

Un libro que encontré en un ordenador 486DX en un formato bastante extraño, pero tras probarlo con distintos editores logre encontrar uno que si lo leyó bastante bien, el libro esta espectacular, es el mejor que he leído y es del 1993, época en el que el BATCH predominaba las computadoras

DEDICADO A LOS AMADORES DE ESTE LENGUAJE

-----

Carlos Mesa

#####

# TECNICAS DE PROGRAMACION PARA ARCHIVOS BATCH #

#####

Se incluye con el disquete de la recopilación UTILIDADES+2

(Sólo en la versión registrada)

Tercera revisión: Noviembre 1994

Indice

-----

Introducción..... 1

Cómo utilizar este libro electrónico..... 2

Capítulo 1

Organización y personalización en el arranque del ordenador

1.1 A propósito de los archivos por lotes..... 4

1.2 El primer fichero batch es el de arranque..... 4

Capítulo 2

Ordenes del DOS

2.1 Los diez mandamientos.....8

2.1.1 Orden @ .....8

2.1.2 Orden ECHO.....8

2.1.3 Orden REM.....	8
2.1.4 Orden PAUSE.....	9
2.1.5 El símbolo % identifica a un par metro.....	9
2.1.6 Orden CALL.....	11
2.1.7 Orden SHIFT.....	12
2.1.8 Orden IF.....	12
2.1.9 Orden GOTO.....	17
2.1.10 Orden FOR.....	18

### Capítulo 3

Nuevos comandos para los ficheros por lotes

3.1 Norton, el rey Midas de la informática, proporciona un nuevo comando.....	20
3.2 El anónimo LETRA.COM.....	24
3.3 El otro DOS.....	25
3.4 Y aún hay más.....	26

### Capítulo 4

Programación con ANSI.SYS y secuencias de escape

4.1 ANSIS.SYS es un controlador de dispositivo.....	29
4.2 Un artista del color en la pantalla.....	30
4.3 No hace falta Windows para perder el cursor.....	33
4.4 Un virtuoso de las teclas.....	36
4.5 La pantalla muestra sus atributos con otras órdenes....	41
4.6 Reduzca el trabajo pesado con otras herramientas.....	43
4.7 Música, maestro.....	44

4.8 Ejecutando reclusos ANSI.....	46
-----------------------------------	----

/ y

## Capítulo 5

### Uso del DEBUG para preguntas interactivas

5.1 El depurador, un tipo duro, que trata a los archivos como series de bytes.....	48
--	----

5.2 "Cómo desnudar a un programa?.....	48
--	----

5.3 Ensamblador para inmortales.....	52
--------------------------------------	----

## Capítulo 6

### Códigos de control para impresora

6.1 El lenguaje de la impresora no es nada vulgar.....	55
--	----

6.2 Los experimentos del doctor Escape.....	56
---	----

6.3 El código ASCII 12 es hembra.....	60
---------------------------------------	----

## Capítulo 7

### M s all del DOS 5

7.1 La batalla de los titanes del DOS.....	62
--	----

7.2 "MS-DOS 6? *No es para tanto!.....	62
--	----

7.3 Novell DOS 7, la última frontera.....	73
---	----

7.4 PC-DOS 6.3, recogiendo las migas.....	76
---	----

7.5 4DOS, todo lo que siempre quiso del DOS y gratis.....	77
---	----

7.6 El legado del DOS.....	81
----------------------------	----

## Capítulo 8

### Ejemplos, trucos y t,cnicas para los archivos batch

8.1 La prueba de fuego.....	82
-----------------------------	----

8.2 Señor batch, ¿me da fuego, por favor?.....	87
8.2.1 Desactivando el eco para todas las versiones del DOS.....	87
8.2.2 Borrar ficheros validando el mensaje de confirmación.....	88
8.2.3 Bucles concatenados.....	88
8.2.4 Comprobar la existencia de un directorio.....	89
8.2.5 Comprobando si un programa está en el directorio o path actual.....	89
8.2.6 Añadir un nuevo directorio al path.....	89
8.2.7 Comparando dos ficheros.....	89
8.2.8 Comprobando parámetros con caracteres comodines.	90
8.2.9 Protección para los ficheros batch.....	90
8.2.10 Cómo obtener el primer carácter de un parámetro	91
8.2.11 Algo más acerca de los parámetros.....	91
8.2.12 Cómo redefinir algunas teclas.....	92
8.2.13 Prompts de gran vistosidad.....	92
8.3 Una obra maestra.....	94
8.4 A toda potencia con el DOS.....	97

Apéndice.....	99
---------------	----

/ ii

## Introducción

-----

Cuando comencé a escribir el presente libro electrónico sobre técnicas de programación

de archivos batch, nunca pens, que se convertiría en un informe de esta extensión. La verdad es que hay mucho que decir acerca de los archivos de proceso por lotes. Lo cierto es que ide, estas páginas como una serie de artículos para una desaparecida publicación informática.

Pero a medida que fuí escribiendo página tras página, probando ejemplos una y otra vez, y buscando información indocumentada sobre el DOS, pens, en la posibilidad de convertir el informe en un libro que acompañase a una serie de utilidades y herramientas imprescindibles para el programador y usuario avanzado de los archivos batch. Así nacieron las UTILIDADES+2, una recopilación de utilidades capaces de convertir los ficheros batch en un potente compilador que no precisa grandes conocimientos de lenguajes de programación.

Así pues, junto al libro electrónico, encontrar los tan anhelados ficheros: una selección de programas propios y de dominio público. El disquete dispone de un fichero de instalación en el disco duro y un archivo de documentación sobre los servicios que presta cada fichero.

Aprender es sólo cuestión de tiempo. Lo digo porque soy el primero en reconocer que no estoy ante un manual dedicado a las órdenes del DOS, apto para todo tipo de usuarios. He de confesar que se requiere conocer el conjunto de órdenes que el DOS utiliza para comunicarse con el usuario, es decir, es obligatorio poseer ciertos conocimientos prácticos. De modo que no se llame a engaño. Si acaso sólo se le exige un nivel básico; es cuestión de hacer pruebas con su sistema operativo, comprobando en la práctica que hace y que, no hace cada comando. \*Así empezamos todos!

