:38   0:01 /sbin/init  
root         2  0.0  0.0     0     0 ?        S    13:38   0:00 [kthreadd]  
root         3  0.0  0.0     0     0 ?        S    13:38   0:00 [ksoftirqd/0]  
root         5  0.0  0.0     0     0 ?        S<   13:38   0:00 [kworker/0:0H]  
root         6  0.0  0.0     0     0 ?        S    13:38   0:00 [kworker/u:0]  
root         7  0.0  0.0     0     0 ?        S<   13:38   0:00 [kworker/u:0H]  
root         8  0.0  0.0     0     0 ?        S    13:38   0:00 [migration/0]  
root         9  0.0  0.0     0     0 ?        S    13:38   0:00 [rcu_bh]  
root        10  0.0  0.0     0     0 ?        S    13:38   0:00 [rcu_sched]  
root       1000  0.0  0.0   6048   1248 pts/0    T    18:44   0:00 nano a.txt  
root       1000  0.0  0.0   6164   1152 pts/0    R+   18:46   0:00 ps aux
```

Comando kill y killall

- Podemos finalizar o matar un proceso que está en ejecución mediante el comando **kill**. Para ello, al ejecutar el comando `ps aux`, vemos que en la segunda columna hay un número que identifica a cada proceso (PID). Para finalizar un proceso concreto, usaremos el comando `kill` seguido de la opción “**-9**” (esta opción fuerza la terminación de un proceso, no es la más recomendable técnicamente, ya aprenderemos porqué, pero por ahora nos vale...) y del “**PID**” que queramos finalizar. También podríamos finalizar todos los programas de un tipo, mediante la orden **killall** seguida del nombre del proceso.



```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ kill -9 9824  
[1]+  Terminado (killed)      nano a.txt  
pmartinez@Ubuntu12:~$  
  
pmartinez@Ubuntu12:~$ killall -9 nano  
[1]+  Terminado (killed)      nano a.txt
```

Comando grep

- Comando **grep**: el comando grep nos permite hacer una búsqueda de cadenas de caracteres. Este comando es muy versátil y tiene multitud de utilidades. A modo de ejemplo, si generamos un fichero con una serie de nombres, podemos filtrar aquellas líneas que contengan una cadena concreta. Este comando se suele utilizar combinado mediante una tubería con otros comandos, como cat.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ cat > nombres.txt  
Manuel  
Antonio  
Juan Carlos  
Jose Manuel  
Juan Pedro  
Jose Miguel  
Fernando
```

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ cat nombres.txt | grep "Juan"  
Juan Carlos  
Juan Pedro  
pmartinez@Ubuntu12:~$ cat nombres.txt | grep "Jose"  
Jose Manuel  
Jose Miguel  
pmartinez@Ubuntu12:~$ cat nombres.txt | grep "Manuel"  
Manuel  
Jose Manuel  
pmartinez@Ubuntu12:~$
```

Comando grep

- Sin embargo, también podemos utilizar este comando sin tubería. Por ejemplo, si queremos encontrar todas las cadenas que aparezcan en los ficheros de nuestro directorio actual, podemos hacerlo. Esto nos mostraría los ficheros que contienen las cadenas buscadas.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ grep "Jose" ./*  
./nombres.txt:Jose Manuel  
./nombres.txt:Jose Miguel  
pmartinez@Ubuntu12:~$ grep "Fer*" ./*  
./nombres.txt:Fernando  
pmartinez@Ubuntu12:~$
```

- Si además quisiéramos buscar dentro de los subdirectorios, utilizaremos el parámetro **-R**

