Filtros y Redireccionamientos en MS-DOS

Los filtros del DOS

Los filtros son un tipo de comandos que reciben un flujo de texto, realizan una operación sobre el y devuelven una salida

More

Muestra el contenido de un archivo pantalla a pantalla. Después de leer los datos en pantalla, podemos pulsar **Enter** para continuar o también **Ctrl+Pausa** para interrumpir.

Sort

Recibe como entrada un conjunto de caracteres organizados en filas y devuelve los mismos caracteres ordenados por fila.

- Sintaxis
  Sort [parámetros] archivo/s

El uso más frecuente de este filtro es **ordenar alfabéticamente** el contenido de un fichero. Admite dos modificadores:

Modificador	Descripción
/R	Ordena inversamente (de la `Z' a la `A' y del `9' al `0')
/+columna	Indica la *columna* del carácter a partir del cual se comparará cada fila.
/o archivo	Indica el archivo donde se almacenará la salida del comando, sino se especifica se devuelve por salida estandar

Find

Recibe como entrada un conjunto de caracteres organizados en filas y devuelve las filas que contengan una cadena determinada. Por defecto, distingue mayúsculas y minúsculas.
- Sintaxis
  Find “Cadena” Fichero [Parámetros]

Modificador	Descripción
/i	No distingue mayúsculas y minúsculas
/v	Muestra las líneas que no contengan la cadena especificada
/c	Muestra sólo el número de líneas que contienen la cadena.
/n	Muestra el número de línea de cada línea que contiene la cadena.

CC BY NC SA

Redireccionamientos

En los shells de consola tenemos

- StdIn (Standard Input): En la entrada por defecto para un comando
- StdOut (Standard Output): Es la salida por defecto para un comando.

Cuando ejecutamos un comando, este recibe su entrada desde el teclado y muestra los resultados en pantalla, ya que los valores por defecto son:

- Stdin: teclado
- Stdout: Monitor

Operadores de Redireccionamiento

Podemos cambiar la entrada y salida estándar empleando los operadores de redirección:

Operador	Descripción
<	Redirección de la entrada
>	Redirección de la salida
>>	Redirección de la salida a un fichero existente
|	Redirección de la salida de un comando a otro comando

Ejemplos

- Redirección de Entrada <
  - `more < ejemplo.txt`
- Redirección de Salida >
  - `dir C:\ /a /s  > Z:\listado.txt`
- Redirección de Engadir >>
  - `dir C:\ /a /s  >> Z:\listado.txt`
- Tubería
  - `dir C:\ /a /s  | more`

¿Qué hace el siguiente comando?

```
assoc |find "doc"
```

Clip

Equivale al portapapeles de Windows. ¿**Que hacen los siguientes comandos?:**

```
DIR | CLIP
CLIP < README.TXT
```


CC BY NC SA

Dispositivos

DOS tiene nombre reservados para dispositivos que no pueden ser utilizados en archivos y carpetas.

Dispositivo	Descripción
CON	• Entrada por defecto (teclado) • Salida por defecto (monitor)
LPTn	Puertos paralelos. DOS admite hasta 3 puertos paralelos: LPT1, LPT2, LPT3.
COMn	Representan los puertos serie. DOS puede reconocer hasta 4 puertos serie: COM1, COM2, COM3, COM4.
NUL	Simula un dispositivo ficticio. Funciona como un agujero negro, cualquier información que se le envíe desaparece. Se suele emplear para ocultar la salida estándar
LPR o PRN	Representan la impresora local

Antiguamente era una forma de comunicarse con el hardware. Hoy raramente se utilizan a excepción del NUL, que sigue siendo muy importante.

BY NC SA

Redirección de la salida Error Estandar. `2>'

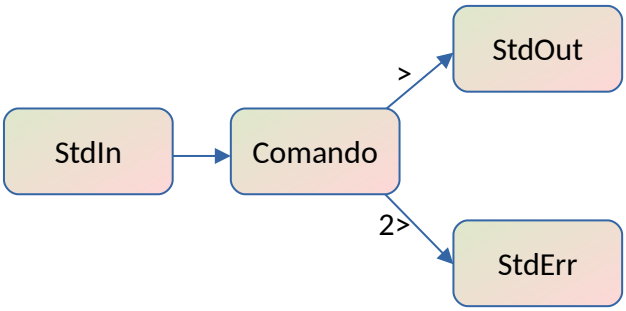
Como ya comentamos '>' sirve para direccionar la salida estándar de un comando.

Problema:

Si ejecuto `dir XX > listado.txt` (XX no existe), nos muestra algo por pantalla "No se encuentra el archivo" y el resto de la salida estándar la envía al fichero.

Esto es así porque **> no redirecciona los errores** que produzca un comando para eso se utiliza **2>**

`dir fich 2> listado.txt` ocurrirá lo contrario, el fichero contendrá el mensaje de error y por pantalla saldrá el error estándar.



Ejemplos:

- `dir fich 2> nul` → No se ve por pantalla el mensaje de error
- `dir fich 2> error.log > estandar.log` → Redirecciona ambas salidas a distintos ficheros de texto
- `dir fich > resultado.log 2>&1` → Ambas salidas se almacenarían en resultado.log

Ejercicio

- Que hace el siguiente comando?

```
xcopy C:\Origen C:\Destino /d/e/y/c/i/h > log.txt 2>&1
```

Ejecutar un comando sólo si el anterior acabó correcta o incorrectamente.

