

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

## ÍNDICE

Comunicación entre procesos y utilización de filtros.....	2
Ejercicio 1.....	2
Ejercicio 2.....	2
Ejercicio 3.....	2
Ejercicio 4.....	3
Ejercicio 5.....	3
Ejercicio 6.....	3
Ejercicio 7.....	3
Ejercicio 8.....	3
Ejercicio 9.....	3
Ejercicio 10.....	3
Ejercicio 11.....	4
Ejercicio 12.....	5
Ejercicio 13.....	5
Ejercicio 14.....	5
Ejercicio 15.....	6
Ejercicio 16.....	6
Ejercicio 17.....	6
Ejercicio 18.....	6
Ejercicio 19.....	6
Ejercicio 20.....	6
Ejercicio 21.....	7
Ejercicio 22.....	7
Ejercicio 23.....	7
Ejercicio 24.....	7
Ejercicio 25.....	8

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

En esta práctica verás cómo podemos hacer que los procesos que utilizamos se comuniquen unos con otros, mediante la utilización de flujos. Por otra parte, aplicaréis los conocimientos teóricos sobre filtros de la unidad.

## Comunicación entre procesos y utilización de filtros

En esta parte de la práctica os remitimos a las explicaciones teóricas de este núcleo de actividad para que resuelva las actividades que se le presentan. Estas actividades no están necesariamente en el mismo orden que las explicaciones.

### Ejercicio 1

**Respecto al paginador less, indique cómo se pueden llevar a cabo las acciones indicadas en la tabla de abajo:**

Paginador <i>less</i>	
Avanzar página a página	AVPÁG
Retroceder una página	REPÁG
Avanzar línea a línea	↓
Retroceder una línea	↑
Busca una palabra hacia delante	/
Busca una palabra hacia atrás	?
Ir a una línea específica	n
Salir	q

### Ejercicio 2

**Escribe un comando que muestre SÓLO los nombres de los usuarios que tienen un nombre que empieza por m. Es necesario que trabaje con el fichero */etc/passwd*.**

```
more /etc/passwd | grep ^m | cut -d: -f1
```

### Ejercicio 3

**¿Qué hace el comando *ls -l | tee primero.txt | sort -r > segundo.txt*?**

Este comando te crea dos archivos en el directorio donde te encuentres, el archivo **primero.txt** (el cual contiene el listado de archivos y directorios de donde nos encontremos) y **segundo.txt** (que contiene lo mismo, pero ordenado a la inversa). **¿Cuál es la diferencia con el comando *ls -l | tee primero.txt | sort -R > segundo.txt*?** La diferencia es la ordenación del

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

contenido del segundo archivo (**segundo.txt**), su orden es al azar. Es necesario que explique, paso a paso, que hace cada uno de los comandos.

#### Ejercicio 4

Escribe un comando que muestre el nombre del segundo archivo más grande del directorio */etc*.

#### Ejercicio 5

Escribe un comando que una dos archivos de texto, uno a continuación del otro, en un tercero.

```
paste primero.txt segundo.txt > tercero.txt
```

#### Ejercicio 6

¿Cómo podríamos copiar un archivo sin utilizar el comando *cp*, utilizando las redirecciones de la entrada y salida estándar? Copie el archivo */etc/passwd* en su directorio de usuario.

```
cat etc/passwd > /home/usuario/directorio/passwd.txt
```

#### Ejercicio 7

¿Cómo podríamos construir una pipe o tubería (*|*) con los redirectores de la entrada y salida estándar (*<>*)? Hay que utilizar un archivo temporal que debemos borrar después de su uso. Escriba el equivalente a *ls -l | wc -l*.

#### Ejercicio 8

Escribe un comando basado en *cat* que tenga el mismo comportamiento que *nl*.

```
cat -n
```

#### Ejercicio 9

Escribe un comando que muestre un listado de todos los usuarios del sistema, ordenados por orden alfabético.

```
more etc/passwd | cut -d: -f1 | sort
```

#### Ejercicio 10

Intente crear un directorio en la raíz de su sistema (*/*) y envíe la salida de error de este comando a un fichero de su directorio de usuario.