Una vez está, desenvuelto en la práctica de los ficheros batch, encontrar que este libro electrónico es un manual imprescindible para hacer sus primeros pinitos. Dispondrá de todo lo necesario para crear programas con menús, gráficos, entradas de datos y sonido. Y le aseguro que desarrollar un programa de este tipo es muy sencillo, tanto que con media docena de líneas de código ya tendrá una aplicación. A menudo le ocurrirá que cuando muestre los resultados de su programa a sus amigos, estos le mirarán con incredulidad diciendo que no es posible que un fichero batch haga todo eso. Deje que otros sigan programando en Clipper o C, si bien usted, con escasa dedicación, habrá desarrollado en un corto espacio de tiempo lo mismo que esos otros han tardado en realizar varios días.

Carlos Mesa

Barcelona, enero 1993 / octubre 1994

Cómo utilizar este libro

-----

Este libro trata de los ficheros batch con la potencia de los ficheros batch.

Capítulo 1

## ORGANIZACION Y PERSONALIZACION EN EL ARRANQUE DEL ORDENADOR

---

### 1.1 A propósito de los archivos por lotes

Sería del máximo atrevimiento afirmar que los archivos de procesamiento por lotes son un lenguaje de programación. Pero con las herramientas adecuadas, un conocimiento exhaustivo del DOS y un gran derroche de imaginación -aquí que han perdido los programadores ante la facilidad de los nuevos compiladores- se pueden construir extraordinarios programas, impensables para quienes desconocen las posibilidades que brindan las órdenes del sistema operativo. El siguiente libro electrónico -complemento de las UTILIDADES+2- aborda en profundidad el estudio de tales órdenes, ofreciendo soluciones, trucos, herramientas, mostrando programas, estudiando ensamblador, y explicando sencillas propuestas de programación gráfica.

### 1.2 El primer fichero batch es el de arranque

Muchos usuarios personalizan su sistema cambiando el indicador del DOS. Se entiende que, por defecto, el DOS presenta el indicador de la unidad de disco con la que estemos trabajando más el símbolo mayor (>).

El comando PROMPT dispone de una serie de combinaciones a las que se denomina metacaracteres. Cada metacaracter va precedido del signo de dólar (\$). Al ejecutarse el PROMPT, se presenta el texto que tiene asociado cada metacaracter o el mensaje que nosotros le hayamos indicado.

Cuando se arranca el ordenador, el DOS busca en el directorio raíz un archivo AUTOEXEC.BAT. Si existe, el DOS lo lee, y ejecuta línea por línea los comandos contenidos, desde el primero al último. Si el DOS no puede encontrar el fichero AUTOEXEC.BAT en el directorio raíz, aparece un mensaje de derechos de autor y ejecuta los comandos DATE y TIME.

Un ejemplo de archivo AUTOEXEC.BAT sería el siguiente:

Código:

```
@ECHO OFF
```

```
PATH C:\DOS
```

```
PROMPT $P$G
```

Como puede verse, el PROMPT personaliza el sistema haciendo aparecer el disco y directorio actuales más el símbolo mayor. La mayoría de usuarios presenta en el indicador del DOS el directorio en el que se encuentran, siendo esta la orden más frecuente de un archivo AUTOEXEC.BAT. No obstante, hay usuarios que se complican mucho más el PROMPT e incluso Digital Research, en su DR-DOS 6.0, agregaba un mensaje en el PROMPT durante el proceso de instalación.

Resulta muy difícil recordar todos los metacaracteres.

Hay que acudir al manual del DOS o a un libro; y, por cierto, he de decir que no siempre

est n contemplados todos ellos.  
Para compensarlo, he aquí la lista:

#### Metacaracter Texto asociado al mismo

##### Código:

\$b	
\$d	Fecha del sistema
\$e	Car cter de escape (al que mas adelante me referir, como parte basica de la programación gráfica y ANSI)
\$g	>
&h	Caracter de retroceso
\$l	<
\$n	Disco actual
\$p	Disco y directorio actuales
\$t	Hora actual del sistema
\$v	Versión actual del sistema
\$_	Vuelta de carro, salto de línea
\$q	=
\$texto	texto

Este último agrega el mensaje que nosotros deseemos hasta un m ximo de 127 caracteres, pues ,ste es el m ximo permitido en cada orden-, personalizando de ese modo nuestro sistema.

Con todo, este comando sirve adem s para simular mensajes con preguntas simples al usuario. Comprob,moslo. Supongamos el siguiente listado:

##### Código:

```
@ECHO OFF

PROMPT $P$G

IF "%1"==" " GOTO ERROR

IF NOT EXIST %1 GOTO ERROR

COPY %1 C:\BACKUP > NUL

ECHO Copiando el fichero %1 en C:\BACKUP

GOTO FIN

:ERROR

PROMPT No ha indicado el programa a copiar o ,ste no existe

:FIN
```

Este fichero que se puede llamar GRABA.BAT, copia el fichero introducido como parámetro de GRABA en el directorio C:\BACKUP. Lo primero que hace el programa es comprobar si se ha introducido algún parámetro y, luego, si existe el fichero que se desea copiar. Si no es así, la orden GOTO envía el control a la etiqueta :ERROR (esto es así, porque los archivos de procesamiento por lotes sólo pueden ejecutar una orden válida en cada línea). El PROMPT emite un mensaje que no desaparece de pantalla, por mucho que se intente, hasta que no se introduzca el nombre del programa con el parámetro adecuado. Cuando esto sucede, el primer PROMPT restablece el indicador del DOS, comprueba de nuevo el parámetro, y copia el fichero al directorio establecido. Con NUL redireccionamos el mensaje del DOS de copiado a un dispositivo nulo, y ECHO hace aparecer el nuestro (en castellano). El próximo GOTO envía la secuencia a la etiqueta :FIN, saltándose el PROMPT que contiene el mensaje de error.