`comando1 || comando2` → El comando 2 se ejecuta si en el comando1 se produce algún error
`comando1 && comando2` → El comando 2 se ejecuta si en el comando1 terminó correctamente
`comando1 & comando2` → El comando 2 se ejecuta después del comando1
`comando1 ; comando2` → Ejecuta dos comandos independientes en la misma línea

Si queremos llevar registro de los errores pero que no aparezcan en pantalla
`dir fich 2> nul || echo error en el comando dir >> errores.log`
Si queremos llevar registro de comandos que acabaron correctamente
`dir fich > nul && dir correcto >> aciertos.log (fich existe)`

Variables de Entorno

¿Cómo encuentra MS-DOS los comandos?

Cuando ejecutamos un programa en consola, ¿Porqué a veces lo encuentra aunque esté en otro directorio y a veces no?

Cuando introducimos un comando y pulsamos Enter, el intérprete de comandos:

1. Comprueba si el comando introducido es un comando interno. Ejemplo: dir o copy (incluidos en el command.com)
2. Comprueba si es un archivo con extensión .com o .exe en el directorio actual
3. Comprueba si es un .bat en el directorio actual
4. Si no está en el directorio actual, comprueba los pasos 2 y 3 en cada uno de los directorios de la variable PATH.

Variables de Entorno

Las variables de entorno son como variables de un programa, pero existen en el Sistema Operativo, pueden ser consultadas por los programas y se utilizan entre otras coas para controlar su funcionamiento, personalizar la apariencia del DOS, etc.

Para asignarles un valor se utiliza el comando **SET**
-Formato:
Set [Variable=[cadena]]
-Ejemplo:
Set palabra=”hola”
Set sin parámetros muestra el valor de todas las variables de entorno definidas.

Si queremos obtener el valor de una variable de entorno **%nombre%**. Ejemplo echo %os%

Algunas de las variables de entorno que trae predefinidas el sistema pueden sernos muy útiles:

USERNAME	Nombre del usuario que inició sesión
USERDOMAIN	Nombre del dominio en el que se inició sesión
CD	Contiene la ruta del directorio actual
DATE	Fecha actual
TIME	Hora actual
RANDOM	Nº aleatorio entre 0 y 32767
ERRORLEVEL	El valor del Errorlevel actual

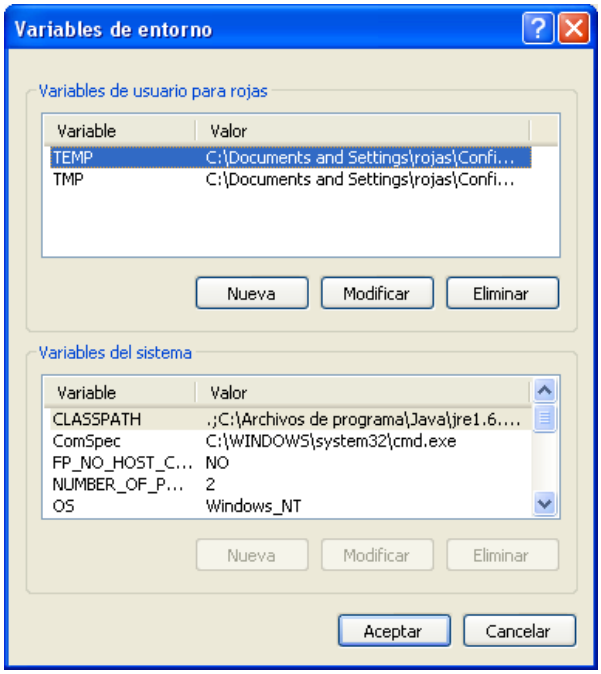
También podemos realizar operaciones con los valores de las variables de entorno, podemos obtener subcadenas y hacer substituciones:

Ejercicio:
Con la ayuda de set /? averigua que hacen los siguientes comandos

- Echo %PATH:~10,5%
- Echo %time:~0,5%
- Set var=1.234.567
- Echo %var:.=@%

BY NC SA

Para modificar las variables de entorno desde el entorno gráfico
MiPC → Propiedades → Opciones Avanzadas → Variables de entorno



Setx

Permite crear variables de entorno de usuario o de sistema con funciones avanzadas. Las variables quedan almacenadas en el sistema. **Nota:** Es necesario reiniciar la consola para utilizarlas.

- `setx Escola IESRODEIRA` → Definida para nuestro usuario
- `setx Aula Aula4 /m` → Definida para el sistema (necesita permisos)

También permite formatear un archivo de texto para quedarnos con el valor que nos interese

Ejemplo:

Este es un ejemplo de documento	`Z:\>setx /f prueba.txt /x`
	`(0,0 Este)(0,1 es)(0,2 un)(0,3 ejemplo)`
	`(1,0 de)(1,1 documento)`
		`Z:\>setx prueba /f prueba.txt /a 0,0`
		`Valor extraído: Este.`
		`CORRECTO: se guardó el valor especificado.`

Crea un archivo llamado prueba.txt con este contenido	Setx formatea el texto asignando coordenadas a cada palabra	Podemos asignar a una variable de entorno sólo una palabra

Ejercicio: Utilizando el comando `net config Workstation`, define una variable de entorno que almacene el dominio al que estamos conectados.

CC BY NC SA