

Ficheros por lotes

¿Qué es un fichero por lotes o archivo "batch"?

Un fichero por lotes es un archivo de texto plano que contiene una serie de órdenes interpretables por el intérprete de comandos o CLI (command line interface).

En muchas ocasiones los usuarios de ms-dos repiten una determinada secuencia de órdenes día tras día, suponiendo una tarea bastante molesta e incómoda. Los ficheros por lotes pretenden automatizar una tarea rutinaria ejecutando una serie de órdenes definidas con antelación.

Todos los ficheros por lotes poseen unas características comunes:

- Extensión. Todos deben llevar obligatoriamente la extensión BAT.

- Contenido. Son ficheros de texto ASCII y, por consiguiente, pueden ser creados por Copy o Edit. Cada línea del fichero debe poseer una orden.

- Ejecución. Para hacerlo funcionar simplemente debemos teclear su nombre a continuación del símbolo del sistema. El fichero por lotes tomará entonces el control del ordenador.

- Interrupción. Podemos detener el procesamiento del fichero por lotes en cualquier momento presionando Ctrl+Pausa.

Todas las órdenes admitidas en los ficheros por lotes son las que las que introducen desde el símbolo del sistema, más un conjunto de instrucciones específico para ficheros por lotes.

Algunas de las órdenes generales son:

DIR [<i>nombre-de-directorio</i>]	Muestra los archivos contenidos en el directorio que aparece en el prompt
MD <i>nombre</i>	Crea un nuevo directorio
CD [<i>nombre-de-directorio</i>]	Cambiamos el directorio activo o muestra el directorio activo
DEL <i>nombre-de-fichero</i>	Borra un fichero
RD <i>nombre-de-directorio</i>	Borra un directorio vacío
REN <i>nombre-old nombre-new</i>	Cambia el nombre a un fichero o directorio
COPY <i>fichero-origen destino</i>	Hace una Copia de un archivo en otro lugar
MOVE <i>origen destino</i>	Mueve un archivo o directorio a otro lugar
FIND " <i>cadena</i> " <i>fichero</i>	Busca una cadena de texto en un fichero
TYPE <i>nombre-de-fichero</i>	Visualiza el contenido de un fichero
SORT <i>fichero</i>	Muestra en pantalla el fichero ordenado alfabéticamente
TREE [<i>nombre-de-directorio</i>]	Nos muestra el árbol de directorios
ECHO [<i>mensaje</i>]	Muestra un mensaje en pantalla
TITLE <i>título</i>	Establece un título para la ventana actual
HELP [<i>comando</i>]	Nos muestra un listado de todos los comandos o información de un comando
CLS	limpia la pantalla
VER	muestra la versión de MS-DOS que estamos ejecutando
DATE [<i>día / mes / año</i>]	muestra/cambia la fecha del sistema
TIME [<i>hora : min : segs</i>]	muestra/cambia la hora del sistema
CMD	Inicia una nueva shell en la ventana actual.
START [<i>comando_externo</i>]	Inicia una shell en una nueva ventana, o ejecuta un comando externo.
EXIT	Finaliza la shell

Algunas órdenes externas son:

Orden **Breve descripción**

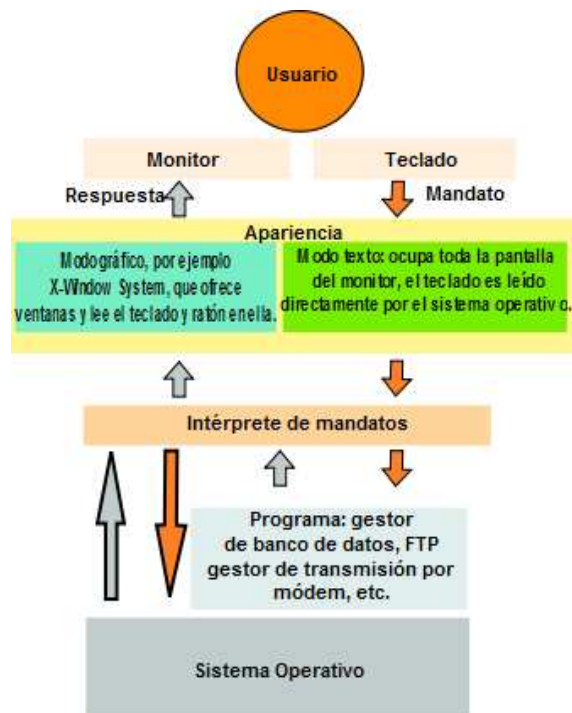
Echo	Controla el eco de las órdenes y visualiza mensajes.
Rem	Introduce comentarios.
Pause	Detiene temporalmente el desarrollo de un programa.
Goto	Desvía incondicionalmente el desarrollo de un programa.
If	Desvía condicionalmente el desarrollo de un programa.
Choice	Permite elegir entre unas opciones establecidas.
For	Repite una misma orden en un conjunto de ficheros.
Call	Llama a un fichero por lotes desde otro.
Shift	Desplaza el valor de los parámetros.