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

**¿Qué ocurre?** El fichero **ficher\_error.txt** contiene el siguiente mensaje de error:

*mkdir: no se ha podido crear el directorio «directorio»: Se ha denegado el permiso*

```
mkdir directorio 2> /home/usuari/fichero_error.txt
```

**Intente lo mismo con la salida estándar, ¿qué pasa?** A través de la salida estándar no nos dejará ejecutar este comando.

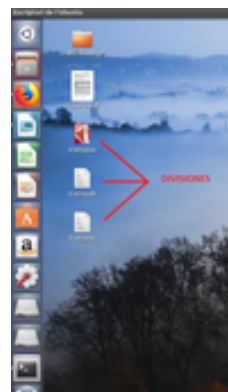
```
mkdir directorio > /home/usuari/fichero_error.txt
```

**¿Cómo podríamos hacer que esta salida de error NO se muestre por pantalla ni se guarde en ningún archivo?** Es necesario que trabaje con el fichero **/etc/passwd**.

## Ejercicio 11

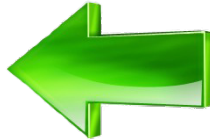
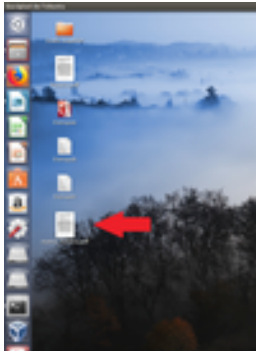
**Tome uno de los ficheros de TEORÍA (formato pdf) del crédito y subdividirlo en partes de 100KB cada una, haciendo que el prefijo de estas partes sea transp.**

```
split -b 100KB fichero.pdf transp
```



**Ejecute un único comando que reunifican el archivo en un nuevo pdf y compruebe que no ha habido ningún problema cuando la hemos vuelto a unir.**

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		



```
cat transp* > nuevo_fichero.pdf
```

## Ejercicio 12

Ejecute un comando que os muestre sólo las líneas repetidas, de forma consecutiva, del archivo `give_it_away.txt`.

```
uniq give_it_away.txt
```

Ejecute un comando que muestre el número de repeticiones de cada una de las líneas del archivo `give_it_away.txt`.

```
uniq -c give_it_away.txt
```

## Ejercicio 13

Utilice el comando `paste` para unir tres ficheros. Haga que todos los tabuladores devueltos se conviertan en espacios. Haga toda esta interacción en una única ejecución.

```
paste fichero1.txt fichero2.txt fichero3.txt
```

## Ejercicio 14

¿Qué opción del comando `tr` permite comprimir caracteres repetidos consecutivamente en uno solo. `tr -c -s ""`. Pruebe este comando con las "o" de la siguiente cadena "boomboompooooowoooooot".

```
sergjine@sergjine-VirtualBox:~$ echo 'boomboompooooowoooooot' | tr -c -s ""
bombompowot
```

```
echo 'boomboompooooowoooooot' | tr -c -s ""
```

¿Cómo lo haríamos si en vez de querer comprimir caracteres repetidos lo hiciéramos para comprimir espacios consecutivos (comprima pues los espacios de una salida del comando `'ls -l'`)?

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

```
sergjime@sergjime-VirtualBox:~$ ls -l | tr -d ' '
total 44
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Descargas
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Documentos
drwxr-xr-x 3 sergjime sergjime 4096 feb 6 23:37 Escritorio
-rw-r--r-- 1 sergjime sergjime 8980 feb 5 12:53 examples.desktop
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Imágenes
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Música
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Plantillas
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Público
drwxr-xr-x 2 sergjime sergjime 4096 feb 5 12:59 Videos
```

*ls -l | tr -d ' '*

## Ejercicio 15

Escribe un comando que, para cada uno de los directorios de la raíz del sistema, cree un archivo de texto con su nombre en el interior (en formato texto). Los archivos de texto generados deberán comenzar con el prefijo 'carpeta\_'.

## Ejercicio 16

Escribe una línea para cada apartado el cual os permita:

- Crear un directorio y, si se ha podido crear correctamente, mostrar el mensaje "directorio creado correctamente".
- Crear un directorio y, si no se ha podido crear correctamente, mostrar el mensaje "no se ha podido crear el directorio".

Para comprobar el funcionamiento de estos apartados intente crear carpetas en su directorio de usuario y / o en la raíz del sistema.

