



---

## **UNITAT DIDACTICA 3**

# **SISTEMA OPERATIU LINUX. ADMINISTRACIÓ I CONFIGURACIÓ**

## **Practica 2**

**Prof. Manuel Enguidanos**



---

# **UNITAT DIDACTICA 3**

## **REDIRECCIONAMENT I TUBERIES**

# Practica 2

Prof. Manuel Enguidanos

# 1. INTRODUCCIÓN

---

- ❑ En los sistemas GNU/LINUX y UNIX los dispositivos como los discos duros y sus particiones, el terminal de pantalla, el teclado, el ratón o incluso la tarjeta de audio están mapeados en el sistema de archivos.
  - ❑ Esta característica permite a los programas interactuar con los dispositivos de la misma manera que los archivos, simplificando su interfase de acceso.
  - ❑ Los dispositivos que mapean las operaciones de E/S hacia un archivo poseen un archivo de dispositivo (device file) que es un objeto especial del sistema de archivos que proporciona la interfase con el dispositivo.
  - ❑ Un programa que lea el archivo de dispositivo del terminal, leerá los caracteres introducidos por teclado
-

## 2. TUBERIES

---

- ❑ Desde el punto de vista de un programa no hay diferencia entre leer texto guardado en un archivo o introducido mediante el teclado.
  - ❑ Escribir un texto en un archivo o en la pantalla del terminal también son operaciones equivalentes.
  - ❑ Extendiendo esta idea es también posible enlazar la salida de un programa con la entrada de otro. Esta operación se puede realizar mediante una tubería (pipe) que une dos comandos en uno solo. Por ejemplo:  
***\$ls -l | more***
  - ❑ La tubería redirecciona la salida del comando `ls -l` hacia el paginador `more`, que permite visualizar la información pantalla a pantalla.
-



### 3. REDIRECCIONAMENT

---

El descriptor stdout de los programas está asociado por defecto a la pantalla del terminal. Pero de la misma manera que es posible redireccionar la salida de un comando a la entrada de otro, también es posible redireccionar cualquiera de los tres descriptores a un archivo.

---

### 3. REDIRECCIONAMENT

| operador                      | descripció   |
|-------------------------------|--|
| cmd < file                    | El contenido de file se utilizará como entrada de la orden cmd   |
| cmd 1> file1<br>cmd > file1   | La salida estándar de la orden cmd se guardará en file. Si el archivo file existe se sobrescribirá.                        |
| cmd 1>> file1<br>cmd >> file1 | La salida estándar de la orden cmd se guardará en file. La información se escribirá al final del archivo.                  |
| cmd 2> file2                  | Los errores de la orden cmd se guardarán en file. Si el archivo file existe se sobrescribirá.                              |
| cmd 2>> file2                 | Los errores de la orden cmd se guardarán en file. La información se escribirá al final del archivo.                        |
| cmd > file 2>&1               | La salida estándar y la de errores de la orden cmd se guardarán en file. Si el archivo file existe se sobrescribirá.       |
| cmd >> file 2>&1              | La salida estándar y la de errores de la orden cmd se guardarán en file. La información se escribirá al final del archivo. |
| cmd1   cmd2                   | Redirecciona la salida del comando cmd1 hacia la entrada del comando cmd2.   |

## 3. REDIRECCIONAMENT

---

### Ejemplos Redireccionamiento

**\$ls -lR > file**

La salida estándar de la orden `ls -lR` (listado de archivos y subdirectorios que contiene el directorio actual de forma recursiva) se guarda en el archivo `file`.

**\$mkdir directory 2> error**

Si al crear el directorio `directory` se produce algún error (por ejemplo el directorio ya existe o el usuario no tiene permisos de escritura) el mensaje del error se guardará en el archivo `error`.

**\$cp -vf /root /home/alumno > file 2> error**

La lista de todos los archivos copiados (atención a la opción `-v` de la orden `cp`) se guardará en el archivo `file`. Los errores que se produzcan en la ejecución de la orden se almacenarán en el archivo `error`.

---





### 3. REDIRECCIONAMENT

---

**\$cp -vf /root /home/alumno >> file**

La lista de todos los archivos copiados se escribirá en el archivo file. Si el archivo file ya existe la lista se escribirá al final del archivo, manteniendo la información previa.

**\$wc -l < file**

Cuenta el número de líneas del archivo file.