La siguiente tabla muestra aquellos símbolos empleados exclusivamente en los ficheros por lotes:

<u>Símbolo</u>	<u>Significado</u>
:etiqueta	Nombre de una etiqueta.
%número	Parámetro del fichero por lotes.
%variable%	Variable del entorno.
%%variable	Variable de la orden For.

¿Qué es el intérprete de comandos?

La Shell, o intérprete de comandos, es el programa que interpreta las peticiones del usuario para ejecutar comandos. Por ejemplo, en MS-DOS, el intérprete de comandos es COMMAND.COM.



Conceptos previos: redireccionamientos

Para comprender mejor el significado del *redireccionamiento*, intente responder primeramente las siguientes preguntas.

1. Cuando el ordenador solicita información, ¿qué *dispositivo* utiliza para recibirla?
2. Cuando el ordenador necesita presentar información, ¿qué *dispositivo* emplea?

Las respuestas correctas son el teclado y el monitor respectivamente. En este capítulo aprenderemos a redirigir (cambiar) las entradas y salidas de las órdenes del ms-dos. El siguiente cuadro presenta la entrada y la salida por defecto de las órdenes del dos.

Función	Dispositivo
Entrada por defecto	Teclado
Salida por defecto	Monitor

Los operadores

Para variar la entrada de una orden, la salida o ambas, se utilizan los operadores de redireccionamiento. Son cuatro:

Operador	Descripción	Sintaxis
<	Redirección de la entrada	orden <fichero/dispositivo
>	Redirección de la salida	orden >fichero/dispositivo
>>	Redirección de la salida a un fichero existente	orden >>fichero
	Redirección de la salida de una orden a entrada de otra orden	orden1 orden2

Los filtros del DOS

Los filtros del dos son las órdenes **More**, **Sort** y **Find**. Estas tres órdenes externas se denominan *filtros* porque su misión es recibir una entrada, filtrarla y devolver una salida.

More

Recibe como entrada un conjunto de caracteres organizados en filas y devuelve los mismos caracteres insertando una pausa cada 23 líneas.

En muchas ocasiones la información mostrada por una orden o el contenido de un fichero no cabe en una sola pantalla y los datos se desplazan a gran velocidad sin darnos tiempo a leerlos. La utilidad de la orden More se encuentra en estas ocasiones, **deteniendo el desplazamiento de información en cada pantalla**. Entonces, More muestra la siguiente línea:

```
-- Más --
```

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 ordenando cada fila alfabéticamente.

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

<u>Modificador</u>	<u>Descripción</u>
/R	Ordena inversamente (de la `Z' a la `A' y del `9' al `0')
/+columna	Indica la <i>columna</i> del carácter a partir del cual se comparará cada fila.

Find

Recibe como entrada un conjunto de caracteres organizados en filas y devuelve las filas que contengan una cadena determinada.

La cadena de caracteres para buscar debe especificarse entre comillas (Mays+2) después de la orden.

Se utiliza, a menudo, para **buscar una cadena de caracteres** en un fichero o en la salida de una orden. El filtro Find distingue entre mayúsculas y minúsculas mientras no le especifiquemos el modificador **/I** entre la orden y la cadena.