Un aviso. Se puede evitar el mensaje de salida de algunas ordenes, redireccionando la salida a un dispositivo nulo, pero existen algunas órdenes que no se ejecutan, de este modo, correctamente. Un ejemplo típico es la orden DEL. Si no encuentra el fichero a borrar, sigue apareciendo un mensaje; en este caso, el de "Fichero no encontrado". Con un poco de inventiva, se pueden soslayar estos problemas. Por ejemplo, el caso anterior se soluciona con un IF EXIST FICHERO DEL FICHERO.

Volviendo al tema; como puede verse, el PROMPT, sabiéndolo usar, puede ser de utilidad en algunas circunstancias.

Pero ya que estamos con las órdenes más frecuentes de un archivo AUTOEXEC.BAT, conviene hacer un inciso sobre uno de los comandos de los que pocas veces nos servimos: PRINT.

Por defecto, siempre que queremos imprimir un fichero de texto ASCII desde la línea de comandos, escribimos PRINT seguido del fichero de texto. El problema está en que, dependiendo del tipo de impresora y del trabajo a realizar habitualmente, obtenemos resultados desastrosos, como colapsos y desajustes del papel por falta de tiempo entre la impresión de archivos. Con los distintos parámetros que admite PRINT se pueden indicar preferencias, como la impresora utilizada (PRN para las impresoras en paralelo y COM1 para las impresoras serie), el tamaño de la cola de impresión (hasta 32 archivos) y varios indicadores que establecen el tiempo empleado en la impresión. Por consiguiente, es necesaria la tabla antes de proseguir con más aclaraciones.

[Parámetros] [Función]

/b Especifica el tamaño de la memoria de impresión.

/c Cancela los nombres de los comandos cuando el DOS los ejecuta como parte de un archivo batch. Pero esto no es todo. En muchas ocasiones querrá mejorar la apariencia de la pantalla y dejar una línea en blanco entre los distintos mensajes que ha asociado a la orden ECHO. Para dejar una línea en blanco tiene que utilizar este comando seguido de un punto (ECHO.). Si no emplea el DOS 5 (o superior) y desea conseguir una línea en blanco, tendrá que usar el carácter 255 del juego de caracteres extendido ASCII (ALT-255, después de la orden ECHO). Por otro lado, la orden ECHO admite parámetros. Cuando aparece un carácter %número que identifica los parámetros de los ficheros por lotes, se sustituye por el valor del parámetro al que corresponde. Para



que lo veamos mas claro, si el primer par metro de un fichero por lotes es C:\NORTON, para la siguiente orden:

Código:

```
ECHO Usando el directorio %1
```

... obtendríamos el siguiente mensaje:

```
Usando el directorio C:\NORTON
```

Tenga en cuenta que los símbolos especiales deben ir encerrados entre comillas (ej.

ECHO Utilice el car cter '>').

### 2.1.3 Orden REM

La orden REM sirve para introducir comentarios internos dentro de un fichero por lotes. Suele identificar la cabecera de un archivo batch con información complementaria acerca del programador y del programa. Y ya que se hace mención del principio de un fichero, vayamos al final de este. Podemos interrumpir la ejecución de un archivo con la secuencia Control-C (F6). Veamos un ejemplo:

Código:

```
REM Programa HORA.BAT
```

```
REM Por Pepe Botella
```

```
TIME
```

```
@ECHO "Es correcta la hora?
```

```
@ECHO Si no es así, salga del programa y c mbiela
```

```
^C
```

```
DATE
```

Cuando ejecute el programa, este le dar, el informe de la hora actual. Si no es correcta, cuando aparezca el mensaje ""Terminar el procesamiento por lotes?", teclee S. Si la hora es correcta, tecleando N, pasar la acción al siguiente comando, que le informar del día actual.

### 2.1.4 Orden PAUSE

Evidentemente, existe una orden para detener temporalmente la ejecución de un fichero por lotes hasta que se pulsa una tecla: PAUSE. Si le asociamos un mensaje, presentare este en pantalla. Pero cuidado, ECHO OFF suprime la aparición del mensaje de usuario que se haya asociado con PAUSE. Para habilitar la aparición de los mensajes debe activar y desactivar el ECHO antes y después de PAUSE.

Código:

```
@ECHO OFF
```

```
VER
```

```
VOL
```

```
ECHO ON
```

```
PAUSE "Quiere seguir con el proceso?"
```

```
ECHO OFF
```

Otro truco consiste en avisar al usuario cuando el proceso haga una pausa, haciendo sonar un pitido en el altavoz del ordenador. Para ello, simplemente debemos añadir la secuencia Control-G. Tomaremos como referencia el ejemplo anterior:

**Código:**

```
@ECHO OFF
```

```
VER
```

```
VOL
```

```
ECHO ON
```

```
PAUSE ^G ^G "Quiere seguir con el proceso?"
```

```
ECHO OFF
```

### 2.1.5 El símbolo % identifica a un par metro

El símbolo %número sirve para identificar dentro de un fichero por lotes, los distintos par metros que recibe del archivo batch al ejecutarse. La constante NUMERO ha de ser un número decimal entero comprendido entre 0 y 9 (ambos incluidos). %1 sería el primer par metro, %2 el segundo par metro, %3 el tercero..., mientras que %0 sería el nombre (con o sin trayectoria) del archivo batch.

Si crea un archivo por lotes que haga uso de tres par metros, piense que algunas órdenes no se pueden usar sin par metros. Por ejemplo, si crea un programa que haga tres TYPE de los par metros asociados al fichero batch y luego sólo recibe dos, el programa generará un error, pues no se puede utilizar la orden TYPE sin par metros.

Además del par metro %NUMERO, el DOS admite par metros con nombre, entendidos como variables de entorno para archivos batch. Cuando el sistema encuentra un par metro con nombre encerrado entre dos signos de tanto por ciento: %NOMBRE%-, explora el entorno buscando una variable con ese mismo nombre.