**\$cd /root/setup 2> null**

Redirecciona la salida de error hacia el dispositivo nulo. El mensaje de error no se escribirá en ninguna parte.

**\$ls -l | grep vmlinux**

Redirecciona la salida de la orden ls hacia la orden grep, que mostrará únicamente las líneas que contienen la palabra vmlinux.

---





## 4. COMANDAMENTS

---

### ❖ more

**Sintaxis:** more file

El comando more es un paginador de archivos que permite ver el contenido del archivo file pantalla a pantalla. Al mostrar una pantalla completa la visualización del documento se detiene hasta que el usuario indica que quiere ver otra pantalla más con la tecla SPACE. La tecla Q/q permite salir del programa.

---

## 4. COMANDAMENTS

---

### ❖ less

**Sintaxis:**        less file

El comando less es un paginador de ficheros más potente que la orden more. Permite desplazarse hacia delante y atrás en los archivos.

**Órdenes:**

| Teclas   | Función   |
|----------|---|
| B ó b    | Desplaza una pantalla hacia atrás el contenido del archivo.   |
| RETURN   | Desplaza una línea hacia delante el contenido del archivo.    |
| SPACE    | Desplaza una pantalla hacia delante el contenido del archivo. |
| /pattern | Busca hacia delante el pattern dentro del archivo file.       |
| H ó h    | Muestra un texto de ayuda.                                    |
| Q ó q    | ↓ Sale del programa.  |

---

## 4. COMANDAMENTS

---

### ❖ **grep**

**Sintaxis:**        `grep [options] regex files`

Busca en los archivos especificados las líneas que contienen una coincidencia con la expresión proporcionada en regex. Por defecto mostrará las líneas que contienen una coincidencia y ocultará las que no tengan ninguna coincidencia.

### **Opciones:**

- c Muestra el número de líneas que contienen la expresión, no muestra el contenido de las líneas.
  - i No hace diferencias entre mayúsculas y minúsculas al comparar la expresión.
  - v Muestra todas las líneas que NO tienen ninguna coincidencia y oculta las que contienen alguna coincidencia.
-

## 4. COMANDAMENTS

---

### Ejemplos Grep:

```
$ls -AIR /* | grep -ic .gif
```

La tubería redirecciona la salida del comando `ls -AIR /*` (listar en formato largo todos los archivos del sistema incluido los ocultos) al comando `grep`. El comando `grep` mostrará el número de coincidencias con la expresión `.gif` independientemente si este escrito en minúsculas o mayúsculas.

```
$ls -AIR /root/* | grep -iv root
```

El comando `grep` mostrará todos los archivos del directorio `/root` que no pertenecen al usuario o grupo `root`.

---



## 4. COMANDAMENTS

---

### ❖ sort

**Sintaxis:**        `sort [options] file`

Ordena los registros o líneas de uno o más archivos y lo muestra por la salida estándar (pantalla). La ordenación se puede hacer por el primer carácter, por el primer campo de la línea o por un campo distinto al primero en el caso de ficheros estructurados.

### Opciones:

- n Ordena los campos numéricos por su valor numérico.
  - r Realiza una ordenación inversa (de mayor a menor).
  - k numero Indica la columna en la que vamos hacer la ordenación.
  - t Indica el delimitador de campos que encontraremos diferente de espacio en blanco (separador por defecto de las columnas)
  - u Suprime todas las líneas repetidas después de la ordenación.
  - o Indica el fichero de salida en lugar de la salida estándar
-

## 4. COMANDAMENTS

---

### Ejemplos sort

**sort fichero** → Se realizaría la ordenación y el resultado se mostraría por pantalla.

**sort fichero > fichero\_ordenado** → Obtener el resultado de la ordenación en un fichero,

**sort fichero1 fichero2 > fichero3** → Ordena varios ficheros y añadimos el resultado a otro.

**sort -o f1 f1** → Ordena un fichero y dejamos el resultado de la ordenación en el mismo fichero

**sort -t, -k3** → Ordena un fichero el que los campos están separados por comas, por el campo número 3:

**cat /etc/passwd | sort -t":" -k3n** → Ordena el fichero de usuarios por la columna uid. El fichero passwd utiliza separadores ":"