FIND "cadena" (*cadena* es diferente que *CADENA*)

FIND /I "cadena" (*cadena* es lo mismo que *CADENA*)

Redirección de la entrada. Operador '<'

Cómo hemos dicho al principio del capítulo, las órdenes del dos solicitan información a través del teclado. *Con este operador hacemos que la información de entrada proceda desde un fichero o un dispositivo.*

Aplicaciones:

- **Mostrar un fichero de texto con pausa en cada pantalla.**

A:\>more <recuerdo.txt

(*More toma como entrada el fichero RECUERDO.TXT y como salida, el monitor*)

- **Ordenar un fichero alfabéticamente y verlo en la pantalla.**

A:\>sort <personal.txt

(*Sort toma como entrada el fichero PERSONAL.TXT y como salida, el monitor*)

- **Ordenar un fichero alfabéticamente a partir de una columna.** Ejemplo: Si el fichero PERSONAL.TXT contiene nombres y datos del personal de una empresa, podemos ordenarlo por la columna que indica el puesto ejercido por cada empleado.

A:\>sort /+40 <personal.txt

- **Encontrar las filas que contienen una cadena de caracteres.**

A:\>**find "JIMENEZ" <personal.txt**

(Find considera como entrada el fichero PERSONAL.TXT y muestra por pantalla todas las filas que contienen la cadena «JIMENEZ»)

Ejercicios de repaso

1. Indique las órdenes necesarias para:

- a) Mostrar el fichero CRONICA.TXT con pausa en cada pantalla.
- b) Mostrar el fichero AVISO.TXT con pausa en cada pantalla.
- c) Ordenar el fichero NUMEROS.TXT alfabéticamente.
- d) Ordenar el fichero LISTIN.TXT alfabéticamente a partir de la columna 30.
- e) Ordenar el fichero AMIGOS.TXT inversamente.
- f) Buscar la cadena "IGNACIO" en el fichero PERSONAL.TXT

2. Si no utilizamos el operador '<', ¿cuál es el dispositivo de entrada por defecto?

Redirección de la salida. Operador '>'

Con este operador dirigimos la salida desde la pantalla a un fichero o dispositivo.

Suele tener dos usos principales:

1. Enviar la salida de una orden a la impresora.

A:\>**type romance.txt >prn**

(imprime el fichero ROMANCE.TXT en impresora)

C:\>**vol >prn**

(imprime la etiqueta y el número de serie en impresora)

2. Enviar la salida de una orden a un nuevo fichero. Si el fichero existe, se pierde su contenido anterior.

C:\>**dir >director.txt**

(crea el fichero DIRECTOR.TXT con el directorio raíz de C:)

C:\>**ver >version.txt**

(crea el fichero VERSION.TXT conteniendo la versión actual de ms-dos)

El operador '>' también se puede combinar con los filtros del dos. Veamos algunas aplicaciones:

- **Ordenar un texto introducido por teclado en un fichero.** El cursor se sitúa debajo de la orden esperando las filas de texto. Al terminar cada fila debemos pulsar Enter y para terminar la introducción de texto, Ctrl+Z y luego Enter.

B:\>sort >amigos.txt

(Sort asume como entrada el teclado y como salida el fichero AMIGOS.TXT)

- **Ordenar un fichero en otro.**

B:\>sort <listin1.txt >listin2.txt

(Sort asume como entrada el fichero LISTIN1.TXT y como salida, el fichero LISTIN2.TXT)

- **Ordenar un fichero en la impresora.**

B:\>sort <listin1.txt >prn

(Sort toma como entrada el fichero LISTIN1.TXT y como salida, la impresora)

- **Buscar una cadena en un fichero y crear otro con el resultado de la búsqueda.**

B:\>find "CONTABLE" <personal.txt >contable.txt

Ejercicios de repaso

1. **Indique la entrada y la salida de las siguientes líneas de órdenes.**

- a) DATE <DIA.TXT
- b) TIME <HORA.TXT
- c) TYPE ROMANCE.TXT
- d) TYPE ROMANCE.TXT >CON
- e) DIR >DIRECTOR.TXT
- f) DIR >PRN
- g) SORT
- h) SORT <DESORDEN.TXT
- i) SORT <DESORDEN.TXT >ORDEN.TXT
- j) SORT <DESORDEN.TXT >PRN
- k) FIND "HARINA" <COMPRA.TXT
- l) FIND "HARINA" <COMPRA.TXT >HARINA.TXT
- m) MORE <AVISO.TXT

Redirección de la salida a un fichero existente. Operador '>>'

Se utiliza para dirigir la salida de una orden a un fichero existente. Al contrario que el operador '>', el operador de adicción '>>' no sobrescribe sino añade al fichero existente.