El comando del DOS, SET, asigna datos a la variable de entorno. Pongamos un caso: si queremos asignar un valor a la variable %FILE%, como por ejemplo, el contenido del AUTOEXEC.BAT, debemos escribir lo siguiente: SET

FILE=AUTOEXEC.BAT. Si queremos comprobar el uso de las variables de entorno, simplemente escribiremos SET sin argumentos. El DOS presentará la lista de variables de entorno en ese momento:

**Código:**

```
C>SET
```

```
COMSPEC=C:\DOS\COMMAND.COM
```

```
PATH=C:\DOS
```

```
FILE=AUTOEXEC.BAT
```

Por el contrario, para eliminar el valor de una variable de entorno, basta con emplear el comando SET de esta forma:

Código:

```
SET FILE=
```

Cuando el DOS encuentra una variable de entorno, recorre el entorno en busca de ,sta. Si no existe, asigna a %NOMBRE% una cadena vacía. En este caso, el sistema operativo crea el archivo en el directorio actual. Si existe, desaparece el valor anterior y permanece el nuevo que se le pueda dar.

Con todo lo aprendido, veamos un ejemplo.

Código:

```
@ECHO OFF
```

```
REM EXTRAE.BAT
```

```
SET EXTRAE=
```

```
FOR %%A IN (/%1) DO IF "%EXTRAE%"==" " SET EXTRAE=%%A
```

```
ECHO %EXTRAE%
```

El programa EXTRAE.BAT, como su propio nombre indica, extrae la primera letra de cualquier palabra que le hayamos pasado como par metro al programa, visualizando el resultado en la pantalla. Para adentrarnos en su explicación, si no se tiene muy claro cómo funcionan las órdenes IF y FOR, habra de saltarse el siguiente párrafo, continuar con la lectura y volver luego a estas explicaciones cuando los conceptos hayan sido esclarecidos.

La primera línea después de ECHO borra la variable de entorno si la hubiese. Fíjese en la barra espaciadora del conjunto entre parentesis. No es ningún separador; se trata de una característica no documentada del MS-DOS. Con ella se logra que en la primera pasada del FOR, la variable %%A tome el valor de la primera letra pasada como par metro, mientras que en la segunda, se tome el valor del resto de las letras del parametro. Es decir, la barra espaciadora separa el primer carácter de una cadena del resto. En nuestro ejemplo, durante la primera pasada, SET coloca la primera letra en la variable de entorno, mientras que en la segunda -y como se ha mencionado-, desaparece el valor anterior y permanece el nuevo, que ser el resto de la cadena a visualizar con ECHO.

## 2.1.6 Orden CALL

Cuando se ejecuta un segundo archivo batch desde un primer archivo batch, se debe llamar a este segundo con una orden de llamada -COMMAND o CALL-. De lo contrario, el DOS deja de ejecutar los comandos del primer archivo batch cuando se ha completado la ejecución del primero.

Si se emplea DOS 3.3 o posterior se debe utilizar el comando CALL -al que se le pueden añadir par metros después del archivo batch-.

Código:

```
@ECHO OFF
```

```
VER
```

VOL

CALL FORMATEA %1

FORMATEA.BAT es un archivo batch que no existe. Continúa la ejecución del archivo batch con el siguiente comando. T,ngase en cuenta que el DOS solo admite una orden válida para cada línea de un fichero por lotes. Si quiere ejecutar dos o m s órdenes, se han de agrupar estas detras de una etiqueta y utilizar una orden 'GOTO :etiqueta' detras de la condición. No obstante, si DOSKEY (programa de MS-DOS 5.0 y superior) fue cargado (en el AUTOEXEC.BAT), puede introducir varias ordenes separando las por el caracter ' ' (CTRL-T). El parametro NOT, si se coloca, invierte la condición.

La condición 'IF [NOT] cadena1==cadena2 orden' comprueba si dos cadenas de caracteres son identicas; caso de que lo sean, ejecuta el comando del DOS siguiente. La comparación de las dos cadenas de caracteres se hace letra a letra. Las cadenas han de coincidir exactamente, incluyendo mayúsculas y minúsculas (no importan los espacios en blanco).

Si no se especifican dos cadenas de caracteres, en la línea de comandos aparece un mensaje de error. Este error es muy comun cuando se comprueba un par metro del archivo batch:

Código:

```
IF %1==MES GOTO BALANCE
```

Si no se especifica un valor para el par metro %1, la condición compararía una única cadena, pues %1 desaparece, resultando un error de sintaxis. Para solucionarlo, basta con colocar entre comillas las cadenas de caracteres:

Código:

```
IF "%1"=="MES" GOTO BALANCE
```

Por esta misma regla de tres, podemos decir que para comprobar si se ha introducido un par metro o no, basta con escribir en el fichero por lotes lo siguiente:

Código:

```
IF "%1"==" " orden
```

La condición 'IF [NOT] ERRORLEVEL número\_código orden' examina la situación de terminación de los comandos y continua con el proceso de la condición de acuerdo con los valores obtenidos. La situación de terminación de algunos comandos se basa en los códigos de salida generados por los comandos del DOS. Normalmente, dichos códigos sirven para describir la causa del error que ha provocado la finalización del programa.

No obstante, solo hay diez ordenes estandar del DOS que devuelven codigos de salida para ser procesados con la orden IF ERRORLEVEL.

Aquí esta la tabla que resume los valores de situación de salida y sus significados:

Comando Valor Significado

BACKUP 0 Realizada copia de seguridad

BACKUP 1 No existen archivos para copiar

BACKUP 2 Conflicto de ficheros compartidos

BACKUP 3 Abortado por el usuario con CTRL-C

BACKUP 4 Error fatal. Copia de seguridad incompleta

DISKCOMP 0 Comparación correcta

DISKCOMP 1 Comparación incorrecta (discos distintos)

DISKCOMP 2 Abortado por el usuario con CTRL-C

DISKCOMP 3 Error de lectura o escritura

DISKCOMP 4 Memoria insuficiente, unidad no v lida

DISKCOPY 0 Copia realizada con ,xito

DISKCOPY 1 Error de lectura o escritura

DISKCOPY 2 Abortado por el usuario con CTRL-C

DISKCOPY 3 Imposible leer disco fuente o escribir

DISKCOPY 4 Memoria insuficiente, unidad no v lida

FORMAT 0 Formateo con ,xito

FORMAT 3 Abortado por el usuario con CTRL-C

FORMAT 4 Error fatal. Operación incompleta

FORMAT 5 Respuesta N a "Formatear otro disco(S/N)?"