---





CIPFP Mislata  
Centre Integrat Públic  
Formació Professional Superior

# **PRACTICA 2**

## **REDIRECCIONAMENT I TUBERIES**

---

# Practica 2

---

## PRACTICA 2

# REDIRECCIONAMENT I TUBERIES

---

**IMPORTANT:**

**PER A REALITZAR ESTOS EXERCICIS TENS QUE TENIR EN COMPTE QUE:**

**1. El nom de la teua màquina te que ser inicial del teu nom + 1r cognom. Exemple: Manuel Enguidanos → nom de la meua màquina: *menguidanos***



# PRACTICA 2

## REDIRECCIONAMENT I TUBERIES

---

**Exercici 1.** Enumera i explica els tres tipus de descriptors associats als programes de l'interpret de comandos.

**Exercici 2.** Llista tots els arxius del sistema mitjançant el format llarg i filtrar mitjançant l'ordre **grep** aquells que contenen la paraula **http** independentment si estan escrits en majúscules o minúscules.

**Exercici 3.** Obtén un llistat dels fitxers del directori **/etc**, ordenat per grandària d'arxiu. Indica com és la columna de l'eixida del comando **ls** que hem d'ordenar mitjançant l'opció **-k** del comando **sort** (utilitzar la pipe **ls | sort**)

**Exercici 4.** Obtén un llistat per pantalla dels fitxers del directori **/etc**, ordenat de major a menor per grandària d'arxiu.

**Exercici 5.** Obtén un llistat dels fitxers del directori actual, ordenat per nom de l'arxiu.

**Exercici 6.** Des del teu directori personal, crea l'arbre de directori **Usuaris/ Administradors/Xarxa\_local** amb una sola instrucció.

---



# PRACTICA 2

## REDIRECCIONAMENT I TUBERIES

---

**Exercici 7.** Crea un fitxer de text anomenat "*usuaris*" en el directori *Xarxa\_local* i completa'l amb el següent contingut:

*Admin, Pass: 1234*

*Pere, Pass: 1325*

*Villacampa, Pass: stucom*

*Xavi, Pass: ben*

*Laia, Pass: gràcia*

**Exercici 8.** Llista el contingut de la teua carpeta d'usuari i afig-lo al final de l'arxiu *usuaris*.

**Exercici 9.** Amb el comando *sort* ordena el fitxer *usuaris*. Repetix el procés bolcant el resultat en el fitxer *SortedUsers*. Mitjançant el comando *cat* visualitza el fitxer *SortedUsers*

**Exercici 10.** Amb el comando *sort* ordena el fitxer *usuaris* segons el camp *password*. Bolca el resultat en el fitxer *SortedPass*. Mitjançant el comando *cat* visualitza el fitxer *SortedPass*.

---



# PRACTICA 2

## REDIRECCIONAMENT I TUBERIES

---

**Exercici 11.** Mitjançant l'ordre *grep* comptar el nombre de línies que contenen el patró "pass" en l'arxiu *usuaris*.

**Exercici 12.** Canviar de nom l'arxiu *usuaris* a *Users*

**Exercici 13.** Crea un fitxer de text anomenat *Admins* en el directori *Xarxa\_local* i completa'l amb el següent contingut:

*Local PASS: 1234*

*Xavi Sánchez*

*Laia Sonsola*

*Remoto1, PASS: 1325*

*Remoto2, PASS: stucom*

**Exercici 14.** Filtra totes les línies que contenen el patró *pass* dels fitxers *Admins* i *Users*. Bolca el contingut en un fitxer *pass*

**Exercici 15.** Visualitza pantalla a pantalla i línia a línia mitjançant l'ordre

*less* el contingut del directori personal de l'usuari actual. Quines ordens del programa *less* heu usat?

---



# PRACTICA 2

## REDIRECCIONAMENT I TUBERIES

---

**Exercici 16.** Crea el directori **test** en el directori personal de l'usuari actual.

**Exercici 17.** Crea una altra vegada el directori **test** en el directori personal de l'usuari actual i redirigir l'eixida d'error a l'arxiu **error.txt**. Amb l'ordre **more** visualitza el contingut de l'arxiu **error.txt**.

**Exercici 18.** Buscar el nombre d'arxius amb terminació **.gif** en tot el sistema (utilitzar l'ordre **grep**)

**Exercici 19.** Llista el contingut de totes els directoris o arxius continguts en el directori **/etc** i al seu torn el contingut de cada directori la sisena lletra del qual del nom siga **a**.

---





---

## **UNITAT DIDACTICA 3**

# **SISTEMA OPERATIU LINUX. ADMINISTRACIÓ I CONFIGURACIÓ**

**Prof. Manuel Enguidanos**