La siguiente orden añade al fichero SOS.TXT la ayuda de la orden Time. Si no existe, se crea el fichero.

```
A:\>time /? >>sos.txt
```

La siguiente añade al fichero SOS.TXT la ayuda suministrada por ms-dos para la orden Date.

```
A:\>date /? >>sos.txt
```

Si ahora visualizamos el fichero SOS.TXT, comprobamos que contiene la ayuda de ambas órdenes.

```
A:\>type sos.txt
```

Muestra o establece la hora del sistema.

```
TIME [hora]
```

Escriba TIME sin parámetros para consultar o cambiar la hora actual. Presione ENTRAR para conservar la misma hora. Presenta o establece la fecha.

```
DATE [dd-mm-aa]
```

dd-mm-aa Establece la fecha que se especifique.

Escriba DATE sin parámetros para ver la fecha actual y poder escribir una nueva.

Presione ENTRAR para conservar la misma fecha.

```
D:\>
```

Ejercicios de repaso

1. Escribir las órdenes necesarias para:

- a) Mostrar el fichero INFORME.TXT con pausa en cada pantalla.
- b) Imprimir el fichero INFORME.TXT en la impresora.
- c) Ordenar el texto introducido por el teclado en la impresora.
- d) Añadir la etiqueta y el número de serie de la unidad actual (orden Vol) al fichero VARIOS.TXT
- e) Añadir el directorio actual al fichero VARIOS.TXT

Redirección de la salida de una orden a la entrada de una segunda. Operador '|'

El uso del operador '|' guarda la siguiente sintaxis:

```
orden1 |orden2
```

Nota: El símbolo de tubería o símbolo de canalización '|' se obtiene pulsando AltGr+1.

La lectura de esta sintaxis es: "tomar la salida de orden1 como entrada de orden2".

Cada vez que utilizamos el símbolo tubería, ms-dos necesita crear un fichero temporal. Para ello utiliza la variable de entorno TEMP. Si esta variable no está bien definida, puede ocurrir algún error.

Aplicaciones:

- **Mostrar con pausa en cada pantalla la salida de una orden.**

```
B:\>type informe.txt |more
```

```
C:\DOS>dir |more
```

```
C:\>tree |more
```

- **Obtener una línea determinada de la salida de una orden.**

```
C:\>dir |find "bytes libres"
```

```
20.443.136 bytes libres
```

(muestra los bytes libres de la unidad C:)

- **C:\>set |find "TEMP"**

```
TEMP=C:\DOS
```

(muestra el valor de la variable TEMP o nada si no existe)

Ejercicios de repaso

5. **Entre las siguientes líneas de órdenes, agrupar por parejas las que hagan lo mismo.**

- COPY EJEMPLO.TXT PRN
- TYPE EJEMPLO.TXT |MORE
- TYPE EJEMPLO.TXT >PRN
- COPY EJEMPLO.TXT CON
- DIR /P
- MORE <EJEMPLO.TXT
- TYPE EJEMPLO.TXT
- DIR |MORE

Programación ficheros por lotes

Orden ECHO

Se puede utilizar de cinco formas:

ECHO	Indica si está activado o desactivado el eco de las órdenes.
ECHO ON	Activa el eco

ECHO OFF	Desactiva el eco
ECHO mensaje	Visualiza un mensaje en pantalla.
ECHO.	Visualiza una línea en blanco en pantalla.

