

Práctica 4: Redireccionamientos y filtros

Sistemas Operativos

Grado en Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Universidad de Cádiz

Práctica 4:
Redirecciona-
mientos y
filtros

Sistemas
Operativos

Redirecciones
de la Entra-
da/Salida

Listas de
órdenes

Agrupación de
órdenes

Filtros

- 1 Entrada/Salida estándar
- 2 Listas de órdenes
- 3 Agrupación de órdenes
- 4 Concepto de filtro
- 5 Ordenación de ficheros
- 6 Selección de caracteres o campos de un fichero
- 7 Eliminación de líneas duplicadas
- 8 Borrado de columnas
- 9 Sustitución de caracteres

- Por convenio, la mayor parte de los programas de Linux:
 - Toman su entrada de la **entrada estándar**.
 - Envían su salida a la **salida estándar**.
 - Envían los mensajes de error a la **salida de errores estándar**.
- Por omisión:
 - Salida estándar → Pantalla o ventana de la terminal.
 - Entrada estándar → Teclado.
 - Salida de errores estándar → Pantalla o ventana de la terminal.
- El shell permite redirigir estos flujos de información mediante el uso de ciertos operadores.

Operadores de redirección

Práctica 4: Redirecciona- mientos y filtros

Sistemas Operativos

Redirecciones de la Entra- da/Salida

Listas de órdenes

Agrupación de órdenes

Filtros

| Operador | Función |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------|
| <i>orden > fichero</i> | Escribe la salida de <i>orden</i> en <i>fichero</i> . Crea <i>fichero</i> si no existe, en otro caso lo reescribe. |
| <i>orden >> fichero</i> | Añade la salida de <i>orden</i> a <i>fichero</i> . Crea <i>fichero</i> si no existe. |
| <i>orden < fichero</i> | Toma la entrada para <i>orden</i> de <i>fichero</i> . |
| <i>orden 2> fichero</i> | Escribe la salida de errores estándar en <i>fichero</i> . Crea <i>fichero</i> si no existe. |
| <i>orden 2>> fichero</i> | Añade la salida de errores a <i>fichero</i> . |
| <i>orden &> fichero</i> | Redirecciona las salidas estándar y de errores al mismo <i>fichero</i> . |
| <i>orden > fichero</i> | Redirecciona la salida a <i>fichero</i> , incluso si éste existe y la opción noclobber está activada. |
| <i>orden1 orden2</i> | Conecta la salida estándar de la <i>orden1</i> a la entrada estándar de la <i>orden2</i> . |

- Los descriptores de ficheros se utilizan para identificar los flujos de datos asociados a un proceso.
- En Linux, inicialmente se asocia a cada proceso 3 descriptores de ficheros:
 - 0 Entrada estándar.
 - 1 Salida estándar.
 - 2 Salida de errores estándar.
- Si el proceso, durante su ejecución, abre ficheros adicionales se les asignan descriptores de ficheros superiores.

Ejemplos de redirecciones

Práctica 4:
Redirecciona-
mientos y
filtros

Sistemas
Operativos

Redirecciones
de la Entra-
da/Salida

Listas de
órdenes

Agrupación de
órdenes

Filtros

- ➊ `> prueba` Crea un fichero vacío denominado `prueba`.
- ➋ `ls -l > prueba` Crea un fichero denominado `prueba` con la salida de la orden `ls -l`, si ya existía lo reescribe.
- ➌ `ls -l >> prueba` ¿Qué contendrá el fichero `prueba`?
- ➍ `wc -l < prueba` La orden `wc -l` toma su entrada del fichero `prueba`.
- ➎ `set -o noclobber; find . -name "*.txt" > prueba` ¿Qué contendrá el fichero `prueba`?
- ➏ `find . -name "*.txt" >| prueba` ¿Y ahora qué contendrá?
- ➐ `find . -name "*.txt" > salida 2> errores` ¿Qué contendrán los ficheros `salida` y `errores`?
- ➑ `find . -name "*.txt" &> salida2` ¿Qué contendrá el fichero `salida2`?
- ➒ `ls -l dir1 | wc -l` ¿Qué salida obtendremos?

- Suponga que queremos enviar la salida de una orden a un fichero y como entrada de otra orden ¿Cómo podríamos conseguirlo?

```
❶ ls -l > listado | wc -l
```

- Suponga que queremos enviar la salida de una orden a un fichero y como entrada de otra orden ¿Cómo podríamos conseguirlo?

❶ `ls -l > listado | wc -l`

Incorrecto: La salida de `ls -l` se almacena en el fichero `listado` pero `wc -l` no la recibe como entrada.

- Suponga que queremos enviar la salida de una orden a un fichero y como entrada de otra orden ¿Cómo podríamos conseguirlo?

❶ `ls -l > listado | wc -l`

Incorrecto: La salida de `ls -l` se almacena en el fichero `listado` pero `wc -l` no la recibe como entrada.

❷ `ls -l | tee listado | wc -l`

- Suponga que queremos enviar la salida de una orden a un fichero y como entrada de otra orden ¿Cómo podríamos conseguirlo?
 - 1 `ls -l > listado | wc -l`
Incorrecto: La salida de `ls -l` se almacena en el fichero `listado` pero `wc -l` no la recibe como entrada.
 - 2 `ls -l | tee listado | wc -l`
- La orden **tee** bifurca la salida de la orden anterior hacia dos sitios diferentes: a un fichero y como entrada de otra orden.

Una lista de órdenes es una secuencia de una o más tuberías separadas por uno de los operadores siguientes:

`;` `&` `&&` `||`

Órdenes múltiples

El shell permite escribir varias órdenes en una misma línea. El resultado es equivalente a dar las órdenes independientemente una de otra.

Ejemplo:

`pwd; cd ..; pwd` es equivalente a dar:

`pwd`

`cd ..`

`pwd`

Ejecución condicional

El shell proporciona operadores condicionales: **&&** y **||**

Ejemplos:

- 1 `ls fich1 && echo el fichero fich1 existe`
- 2 `ls fich1 || echo el fichero fich1 no
existe`

- El shell proporciona dos formas de agrupar órdenes: `{}` y `()`.
- El shell trata sintácticamente un grupo como si fuera una orden simple a efectos de interconexiones, redirecciones u orden de evaluación en mandatos condicionales.

Ejemplo:

¿Qué mostrarán las siguientes líneas de órdenes?

- 1 `echo N° de líneas;; cat fich1 | wc -l`
- 2 `{ echo N° de líneas;; cat fich1; } | wc -l`

- La utilización de `()` es similar, pero la lista de órdenes se ejecuta en un subshell.

- Los filtros son programas que hacen una selección o transformación del flujo de datos de la entrada estándar o de un fichero, en la salida estándar.
- Los filtros pueden utilizarse junto con los redireccionamientos y las interconexiones.
- Algunos filtros:

| Filtro | Acción |
|--------------------|-------------------------------------------------------------|
| <code>sort</code> | Ordena las líneas de la entrada. |
| <code>cut</code> | Selecciona caracteres o campos de cada línea de la entrada. |
| <code>uniq</code> | Elimina líneas duplicadas consecutivas de la entrada. |
| <code>colrm</code> | Elimina columnas de la entrada. |
| <code>tr</code> | Sustituye o borra caracteres de la entrada. |