

UNIDAD DIDACTICA 5

REDIRECCIONAMIENTO Y TUBERIAS

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 interface 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 interface con el dispositivo.

☞ Un programa que lea el archivo de dispositivo del terminal, leerá los caracteres introducidos por teclado

1. ENTRADA/SALIDA ESTANDAR Y LOS DESCRIPTORES DE ARCHIVO

∞ La Entrada/Salida estándar es la capacidad del intérprete de comandos o shell para controlar y dirigir:

- La entrada de datos de los programas,
- La salida de información útil
- La salida de información de errores.

∞ Cuando un programa se ejecuta, automáticamente se le proporcionan 3 descriptores de archivo:

- **Entrada estándar (stdin)**
- **Salida estándar (stdout)**
- **Error estándar (stderr)**

1. ENTRADA/SALIDA ESTANDAR Y LOS DESCRIPTORES DE ARCHIVO

Entrada estándar (stdin) o descriptor de archivo 0

El descriptor de archivo STDIN está asociado a la entrada de texto. Por defecto está asociado al teclado.

Salida estándar (stdout) o descriptor de archivo 1

El descriptor de archivo STDOUT se corresponde con la salida de texto normal de los programas. Por defecto está asociado al terminal de pantalla.

Error estándar (stderr) o descriptor de archivo 2

El descriptor de archivo STDERR también es una salida de texto, pero se usa exclusivamente para mostrar los errores generados por los programas. Por defecto también está asociado al terminal de pantalla.

2. TUBERIAS

- œ Desde el punto de vista de un programa no hay diferencia entre leer texto guardado en un archivo o introducido mediante el teclado.
- œ Escribir un texto en un archivo o en la pantalla del terminal también son operaciones equivalentes.
- œ 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`

- œ La tubería redirecciona la salida del comando `ls -l` hacia el paginador `more`, que permite visualizar la información pantalla a pantalla.

3. REDIRECCIONAMIENTO

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. REDIRECCIONAMIENTO

operador	descripción
cmd < file	El contenido de file se utilizará como entrada de la orden cmd
cmd 1> file1 cmd > file1	La salida estándar de la orden cmd se guardará en file. Si el archivo file existe se sobrescribirá.
cmd 1>> file1 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. REDIRECCIONAMIENTO

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. REDIRECCIONAMIENTO

\$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. COMANDOS

🔗 **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. COMANDOS

❧ 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. COMANDOS

🔗 **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. COMANDOS

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. COMANDOS

❧ 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. COMANDOS

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 ":"

Ejercicios

REDIRECCIONAMIENTO Y TUBERIAS

Paso 1. Enumera y explica los tres tipos de descriptores asociados a los programas del intérprete de comandos.

Paso 2. Listar todos los archivos del sistema mediante el formato largo y filtrar mediante la orden grep aquellos que contienen la palabra http independientemente si están escritos en mayúsculas o minúsculas.

Paso 3. Obtener un listado de los ficheros del directorio /etc, ordenado por tamaño de archivo. Indica cual es la columna de la salida del comando ls que debemos ordenar mediante la opción -k del comando sort (utilizar la pipe **ls | sort**)

Paso 4. Obtener un listado por pantalla de los ficheros del directorio /etc, ordenado de mayor a menor por tamaño de archivo.

Paso 5. Obtener un listado de los ficheros del directorio actual, ordenado por nombre del archivo.

Paso 6. Desde tu directorio personal, crea el árbol de directorio Usuarios/Administradores/Red_local con una sola instrucción.

Ejercicios

REDIRECCIONAMIENTO Y TUBERIAS

Paso 7. Crea un fichero de texto llamado "usuarios" en el directorio Red local y complétalo con el siguiente contenido:

Admin, Pass: 1234

Pere, Pass: 1325

Villacampa, Pass: stucom

Xavi, Pass: bcn

Laia, Pass: gracia

Paso 8. Lista el contenido de tu carpeta de usuario y añádelo al final del archivo Usuarios.

Paso 9. Con el comando sort ordena el fichero Usuarios. Repite el proceso volcando el resultado en el fichero SortedUsers. Mediante el comando Cat visualiza el fichero SortedUsers

Paso 10. Con el comando sort ordena el fichero Usuarios según el campo password. Vuelca el resultado en el fichero SortedPass. Mediante el comando Cat visualiza el fichero SortedPass.

Ejercicios

REDIRECCIONAMIENTO Y TUBERIAS

Paso 11. Mediante la orden grep contar el número de líneas que contienen el patrón "pass" en el archivo Usuarios.

Paso 12. Renombrar el archivo Usuarios a Users

Paso 13. Crea un fichero de texto llamado Admins en el directorio Red local y complétalo con el siguiente contenido:

Local PASS: 1234

Xavi Sánchez

Laia Sonsola

Remoto1, PASS: 1325

Remoto2, PASS: stucom

Paso 14. Filtra todas las líneas que contienen el patrón pass de los ficheros Admins y Users. Vuelca el contenido en un fichero pass

Paso 15. Visualizar pantalla a pantalla y línea a línea mediante la orden less el contenido del directorio personal del usuario actual. ¿Qué ordenes del programa less habéis usado?

Ejercicios

REDIRECCIONAMIENTO Y TUBERIAS

Paso 16. Crear el directorio test en el directorio personal del usuario actual.

Paso 17. Crear otra vez el directorio test en el directorio personal del usuario actual y redireccionar la salida de error al archivo error.txt. Con la orden more visualizar el contenido del archivo error.txt.

Paso 18. Buscar el número de archivos con terminación .gif en todo el sistema (utilizar la orden grep)

Paso 19. Lista el contenido de todas los directorios o archivos contenidos en el directorio /etc y a su vez el contenido de cada directorio cuya sexta letra del nombre sea a.