El eco de una orden es el propio nombre de aquella escrito en la pantalla. Al poner en marcha el fichero anterior, observará como aparece el nombre de cada orden en pantalla antes de ejecutarse. Al fichero del ejemplo podemos añadirle la línea ECHO OFF para desactivar el eco en lo sucesivo.

Para eliminar el eco de la propia orden ECHO OFF se antepone el *símbolo arroba* (@) al nombre de la orden. Veámoslo más claro en los ejemplos siguientes.

Programa:

```
echo off
date
time
```

Ejecución:

```
C:\BATH>tiempo
```

```
C:\BATH>echo off
La fecha actual es Mar 15/08/1995
Escriba la nueva fecha (dd-mm-aa):
La hora actual es 0:53:54,12
Escriba la nueva hora:
```

Programa:

```
@echo off
date
time
```

Ejecución:

```
C:\BATH>tiempo
La fecha actual es Mar 15/08/1995
Escriba la nueva fecha (dd-mm-aa):
La hora actual es 0:53:54,12
Escriba la nueva hora:
```

La orden Echo también permite mostrar mensajes al usuario del fichero. En el siguiente ejemplo se ha introducido el mensaje "Este es el fichero TIEMPO.BAT":

Programa:

```
@echo off
echo Este es el fichero TIEMPO.BAT
date
time
```

Ejecución:

```
C:\BATH>tiempo
Este es el fichero TIEMPO.BAT
La fecha actual es Mar 15/08/1995
Escriba la nueva fecha (dd-mm-aa):
La hora actual es 1:10:22,72
Escriba la nueva hora:
```

La orden REM

Permite introducir comentarios internos en el código de un fichero por lotes. Las líneas precedidas por la palabra Rem serán ignoradas durante el funcionamiento del fichero por lotes. Se suele utilizar esta orden para introducir aclaraciones en el código del fichero. Objetivo: facilitar su lectura y corrección de errores.

```
@echo off
rem Programa: TIEMPO.BAT
rem Cometido: Mostrar al usuario la fecha y la hora del
rem sistema y ofrecerle la oportunidad de modificarla.
date
time
```

Gestión de los parámetros

Los parámetros son informaciones adicionales colocadas detrás del nombre de una orden. Si la mayoría de las órdenes de ms-dos admiten parámetros, también será posible gestionar parámetros en los ficheros por lotes.

Vamos a confeccionar un fichero por lotes que borre dos ficheros introducidos como parámetros.

Programa:

```
@echo off
rem Programa: BORRA2.BAT
del %1
del %2
```

Ejecución:

```
A:\>borra2 juan.txt maria.txt
```

En la línea de órdenes, cada parámetro debe estar separado con un espacio en blanco del anterior. De la forma anteriormente explicada podemos gestionar hasta nueve de ellos (del %1 al %9).

Para referirnos a un parámetro introducido en la línea de órdenes del programa, debemos escribir el signo de porcentaje (%) seguido del número de parámetro.

El siguiente ejemplo copia los ficheros introducidos como parámetros al disquete de la unidad B:

Programa:

```
@echo off
rem Programa: COPIAB.BAT
echo Se están copiando los ficheros %1, %2 y %3 a la unidad B:
copy %1 b:\
copy %2 b:\
copy %3 b:\
```

Ejecución:

```
A:\>copiab juan.bak alberto.bmp marta.dbf
```

La orden PAUSE

Detiene temporalmente el desarrollo de un programa. Cuando el dos encuentra una orden Pause en un fichero por lotes visualiza un mensaje en pantalla y espera una tecla para proseguir. También podemos presionar Ctrl+Pausa y así, interrumpir el desarrollo del fichero.

Presione cualquier tecla para continuar . . .

Para mejorar la presentación, podemos insertar un mensaje indicando el motivo de la detención del programa gracias a la orden Echo:

```
echo Inserte un disquete en la unidad B:
pause
```

Si no deseamos ver el mensaje de Pause, podemos redireccionarlo al dispositivo ficticio NUL.

```
echo Inserte un disquete en B: y pulse Enter
pause >nul
```

*Ejercicios de repaso***6. ¿Qué hace la orden PAUSE >NUL ?***La orden GOTO*

Normalmente un fichero por lotes se desarrolla secuencialmente, desde la primera línea hasta la última. Sin embargo, *la orden Goto permite desviar la ejecución del programa hasta una etiqueta especificada como parámetro.*

GOTO [:]etiqueta

Las etiquetas deben ir precedidas de dos puntos (:) para diferenciarse de las órdenes. Admiten hasta ocho caracteres significativos. Esto último significa que la etiqueta `BALANCE_DE_AGOSTO' es idéntica a `BALANCE_DE_ENERO'.

