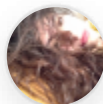


# WUOLAH



emewinchester

[www.wuolah.com/student/emewinchester](http://www.wuolah.com/student/emewinchester)



362

## Practica 3 APSO resuelta.pdf

*PRÁCTICAS RESUELTAS (incluye teoría)*



2º Administración y Programación de Sistemas Operativos



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería  
UHU - Universidad de Huelva

# 5

**CONSIGUE UN 5  
PELAO EN INGLÉS  
Y GANA UN VIAJE  
A NUEVA YORK**

NO TE LIES  
**VIAJA**

**MY  
ENGLISH  
COOL**  
EASIER,  
CHEAPER...  
BETTER

**PARTICIPA Y GANA**

# APSO: PRÁCTICA 3

---

## MANEJO DE LA SHELL (III)

Marina Delgado Pérez  
GRADO INGENIERÍA INFORMÁTICA | HUELVA

# 2x1

*carné universitario*



# FOSTER'S HOLLYWOOD

\*Consulta las condiciones de la promoción en [fostershollywood2x1universitario.com](http://fostershollywood2x1universitario.com)

## Contenido

1.- REDIRECCIÓN .....	2
2.- TUBERÍAS.....	4
3.- SUSTITUCIÓN DE COMANDOS .....	4
4.- EJECUCIÓN EN SEGUNDO PLANO .....	5

# PRELUDIO TEÓRICO

## 1.- REDIRECCIÓN

Para entender la redirección, debemos saber que todos los procesos tienen una **tabla de canales**.

Una tabla de canales está formada por una serie de posiciones, cada una enumerada con un número entero. Cada proceso tiene una tabla de canales propia.

TABLA DE CANALES (T.C.)

0		ENTRADA ESTÁNDAR
1		SALIDA ESTÁNDAR
2		SALIDA ESTÁNDAR DE ERROR
3		
4		

Por ahora, solo tendremos en cuenta las 3 primeras posiciones de la tabla.

Todo proceso tiene su tabla de canales, y todos los procesos hijos de otro proceso heredan la tabla de canales del proceso padre.

En una máquina UNIX, el proceso existente inicialmente es el proceso *bash*. A continuación, la tabla de canales del proceso:

TABLA DE CANALES (T.C.)

0	TECLADO
1	PANTALLA
2	PANTALLA
3	
4	

Entrada estándar (teclado): recoge información por el teclado.

Salida estándar (pantalla): muestra la información resultante por pantalla.

Salida estándar de error (pantalla): idem, pero mostrando mensajes de error.

# 5

SACA UN 5 PELAO  
EN NUESTRO TEST  
DE INGLÉS



Y GANA  
UN VIAJE A  
NUEVA YORK

PARTICIPA Y GANA

## Administración y Programación de Sistemas Operativos

Si desde el proceso *bash* lanzamos el proceso *cat*, la tabla de canales del proceso *cat* será la misma que la del proceso *bash*.

**OBJETIVO:** Modificar las entradas y salidas estándar al lanzar un proceso hijo.

Sintaxis:

- < Modifica entrada estándar
- > Modifica salida estándar
- 2> Modifica salida estándar de error (2: posición tabla TC)
- >> Redirecciona la salida, pero si el fichero ya existe, no lo machaca, comienza a escribir por el final del fichero.

### EJEMPLOS:

- 1) Tomamos el comando *cat* (todo lo que le llega por la entrada lo muestra por pantalla, su salida estándar)

`cat >f3.txt`

La entrada se produce por teclado y la salida por un fichero llamado *f3.txt*. Si este fichero existe, se sobrescribe. Si no existe, se crea. Control + d para guardar y salir.

`cat <f1.txt >f3.txt`

Copia y pega *f1.txt* con nombre *f3.txt*

- 2) Al hacer una búsqueda por todo el sistema (*find /*), obtendremos mensajes de error por pantalla cuando intentemos acceder a directorios a los que no tengamos permisos.

`find / >errores.txt`

Los errores no salen por pantalla, sino que son escritos en un fichero.

`find / >/dev/null`

Los mensajes de error van a la papelera (ruta de la papelera: */dev/null*). Todo lo que llega a la papelera se elimina al momento.