GRAFTABL 0 Tabla de caracteres cargada correctamente

GRAFTABL 1 Se substituyó la tabla previamente cargada

GRAFTABL 2 Error de fichero

GRAFTABL 3 Par metro incorrecto

GRAFTABL 4 Versión del DOS incorrecta

REPLACE 0 Actualización de fichero realizada

REPLACE 2 Ficheros origen no encontrados

REPLACE 3 Trayectoria no encontrada

REPLACE 5 Archivo destino sólo de lectura

REPLACE 8 Memoria insuficiente

REPLACE 11 Comando incorrecto

REPLACE 15 Unidad de disco no v lida

REPLACE 22 Versión del DOS incorrecta (no funciona para MS-DOS)

RESTORE 0 Ficheros restaurados correctamente

RESTORE 1 Ficheros a restaurar no encontrados

RESTORE 2 Conflicto debido a archivo compartido

(no funciona para MS-DOS)

RESTORE 3 Abortado por el usuario con CTRL-C

RESTORE 4 Error fatal. RESTORE incompleto

Pero es que el MS-DOS 5 proporciona, además, nuevos códigos de salida para algunos comandos:

Comando Valor Significado

KEYB 0 Fichero de teclado cargado correctamente

KEYB 1 Código de teclado incorrecto

KEYB 2 KEYBOARD.SYS no encontrado o incorrecto

KEYB 3 No se puede crear la tabla

KEYB 4 Error al comunicarse con dispositivo CON

KEYB 5 La p gina de códigos no ha sido preparada

KEYB 7 Versión del DOS incorrecta

KEYB 8 Memoria insuficiente

SETVER 0 SETVER ejecutado correctamente

SETVER 1 Par metro no v lido

SETVER 2 Nombre de archivo incorrecto

SETVER 3 Memoria insuficiente

SETVER 4 Número de versión incorrecto

SETVER 5 Entrada no hallada en la lista de SETVER

SETVER 6 No se encuentra el archivo SETVER.EXE

SETVER 7 Unidad incorrecta

SETVER 8 Demasiados par metros

SETVER 9 Falta algún par metro

SETVER 10 Error al leer la tabla de SETVER.EXE

SETVER 11 La tabla de SETVER es incorrecta

SETVER 12 SETVER no soporta una lista de programas

SETVER 13 Espacio insuficiente en la lista

SETVER 14 Error de escritura

XCOPY 0 Copia realizada correctamente

XCOPY 1 No hay ficheros para copiar

XCOPY 2 Abortado por el usuario con CTRL-C

XCOPY 4 Error de inicialización

XCOPY 5 Error al escribir los ficheros (INT 24)

Mientras que MS-DOS 6 utiliza los siguientes códigos de error.

Comando Valor Significado

CHOICE 0 Interrumpido por el usuario con CTRL-C.

CHOICE 1 Aceptación del primer carácter dado después del par metro C:teclas

CHOICE 2 Aceptación del segundo carácter dado después del parámetro C:teclas, y así sucesivamente

CHOICE 255 Error general

DEFRAG 0 Desfragmentación correcta

DEFRAG 1 Error interno de DEFRAG

DEFRAG 2 Disco lleno

DEFRAG 3 Desfragmentación abortada por usuario

DEFRAG 4 Error general

DEFRAG 5 Error de lectura en cluster

DEFRAG 6 Error de escritura en cluster

DEFRAG 7 Error al modificar la FAT

DEFRAG 8 Error de memoria

DEFRAG 9 No hay memoria suficiente

DELTREE 0 Eliminación de ficheros correcta

EMM386 EMMXXXX0 Soporte de memoria expandida activado (este código constituye una excepción, al tratarse en realidad de una cadena)

Examinemos unos ejemplos de utilidad. El primero, DESARC.BAT descripta los ficheros con extensión .LZH (del popular programa compactador de dominio público).

Código:

```
FOR %%A IN (*.LZH) DO LHA U %%A
```

El segundo, ORDENA.BAT, emplea la orden IF EXIST para borrar un archivo donde se guardan los datos de los ficheros con extensión que especifiquemos como parámetro de ORDENA.BAT.

Después, el programa usa ECHO para direccionar y adicionar con '>>' la lista de pantalla al archivo ORDENA.DAT. Una vez creado el archivo, SORT ordena su contenido, presentando los nombres en pantalla. Por último, se borra el archivo .DAT que ya no se necesita.

Código:

```
@ECHO OFF
```

```
IF EXIST ORDENA.DAT DEL ORDENA.DAT
```

```
FOR %%A IN (%1) DO ECHO %%A >> ORDENA.DAT
```



```
SORT < ORDENA.DAT
```

```
DEL ORDENA.DAT
```

```
*****
```

```
* > Redirección de salida para un archivo *
```

```
*****
```

```
* >> Adición de un fichero *
```

```
*****
```

```
* < Redirección de entrada para un archivo *
```

```
*****
```

El tercero, SALIR.BAT, sale de un supuesto menú creado por un fichero por lotes y presenta un mensaje por pantalla.

En cualquier caso, el comando FOR se ejecuta tantas veces como elementos hay en conjunto.

Código:

```
@ECHO OFF
```

```
CLS
```

```
FOR %%A IN (0 1 2 3 4 5 6 7 8 9 10 11) DO ECHO.
```

```
ECHO GRACIAS POR TRABAJAR CON NUESTRO PROGRAMA
```

```
FOR %%A IN (0 1 2 3 4 5 6 7 8 9 10) DO ECHO.
```

Como puede verse a través de estos ejemplos, el comando FOR añade una enorme flexibilidad a los archivos batch, pero lo que muchos no saben es que también puede ejecutarse desde la línea de comandos, sin necesidad de crear una línea para un fichero por lotes. En este caso, las variables han de escribirse con un sólo carácter '%', es decir %LETRA.