El siguiente ejemplo muestra cómo se usan las etiquetas:

Programa:

```
@echo off
ver
goto Final
vol
:Final
```

Ejecución:

Versión MS-DOS 6.22

En este ejemplo, la orden Vol nunca se ejecutará: al llegar la orden GOTO FINAL, ms-dos salta hasta la etiqueta :Final y termina el fichero porque no hay más líneas.

Vamos a crear un programa para copiar varios disquetes desde la unidad A: al directorio actual.

```
@echo off
rem Programa: DEMO.BAT
echo *****
echo ** Este programa copia todos los ficheros **
echo ** de la unidad A: al directorio actual **
echo *****
pause
```

```
:Proceso
copy a:\ .
echo Introduzca el siguiente disquete en A: y pulse Enter
pause >nul
goto Proceso
```

En primer lugar el programa muestra en pantalla su cometido y así, el usuario decide entre continuar (Enter) o abandonar (Ctrl+Pausa). Si pulsa Enter se copian todos los ficheros del directorio raíz de A: en el directorio actual.

Luego, se ofrece la posibilidad de introducir otro disquete. Si pulsamos Enter se ejecuta la orden GOTO PROCESO, desviando la ejecución del programa hasta la etiqueta :Proceso. En cambio, si pulsamos Ctrl+Pausa el programa finaliza, apareciendo el símbolo del sistema.

*Ejercicios de repaso***7.¿Qué hace el siguiente fichero por lotes?**

```
@echo off
dir c:\
goto etiqueta2
:etiqueta1
ver
:etiqueta2
```

La orden IF

Desvía condicionalmente el proceso de ejecución de un fichero por lotes. Admite 6 sintaxis diferentes:

IF EXIST fichero orden	Si existe el fichero se ejecuta la orden.
IF NOT EXIST fichero orden	Si no existe el fichero se ejecuta la orden.
IF cadena1==cadena2 orden	Si ambas cadenas son iguales se ejecuta la orden.
IF NOT cadena1==cadena2 orden	Si ambas cadenas son diferentes se ejecuta la orden.
IF ERRORLEVEL número orden	Si el código de salida del último programa es igual o superior al número, se ejecuta la orden.
IF NOT ERRORLEVEL número orden	Si el código de salida del último programa es inferior al número, se ejecuta la orden.

IF EXIST

Un programador debe considerar todas y cada una de las situaciones posibles al ejecutarse su programa y, por tanto, debe evitar la aparición de mensajes de error inesperados en la pantalla.

Suponga el siguiente fichero por lotes:

```
@echo off
del %1
```

Si ahora lo hacemos funcionar introduciendo como parámetro un fichero inexistente, el ms-dos tomará el control y dará su correspondiente mensaje de error. Este fichero resulta más correcto si contiene las siguientes líneas:

```
@echo off
if not exist %1 echo ¡Es imposible borrar un fichero que no existe!
if exist %1 del %1
De esta forma, la orden Del sólo funciona si el fichero existe.
```

IF cadena1==cadena2

Encuentra su utilidad al trabajar con parámetros en un fichero por lotes.

En el fichero BORRA2.BAT hemos supuesto que el usuario siempre introduce dos parámetros. Sin embargo, se producirá un error si el usuario sólo introduce uno de ellos: la orden DEL %2 quedará transformada en DEL a secas y como Del no funciona sin parámetros, ms-dos mostrará una advertencia. Este problema se soluciona con lo siguiente:

```
@echo off
rem Programa: BORRA2B.BAT
if "%1"==" " echo Debe introducir uno o dos ficheros como parámetros.
if not "%1"==" " del %1
if not "%2"==" " del %2
```

La primera línea If avisa si no se ha introducido ningún parámetro. La segunda línea únicamente borrará el primer parámetro si se ha introducido. La tercera línea hace lo mismo pero con el segundo parámetro.