NO TE LIES  
VIAJA

PARTICIPA  
AQUÍ



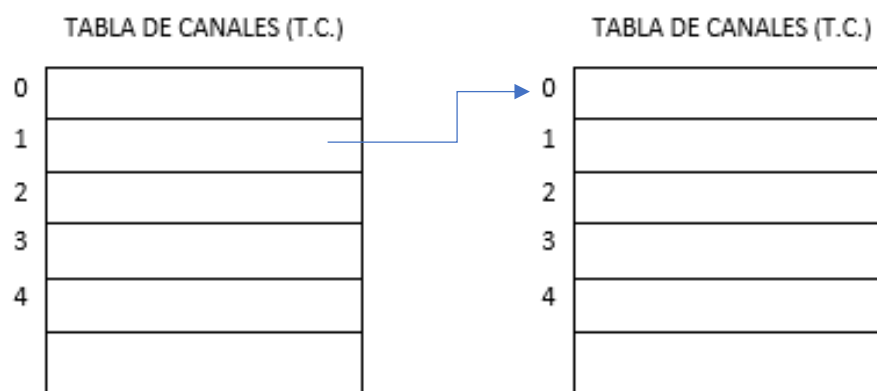
notelies.  
myenglishcool.  
com

MY  
ENGLISH  
COOL  
EASIER...  
CHEAPER...  
BETTER



## 2.- TUBERÍAS

**DEFINICIÓN:** La salida del comando 1 se convierte en la entrada del comando 2.



**EJEMPLO:**

1)

```
cd /home
ls
ls | more
```

Al ejecutar `ls` se muestra por pantalla el contenido de `/home`. Sin embargo, al ejecutar `ls | more` se muestra línea a línea el contenido de `/home`. Se pierden los colores propios del comando `ls` porque la salida de `ls` pasa a ser la entrada de `more` y este comando solo recibe caracteres.

2) `who | wc -l`

Muestra por pantalla el número de sesiones activas.

Podemos enlazar todas las tuberías que queramos (usualmente haremos un máximo de 3 comandos enlazados por tuberías).

**Importante:** las tuberías son muy comunes a la hora de contabilizar (comando `wc`).

## 3.- SUSTITUCIÓN DE COMANDOS

**DEFINICIÓN:** La(s) salida(s) del comando1 se convierte(n) en el(los) parámetro(s) del comando2.

**SINTAXIS:** `comando_2 $(comando_1)`

La sustitución de comandos suele emplearse para **copiar, mover y borrar ficheros**, entre otras funciones.

**EJEMPLO:** (ejecutar desde /home/so/ficheros)

`ls | more`

Muestra línea a línea el contenido de la carpeta.

`more $(ls)`

Entra en todos los ficheros u los muestra línea a línea

**¡OJO!** Las tuberías y las sustituciones son difíciles de diferenciar, pero a su vez producen resultados muy distintos. Por ello, es importante reconocer bien cuándo hacer una técnica u otra.

#### 4.- EJECUCIÓN EN SEGUNDO PLANO

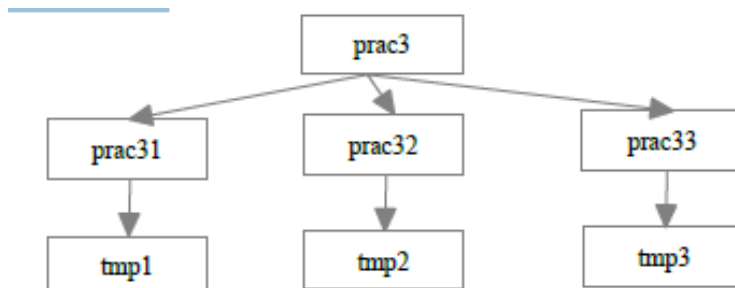
Al lanzar un comando, el prompt no vuelve a salir hasta que termine de ejecutarse el proceso generado por el comando.

Para poder llevar a cabo otros procesos mientras un proceso largo se finaliza, podemos poner el proceso largo en segundo plano, apareciendo así el prompt por pantalla y permitiéndonos seguir ejecutando comandos. La manera de hacerlo es añadir **&** detrás del comando que vaya a generar el proceso que queremos dejar en segundo plano.