## NUEVOS COMANDOS INTERACTIVOS PARA LOS FICHEROS POR LOTES

-----

### 3.1 Norton, el rey Midas de la informática, proporciona un nuevo comando

Podemos extraer como una primera conclusión que: los archivos batch son herramientas muy potentes para mejorar la eficiencia del trabajo con DOS. Desgraciadamente, el DOS no proporciona los comandos necesarios para trabajar en tiempo real, basándose en la entrada desde el teclado. Esta omisión limita seriamente las posibilidades de los ficheros por lotes.

No obstante, podemos hacer uso de programas externos, como

Ampliar Batch, de las Utilidades Norton, donde se añaden nuevas posibilidades. A pesar de esto, como se ver mas adelante, el DOS guarda algunos trucos para realizar algunos de los trabajos que se esperan de estas utilidades.

El comando AB (Ampliar Batch) de las Utilidades Norton (BA en la versión americana) obtiene nuevos modos de funcionamiento para los ficheros por lotes. Dado que estas utilidades son muy conocidas por los usuarios, conviene estudiar AB con detalle, a modo de ejemplo de expansion con otros comandos externos del DOS.

AB hace uso de subcomandos y par metros para estos subcomandos. Así, el subcomando AB ?, ofrece pantallas de ayuda con texto acerca del modo de trabajo de Ampliar Batch.

Los subcomandos de AB son: PREGUNTA, SONIDO, RECUADRO,

CLS, PAUSA, GOTO, IMPCAR, FILACOL, VENTANA y AP. Todos estos subcomandos manejan, a su vez, una serie de par metros que ahora conoceremos.

Cabe recordar que para poder utilizar el comando AB desde cualquier directorio, aparte de la instalación de las Utilidades Norton, se ha de añadir el nombre del directorio en el que se encuentran las utilidades a la instrucción PATH del AUTOEXEC.BAT.

Por cierto que si queremos añadir un PATH al archivo AUTOEXEC.BAT desde un fichero por lotes de instalación, solo hay que combinar un fichero con otro. Por ejemplo, el programa INSTALA.BAT añade una línea al AUTOEXEC.BAT.

**Código:**

```
@ECHO OFF
```

```
CLS
```

```
REM Programa INSTALA.BAT
```

```
ECHO "INSTALA [unidad origen:] [unidad destino:]?"
```

```
ECHO Si no es correcto pulse CTRL-C
```

```
PAUSE > NUL
```

```
%2
```

```
CD %2\
```

```
COPY %2\AUTOEXEC.BAT+%1\AYUDA %2\AUTOEXEC.BAT
```

```
MD INSTALL
```

```
CD INSTALL
```

```
COPY %1\*.*
```

El fichero AYUDA contendría lo siguiente:

Código:

```
PATH=C:\NORTON
```

El programa INSTALA.BAT envía una pregunta acerca de los parámetros dados a INSTALA. El fichero por lotes detiene el proceso y direcciona el mensaje de salida de la orden PAUSE a un dispositivo nulo. El parámetro %2 obliga al programa a activar esa unidad, mientras que CHDIR (CD) cambia el directorio activo al directorio raíz.

La orden COPY utilizada de esta forma, combina el fichero de texto AYUDA -el cual contiene una línea PATH- con el AUTOEXEC.BAT, añadiendo esta última línea al final.

Las otras tres órdenes abren un nuevo directorio y copian el programa que requiere de este PATH al directorio especificado.

1. El subcomando PREGUNTA hace una pregunta y espera una respuesta. Cuando se llama al comando PREGUNTA aparece el texto de la pregunta con varias opciones. Se responde a PREGUNTA pulsando una tecla de la lista de contestaciones.

Después de seleccionar una de las teclas de la lista, PREGUNTA devuelve la información sobre la tecla elegida con un ERRORLEVEL. Así, la primera opción correspondería a ERRORLEVEL 1, la segunda a ERRORLEVEL 2, la tercera a ERRORLEVEL 3, etc.

El programa batch puede bifurcarse a diferentes etiquetas, según el ERRORLEVEL indicado.

```
AB PREGUNTA "Ejecutar (W)ordPerfect, (d)BASE o (S)alir", wds
```

```
IF ERRORLEVEL 3 GOTO SALIR
```

```
IF ERRORLEVEL 2 GOTO DBASE
```

```
IF ERRORLEVEL 1 GOTO WP
```

```
:WP
```

```
CD\WP51
```

```
WP
```

```
GOTO SALIR
```

```
BASE
```

```
CD\DBASE
```

```
DBASE
```

GOTO SALIR

:SALIR

La sintaxis de este subcomando es la siguiente:

AB PREGUNTA MENSAJE [,TECLAS] [DEFECTO=TECLA] [ESPERA=N]  
[AJUSTE=N] [COLOR]

Donde TECLAS correspondería a las iniciales de la lista de opciones, DEFECTO=TECLA a la tecla de respuesta, ESPERA=N al tiempo en segundos para esperar antes de tomar como respuesta la tecla, AJUSTE=N al ajuste del valor devuelto por ERRORLEVEL (hasta un máximo de 254), y COLOR al parámetro que colorea el texto del mensaje.

La sintaxis de todas las ordenes se encuentran convenientemente detalladas en la guía del usuario de las utilidades Norton. Las explicaciones aquí referidas han de tomarse como ejemplo de las posibilidades que abren otros comandos externos del DOS.

Existe otro modo, todo para formular preguntas al usuario el controlador de dispositivo. Para remediarlo, puede arrancar un controlador ANSI.COM. El ANSI ejecutable no es tan difícil de encontrar.

Una primera versión producida por Jordi Mas y este servidor ha sido incluida en la recopilación de utilidades adjunta al libro.

Se pueden emplear distintos colores, como un fondo de color, o atributos como vídeo inverso o texto parpadeante con este subcomando. También se puede utilizar AP junto con SONIDO para producir efectos interesantes. Los resultados obtenidos varían en función de la pantalla gráfica que tengamos; habrá que experimentar para comprobar las mejores combinaciones. Los programadores de ficheros por lotes habrán de probar en distintos ordenadores antes de decidirse por incluir una pantalla así creada en su programa.