IF ERRORLEVEL

Cada orden externa de ms-dos genera un *código de salida* a su término indicando si pudo realizarse satisfactoriamente.

Generalmente un código de salida 0 indica que no hubo *ningún problema* y un código de salida superior hace referencia a *diferentes errores*.

Muchos ficheros por lotes necesitan saber si la orden anterior cumplió su cometido correctamente: para ello utilizan la orden If errorlevel.

Es muy importante recordar que la orden se ejecutará si el código de salida es igual o superior al especificado detrás de ERRORLEVEL.

A modo de ejemplo tenemos a continuación los códigos de salida de Xcopy:

<u>Código</u>	<u>Significado</u>
0	Los ficheros fueron copiados sin error.
1	No se encontraron ficheros para copiar.
2	El usuario presionó Ctrl+Pausa para suspender el proceso de Xcopy.
4	Ocurrió un error de inicio. No hay suficiente memoria o espacio en el disco, se introdujo un nombre de unidad no válida o se utilizó una sintaxis incorrecta en la línea de órdenes.
5	Ocurrió un error de escritura de disco.

Vamos a crear un fichero por lotes para copiar los ficheros de la unidad A: a la B: e informe del resultado de la copia.

```
@echo off
rem Programa: COPIA-AB.BAT
xcopy a:\ b:\
if errorlevel 1 goto Error
if errorlevel 0 echo ¡La copia fue correcta!
goto Final

:Error
    echo Se produjo un error durante la copia

:Final
```

En primer lugar, Xcopy intenta realizar la copia de ficheros y devolverá un código de salida. Si se ha producido algún error el código será 1 o superior y entonces, el programa se desvía hasta la etiqueta :Error, muestra el mensaje y finaliza. Si la copia fue satisfactoria, el código de salida es 0. La segunda línea If mostrará el mensaje de éxito, saltando después a la etiqueta :Final y como no hay más líneas, termina el proceso.

En muchas ocasiones puede ser fuente de complicaciones que *If errorlevel número* se cumpla si el número es igual o mayor. Para cumplirse *exclusivamente* si el código de salida es 5 -por ejemplo- podemos usar lo siguiente:

```
if errorlevel 5 if not errorlevel 6 dir
```

Esta compleja línea se traduce así: «*Si el código de salida es 5 o superior pero inferior a 6 ejecutar Dir*», es decir, si el código es 5 ejecutar Dir.

La orden CHOICE

Permite escoger una opción entre varias y, dependiendo de la opción elegida, devuelve un código de salida.

Su sintaxis es:

CHOICE [mensaje] [/C:opciones] [/N] [/S] [/T:opción,segundos]

/C:opciones	Especifica las opciones posibles. Si el usuario pulsa la primera de las opciones, Choice devolverá un código de salida 1; si pulsa la segunda opción, Choice devuelve el código 2 y así sucesivamente. Si no se especifica este parámetro se asumen las opciones por defecto (SN).
/N	No muestra las opciones admitidas detrás del mensaje.
/S	Hace distinción entre mayúsculas y minúsculas. Si no se especifica este parámetro se toman como la misma opción.

/T:opción,segs	Toma la opción indicada si no se pulsa ninguna otra tecla en los segundos especificados.
mensaje	Contiene el mensaje mostrado al usuario pidiendo que introduzca una de las opciones admitidas.

El programa DEMO.BAT podemos mejorarlo sensiblemente si cambiamos las órdenes Pause por órdenes Choice.

```
@echo off
rem Programa: DEMOB.BAT
echo *****
echo ** Este programa copia todos los ficheros **
echo ** de la unidad A: al directorio actual **
echo *****
choice ¿Desea continuar?
if errorlevel 2 goto Final

:Proceso
  copy a:\ .
  choice Para continuar con otro disquete pulse C y para finalizar, F
/C:FC
  if errorlevel 2 goto proceso

:Final
```

En la primera orden Choice se toman las opciones por defecto S y N. `S' corresponde a un código de salida 1 y `N' a un código 2. En la segunda orden Choice se toman las opciones F y C. `F' corresponde a un código 1 y `C' a un código 2.