## Ejercicio 17

¿Qué efecto tiene el comando `echo "-----" | cat archivo1 - archivo2?`  
¿Qué efecto tiene el comando `cat archivo1 - archivo2 <archivo3?`

## Ejercicio 18

Los ficheros de configuración suelen indicar los comentarios con líneas que comienzan con #. Escribe un comando que muestre todas las líneas de comentarios del archivo `grub.cfg` (indica la ruta completa). Escribe un comando que muestre el contenido del archivo `grub.cfg` eliminando los comentarios y también las líneas vacías. Una línea vacía es aquella que no contiene ningún carácter.

## Ejercicio 19

El comando `tail` dispone de una opción que permite ir mostrando todas

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

las líneas que se vayan añadiendo a un fichero sin cerrar este archivo. ¿Cuál es? En un terminal, cree un archivo. Desde otro terminal, haga el seguimiento línea en línea de este archivo. Al terminal inicial, vaya modificando y guardando el archivo. Explique qué ocurre. ¿Para que cree que puede ser útil esta opción?

## Ejercicio 20

Tome el archivo de texto *give\_it\_away.txt* y realice las acciones siguientes (en el mismo orden, comunicadas con tuberías o pipes):

1. Elimine las líneas repetidas consecutivas.
2. Cambie todas las letras mayúsculas que haya por minúsculas.
3. Elimine todas las líneas vacías que haya.
4. Ordene las líneas resultados en orden alfabético inverso.
5. Numerad todas las líneas.
6. Quédese sólo con las líneas que van de la 10 a la 40.
7. Eliminar la numeración de las líneas que habíais añadido en el paso 5.
8. Reordenar las líneas en orden alfabético por la segunda palabra de cada línea.

Explica cada uno de los pasos utilizados en este archivo.

## Ejercicio 21

Escribe un comando que devuelva la MAC de nuestra tarjeta de red. Para hacerlo, es necesario que:

1. Ejecute el comando *ifconfig*.
2. Seleccione de alguna manera la línea que contiene la MAC de nuestra tarjeta de red.
3. Comprime todos los espacios en blanco en uno solo.
4. Seleccione la columna correspondiente a la MAC.

Explica cada uno de los pasos utilizados en este comando.

## Ejercicio 22

Escribe un comando que muestre los PID de los 5 procesos que consumen más memoria del sistema.

## Ejercicio 23

Escribe un comando que permita mostrar por pantalla el siguiente mensaje: **\*\*\*MICRO\$OFT RULE\$\*\*\***.

<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b>	Práctica 11 - Procesos (PIPES)		

## Ejercicio 24

¿Para qué sirve cada una de las expresiones regulares de *grep* que se indican a continuación?

Comando	Resultado
<code>^</code>	
<code>\$</code>	
<code>[...]</code>	
<code>[^...]</code>	
<code>[m-r]</code>	
<code>.</code>	
<code>c*</code>	
<code>.*</code>	
<code>\{n\}</code>	
<code>\{n,\}</code>	
<code>\{n,m\}</code>	
<code>\</code>	

## Ejercicio 25

Explica qué hace cada uno de los comandos siguientes:

Comando	Resultado
<code>grep give *</code>	Busca líneas de texto que contengan una cadena o patrón dentro de uno o varios archivos y las las imprime.
<code>grep -l give *</code>	Lista los archivos en donde se han encontrado coincidencias sin mostrar el texto.
<code>grep -i give *.txt</code>	Busca la palabra give ignorando la diferencia entre mayúsculas y minúsculas.
<code>grep -i -h give *.txt</code>	
<code>grep -n give give_it_away.txt</code>	Muestra el texto de la línea y el número de línea de give dentro del archivo.



<b>Nombre:</b>	Sergio	<b>Apellidos:</b>	Jiménez Sastre
<b>Asignatura:</b>	Sistemas informáticos	<b>Curso y Grupo:</b>	DAW1A
<b>Práctica:</b> Práctica 11 - Procesos (PIPES)			

grep -cv give give_it_away.txt	
grep give\$ give_it_away.txt	
grep ^give\$ *	
grep [A-Z] give_it_away.txt	
grep 'Sta\{1,2\}man' *	
grep '^error' *.log > problemas.txt	
grep '^dump' *.log >> problemas.txt	
grep -c /bin/bash /etc/passwd	