Comando find

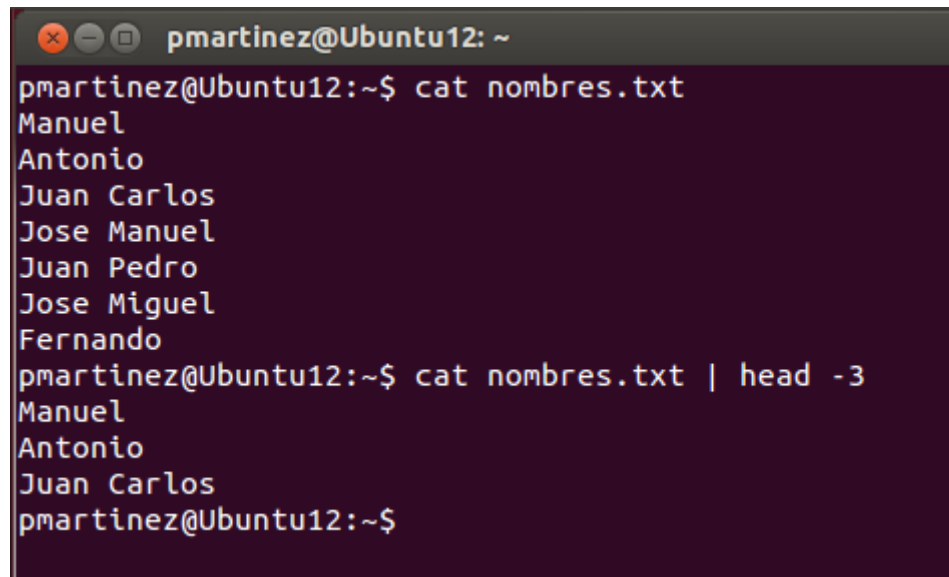
- Comando **find**: nos permite buscar ficheros por su nombre. En su uso más elemental, especificamos la ruta dónde queremos buscar (recordad que un “.” indica el directorio actual), seguido del parámetro “-name” para indicar el nombre a buscar. El resultado de la búsqueda son los directorios donde está el fichero buscado. Si no recordamos el nombre completo, podemos utilizar el carácter “*”, para que nos encuentre todos los ficheros que cumplan un patrón.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ find . -name "*.txt"  
./config/libreoffice/3/user/uno_packages/cache/log.txt  
./codigo/prueba/fich2.txt  
./codigo/real/fich3.txt  
./codigo/real/fich1.txt  
./nombres.txt  
./salida.txt  
./mozilla/firefox/wf3zmn8f.default/urlclassifierkey3.txt  
./texto.txt  
./a.txt  
./datos/fich1.txt  
pmartinez@Ubuntu12:~$
```

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ find /home/pmartinez -name "*.txt"  
/home/pmartinez/.config/libreoffice/3/user/uno_packages/cache/log.txt  
/home/pmartinez/codigo/prueba/fich2.txt  
/home/pmartinez/codigo/real/fich3.txt  
/home/pmartinez/codigo/real/fich1.txt  
/home/pmartinez/nombres.txt  
/home/pmartinez/salida.txt  
/home/pmartinez/.mozilla/firefox/wf3zmn8f.default/urlclassifierkey3.txt  
/home/pmartinez/texto.txt  
/home/pmartinez/a.txt  
/home/pmartinez/datos/fich1.txt  
pmartinez@Ubuntu12:~$
```

Comandos head y tail

- Comando **head**: acompañado de un número natural como parámetro “-n”, nos muestra las **n primeras** líneas de un fichero. Es muy útil usar este comando en combinación con otros, mediante tuberías.



```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ cat nombres.txt  
Manuel  
Antonio  
Juan Carlos  
Jose Manuel  
Juan Pedro  
Jose Miguel  
Fernando  
pmartinez@Ubuntu12:~$ cat nombres.txt | head -3  
Manuel  
Antonio  
Juan Carlos  
pmartinez@Ubuntu12:~$
```

Comandos head y tail

- Comando **tail**: de forma similar al comando anterior, el comando **tail** nos muestra las **n últimas** líneas de un fichero. Es habitual combinar el uso de head y tail.

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ cat nombres.txt  
Manuel  
Antonio  
Juan Carlos  
Jose Manuel  
Juan Pedro  
Jose Miguel  
Fernando  
pmartinez@Ubuntu12:~$ cat nombres.txt | tail -2  
Jose Miguel  
Fernando  
pmartinez@Ubuntu12:~$
```

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ cat nombres.txt  
Manuel  
Antonio  
Juan Carlos  
Jose Manuel  
Juan Pedro  
Jose Miguel  
Fernando  
pmartinez@Ubuntu12:~$ cat nombres.txt | head -3 | tail -1  
Juan Carlos  
pmartinez@Ubuntu12:~$
```