## RESOLUCIÓN PRÁCTICA 3

1. Cree con un solo comando el siguiente árbol de directorios a partir de su directorio personal



```
mkdir prac3 prac3/prac31 prac3/prac31/temp1 prac3/prac32
prac3/prac32/temp2 prac3/prac33 prac3/prac33/temp3
```

2. Quite todo tipo de privilegios para el grupo y el resto de los grupos a los directorios `prac3`, `prac31`, `prac32` y `prac33` usando rutas relativas. El propietario deberá tener todos los permisos sobre esos directorios.

```
chmod go-rx prac3 prac3/prac3[123]
```

3. Visualice todos los ficheros y directorios de su directorio personal (incluso los ocultos), con los permisos establecidos sobre cada uno de ellos.  
(NOTA: los ficheros ocultos son aquellos que comienzan por punto (.))

```
ls -l -a
```

4. Con el comando `find`, el `wc` y las tuberías cuente cuántos directorios tiene a partir de su directorio personal que tengan una `m` o un número en su nombre.

```
find . -type d -iname "[m0-9]*" | wc -l
```

5. Busque, a partir del directorio `/bin`, todos los ficheros (no directorios) que comiencen por vocal y se hayan modificado hace más de 1 día. La salida de ese comando la debe enviar a un fichero llamado `comandos` que se guardará en el directorio `tmp1` de `prac31`. No deben aparecer los errores por pantalla. Hágalo con sólo una línea de comando.

```
find >prac3/prac31/tmp1/comandos 2>/dev/null /bin ! -type
d -iname "[aeiou]*" -mtime +1
```



Administración y Programación de Sistemas Operativos

6. Encuentre todos los ficheros del sistema (del tipo que sean: normales, directorios, enlaces, de dispositivo, etc....) cuyo primer carácter sea una vocal o un número, la segunda no sea un número, termine en .c o .p y su nombre tenga sólo 6 caracteres (contando el punto y la extensión). La salida debe ser visualizada página a página y los errores no deben mostrarse por pantalla.

```
find / 2>/dev/null -iname "[aeiou0-9][!0-9]??.[cp]" | more
```

7. Copie todos los ficheros que tenga a partir de su directorio personal que empiecen con el texto sol y tengan una p en su nombre en el directorio tmp2 del directorio prac32. Use para ello sustitución de comandos.

```
cp $(find . -iname "sol*[p]*") prac3/prac32/tmp2
```

8. Muévase al directorio prac33. Mueva todos los ficheros que a partir de su directorio personal terminan en ~ al directorio tmp3. Borre del directorio tmp3 todos los ficheros que terminan en ~. (Asegúrese antes que sólo tiene ficheros que terminan en ese carácter - Alt+126).

```
mv $(find ../../ -iname "*[~]") tmp3  
ls tmp3  
rm tmp3/*[~]
```

9. Muévase al directorio prac3 con rutas relativas. Cree con el comando cal, y tras ejecutarlo 4 veces, un fichero llamado 2\_cuatrimestre en el directorio prac32 con el calendario de los meses de marzo, abril, mayo y junio de este año.

```
cd ..  
cal 3 2019 | cat >prac32/2_cuatrimestre  
cal 4 2019 | cat >>prac32/2_cuatrimestre  
cal 5 2019 | cat >>prac32/2_cuatrimestre  
cal 6 2019 | cat >>prac32/2_cuatrimestre
```

10. Introduzca la siguiente lista por teclado mediante el comando sort de forma que al final obtenga un fichero llamado ciudades en el directorio prac32 con la lista ordenada (en orden descendente) de lo introducido por teclado:

cordoba  
sevilla  
huelva  
almeria  
cadiz  
malaga  
jaen  
granada

```
cat | sort >prac32/ciudades
```

11. Cree un fichero llamado `usuarios` en el directorio `prac32` con la lista de todos los directorios personales de los usuarios del sistema, ordenada alfabéticamente en orden ascendente y con una sola línea de comando (Para ello use el `find` con las opciones oportunas, el `sort`, tuberías y redirección) (Suponga que cada directorio del directorio `/home` es de un usuario del sistema) (Deben salir sólo los directorios y sólo su nombre -no la ruta- (se consigue con la opción `-printf`))(Cada directorio estará en una línea del fichero)(No hay que adentrarse en las carpetas de cada usuario (opción `-maxdepth`)).

```
find /home -maxdepth 1 -type d -iname "*" -printf "\n" -  
printf %f | sort >prac32/usuarios
```

12. Cree en el directorio `prac32` un fichero llamado `personal` que contenga 4 líneas con la siguiente información: nombre de usuario, identificador de usuario, nombre de grupo, identificador de grupo. No está permitido usar ningún editor de textos para crear este fichero. (Hay que ejecutar 4 veces el comando `id`).

```
id -n -u | cat >prac32/personal  
id -u | cat >>prac32/personal  
id -n -g | cat >>prac32/personal  
id -g | cat >>prac32/personal
```