Con la orden Choice y de una forma muy sencilla podemos crear menús con diferentes opciones:

```
@echo off
rem Programa: UTIL.BAT
:Menu
  cls
  echo UTILIDADES DE MS-DOS
  echo -----
  echo.
  echo A. Anti-Virus
  echo B. Backup
  echo D. Defragmentar
  echo E. Editor
  echo S. Salir
  echo.

  choice ¿Qué utilidad desea comenzar? /c:abdes /n /t:s,15
  if errorlevel 5 goto Salir
  if errorlevel 4 goto Editor
  if errorlevel 3 goto Defrag
  if errorlevel 2 goto Backup
  if errorlevel 1 goto Anti
  if errorlevel 0 goto Menu

:Anti
```

```

MSAV
goto Menu
:Backup
MSBACKUP
goto Menu
:Defrag
DEFRAG
goto Menu
:Editor
EDIT
goto Menu
:Salir
echo.

```

Observe la orden Choice: el modificador /C indica las opciones admitidas. Si se pulsa la `A' se generará un código de salida 1 y así sucesivamente hasta la `S' que corresponde a un código 5. Gracias al modificador /N Choice no muestra las teclas admitidas detrás del mensaje. El modificador /T toma como opción por defecto la `S' si pasan 15 segundos sin pulsar ninguna tecla.

Observe, asimismo, cómo se ha comenzado en las líneas If por el errorlevel más alto: así se evitan conflictos. El código de salida 0 se obtiene si el usuario responde con Ctrl+Pausa al mensaje de Choice.

Ejercicios de repaso

8. Escriba un fichero de procesamiento por lotes para borrar el fichero introducido como parámetro. El programa debe comprobar previamente si se ha introducido algún parámetro y si el fichero existe.

9. ¿Para qué devuelven un código de salida las órdenes externas?

Otras órdenes

La orden FOR

Su sintaxis es:

```
FOR %%variable IN (conjunto) DO orden
```

Esta orden repite la *orden* especificada para cada valor del *conjunto*. *Conjunto* es una lista de nombres de ficheros. En ella, se pueden establecer varios nombres separados por espacios y también, utilizar comodines.

Ejemplo

```
for %%I in (juan.txt maria.txt *.dat) do type %%i
```

La variable %%I va tomando cada uno de los valores del conjunto y se los envía a la orden Type. En este ejemplo se visualizan en pantalla los ficheros JUAN.TXT, MARIA.TXT y todos los que tengan extensión DAT.

La orden SHIFT

Se traduce al español por desplazamiento. *Mueve el valor de cada parámetro a la variable anterior.* Por ejemplo, si existen 3 parámetros (%1, %2 y %3) y se utiliza la orden Shift, el valor de %1 lo tomará %0, el valor de %2 lo tomará %1 y el valor de %3 lo tomará %2.

Programa:

```
@echo off
rem Programa: DEMO2.BAT
echo El parámetro 1 es %1
shift
echo El parámetro 2 es %1
shift
echo El parámetro 3 es %1
```

Ejecución:

```
C:\BATH>demo2 juan maria alberto
El parámetro 1 es juan
El parámetro 2 es maria
El parámetro 3 es alberto
C:\BATH>
```

La orden CALL

Se utiliza para llamar a un fichero por lotes desde el interior de otro. Su sintaxis es:

CALL fichero [ParámetrosDelFichero]

En el siguiente ejemplo, el fichero PRG1.BAT llama a PRG2.BAT. Cuando la ejecución de PRG2.BAT termina, continua PRG1.BAT en la siguiente línea a Call.

Programa PRG1.BAT:

```
@echo off
echo línea 1
call prg2.bat
echo línea 4
echo línea 5
```

Programa PRG2.BAT:

```
@echo off
echo línea 2
echo línea 3
```

Ejecución:

```
C:\BATH>prg1  
línea 1  
línea 2  
línea 3  
línea 4  
línea 5  
C:\BATH>
```