La sintaxis del subcomando es la siguiente:

Código:

AB AP [INTENSIDAD] [PRIMER-PLANO] [EN FONDO] [/N] [/CLS]

Los valores para intensidad son Brillo, Negrita o Parpadeo. Los valores de primer-plano y fondo son Blanco, Azul, Negro, Rojo, Verde, Magenta, Cyan y Amarillo. Todas

estas opciones pueden abreviarse con las tres primeras letras.

Puede prescindirse de todas las opciones y escribirse unos valores generales, que son: Normal, Inverso y Subrayado.

El parámetro /N no varía el color del borde, mientras que el parámetro /CLS borra la

pantalla despu,s de cambiar los atributos de color.

Para las versiones americanas de las Utilidades Norton, consulte el manual de Symantec donde encontrar la correspondencia de los subcomandos en castellano con los similares en lengua inglesa.

### 3.2 El anonino LETRA.COM

Como se esta viendo, la capacidad de funcionamiento en modo interactivo de los ficheros batch, es bastante potente, desmintiendo las palabras de aquellos que, sin entender demasiado, pretenden desacreditar el potencial de los ficheros por lotes. A ello, pueden achacar que se trata de programas donde, una vez comenzado su proceso, es imposible interrumpir su funcionamiento. Bueno, ya se ha visto que esto no es así.

Comandos como AB PREGUNTA de las Utilidades Norton requiere de información externa, así como hay trucos con el DOS 5 y 6 o el DEBUG que permiten esto mismo. Con todo, existe un programa de dominio publico, incluido en las UTILIDADES+2, que utiliza el concepto de ERRORLEVEL para crear ficheros interactivos. El autor de este programa es desconocido. Se trata de LETRA.COM.

La unica información disponible sobre este programa, es el ejemplo que viene a continuación.

```
1 @ECHO OFF
```

```
2 ECHO PULSE (A-C) PARA VER LOS MENSAJES O 'S' PARA SALIR
```

```
3 LETRA ABCS
```

```
4 REM Los argumentos de LETRA (por ejemplo 'ABCS' deben darse
```

```
5 REM en mayusculas.
```

```
6 REM Notese que el usuario puede introducir tanto mayusculas
```

```
7 REM como minúsculas en su respuesta.
```

```
8 IF ERRORLEVEL 19 GOTO FIN
```

```
9 IF ERRORLEVEL 3 GOTO MENSAJEC
```

```
10 IF ERRORLEVEL 2 GOTO MENSAJEB
```

```
11 IF ERRORLEVEL 1 GOTO MENSAJEA
```

```
12 REM Obs,rvese que ERRORLEVEL se define de acuerdo con el
```

valor

13 REM ASCII del caracter en mayusculas menos 64 (por ejemplo

14 REM 'A' = (65-64) = 1)

15 :MENSAJEA

16 ECHO ^G

17 ECHO Se ha pulsado la tecla A

18 GOTO FIN

19 :MENSAJEB

20 ECHO ^G

21 ECHO Se ha pulsado la tecla B

22 GOTO FIN

23 :MENSAJEC

24 ECHO ^G

25 ECHO Se ha pulsado la tecla C

26 :FIN

Fíjese que la línea "LETRA ABCS" define las respuestas permitidas. Cualquier intento de teclear otra letra congela el fichero hasta obtener la respuesta definida. El esquema de ERRORLEVEL consiste en restar 64 del valor ASCII de cada respuesta permitida. Como se citó en el apartado correspondiente, ERRORLEVEL analiza el número devuelto mayor, desde el más alto hasta el más bajo. Combinando los errores devueltos por LETRA.COM con las bifurcaciones, se pueden desarrollar ficheros interactivos. En el ejemplo, cuando se pulsa la tecla correcta -que bien pudiera emitirse un mensaje con ECHO para dar a entender cuáles son las opciones de respuesta- se transfiere el proceso.

En otro ejemplo imaginario, una respuesta S devolvería 19 como ERRORLEVEL (83-64) y una respuesta N un ERRORLEVEL 14 (78-64). Piense en sus aplicaciones. Notese, por último, que la resta se hace del valor ASCII de las mayúsculas.

### 3.3 El otro DOS

El sistema operativo DR DOS 6, de Digital Research, utilizaba dos órdenes en los ficheros por lotes, no contempladas en MS-DOS. Puede valerse de ellas si dispone del sistema operativo o copiarlas directamente al sistema operativo de Microsoft. Las órdenes son: GOSUB y SWITCH.

GOSUB es similar a GOTO. Ambos se utilizan para transferir el control de la ejecución. La única diferencia estriba en que GOSUB debe incluir el comando RETURN para que el procesamiento continúe en la línea que sigue a GOSUB. Es similar en cuanto a características a la orden de Basic.

Veamos un ejemplo para añadir al AUTOEXEC.BAT:

**Código:**

```
IF EXIST *.BAK GOSUB BORRABAK
```

```
GOTO SIGUE
```

```
:BORRABAK
```

```
DEL *.BAK
```

```
RETURN
```

```
:SIGUE
```

El archivo AUTOEXEC.BAT comprueba en el directorio raíz si existe alguna copia de seguridad. Caso de que encontrase alguna, el procesamiento saltaría a la etiqueta :BORRABAK y borraría estas copias de seguridad. El comando RETURN devolvería el control a la línea siguiente que pasaría a la etiqueta :SIGUE, donde se continuarían las ordenes que uno desee del AUTOEXEC.

SWITCH es muy parecido a GOSUB, salvo que SWITCH permite conmutar entre distintas reas de trabajo del fichero por lotes. La orden es similar en cuanto a características a la de QBASIC/ QuickBASIC y a la de cualquier compilador de C. Veamos como actúa a través de un ejemplo:

- ECHO 1 = WORDPERFECT

```
ECHO 2 = dBASE
```

```
ECHO 3 = LOTUS 1-2-3
```

```
SWITCH 1, 2, 3
```

```
GOTO FIN
```

```
:1
```

```
CD\WP51
```

```
WP
```

```
RETURN
```

```
:2
```

```
CD\DBASE
```

```
DBASE
```

```
RETURN
```

```
:3
```

CD\LOTUS

LOTUS

RETURN

:FIN

Como puede ver, al principio del fichero se presentan tres opciones -que pueden ser tantas como uno quiera, hasta un total de nueve etiquetas-, visualizadas en pantalla merced al comando ECHO. SWITCH espera a que seleccione entre 1, 2 ó 3.

Una vez realizada la elección, SWITCH le lleva a la etiqueta adecuada, se ejecutan los comandos de ,sta y RETURN devuelve el control de la ejecución a la sentencia que sigue a SWITCH, en este caso un GOTO a la etiqueta FIN. También se podría haber enviado el control a una etiqueta al principio del programa, con lo que este entraría en un bucle infinito que sólo permitiría la ejecución de los programas indicados en las etiquetas.

Después de DR DOS 6, Novell absorbió el sistema operativo de Digital Research y lanzó Novell DOS 7, que compatibiliza las mismas ordenes interactivas de su antecesor. Si quiere saber más sobre este y otros sistemas operativos de reciente aparición, remítase a las posiciones de un array. El programa copia la cadena pasada como argumento o parámetro al fichero que se mencionó como primer parámetro. 'strcat' concatena la cadena batch, dándole esta extensión al nombre del fichero. Se crea un fichero y el bucle 'for' escribe cada parámetro en una línea del fichero batch, con la longitud precisa de cada cadena ('strlen'). Después de pasados todos los parámetros, el programa cierra el fichero.

Los que utilicen otro compilador, como Borland C++, habrán de obviar los mensajes fatídicos que señalan el tipo de dato del valor devuelto por la función. El programa se ejecuta correctamente, aunque con unas cuantas correcciones y variaciones se podría mejorar muchísimo. Lo dejo por cuenta del lector y usuario avanzado con conocimientos de C.

## Capítulo 4

### PROGRAMACION CON ANSI.SYS Y SECUENCIAS DE ESCAPE

-----

#### 4.1 ANSI.SYS es un controlador de dispositivo

ANSI.SYS es un controlador de dispositivo formado por un conjunto de rutinas de programación, que se encargan de supervisar la entrada desde el teclado y la salida por la pantalla. Cada vez que se transfieren datos por esta entrada o salida estándar, las rutinas del controlador de dispositivo analizan los datos buscando la presencia de una secuencia de caracteres especial identificadora de las órdenes ANSI. Hay que tener en cuenta que todos los caracteres de entrada y salida son comprobados por ANSI.SYS. Así, cuando encuentra esta secuencia de caracteres especial, se ejecuta la orden indicada en dicha secuencia, siempre y cuando tengamos cargado el controlador de



dispositivo en la memoria. Para resumirlo brevemente, ANSI.SYS es un filtro superpuesto a la entrada y salida de teclado y pantalla que solamente interviene cuando aparezca una secuencia de caracteres determinada que este pueda reconocer. Si no existe esta secuencia, el controlador filtra los caracteres y cede el control a otras rutinas que gestionan el teclado y la pantalla habitual.

Como se puede ver, ANSI.SYS es un programa que incrementa la capacidad del teclado y la pantalla del ordenador. La extensión .SYS quiere decir que el archivo pertenece al sistema operativo. Para cargarlo en memoria, debe insertarse una línea en el CONFIG.SYS, que es el archivo que emplea el DOS cada vez que el ordenador se arranca para configurar la memoria. La línea debe ser:

Código:

```
DEVICE=ANSI.SYS
```

O en todo caso, con la vía completa de acceso si el DOS se encuentra en un PATH diferente del raíz. También es posible cargar el controlador de dispositivo en la memoria superior (en un bloque UMB), ahorrando memoria convencional, con la orden `DEVICEHIGH=`.

Si a la línea ANSI.SYS le añadimos el parámetro /K, se fuerza a que el controlador use las funciones del teclado normal (84 teclas), aunque el sistema tenga presente un teclado ampliado (101 teclas). El parámetro /K proporciona compatibilidad con los programas que no soportan las funciones del teclado ampliado. Si se usa este parámetro debe introducirse además en el CONFIG.SYS la orden `SWITCHES = /K`.

El parámetro /X añadido a la línea ANSI.SYS permite que las teclas repetidas de los teclados ampliados puedan ser reasignadas independientemente (este tema lo veremos después con más detalle).

Una vez activado ANSI.SYS se puede realizar cualquier operación que tenga que ver con la pantalla y el teclado, como mover el cursor, borrar la pantalla, salvar y restaurar la posición del cursor, activar varios modos de vídeo, modificar los colores, emitir un parpadeo y hasta remapear todo el teclado asignando a cada tecla un valor diferente. He aquí la importancia del controlador de dispositivo, bien usado, para la programación de los ficheros por lotes.

Recuerde que todas estas ventajas sólo serán válidas, siempre y cuando se tenga instalado ANSI.SYS en el CONFIG.SYS.

Así que ahora viene una cuestión de suma importancia. Como programador de ficheros batch ¿cuando estoy seguro de que el usuario al que le haya vendido mi programa tenga instalado el controlador de dispositivo? La respuesta es \*que usted no lo sabe! Pero todo tiene solución. Mi amigo Jordi Mas y yo creamos un programa llamado EMUANSI.COM que sería el primero en ejecutarse en su fichero por lotes antes de hacer uso de las secuencias de escape. Si esto es así, no debe preocuparle que este o no instalado el controlador de dispositivo original. Esta herramienta se encuentra en las UTILIDADES+2.

Las órdenes de ANSI.SYS se identifican mediante las llamadas secuencias de caracteres

(secuencias de escape).

Estas secuencias de caracteres se envían mediante dos métodos: empleando el comando ECHO o el comando PROMPT. La orden TYPE seguido de un fichero de texto también acepta ordenes ANSI para desarrollos mas complejos.

Una secuencia de escape es una combinación única de caracteres que empieza con el carácter de escape ASCII 27. El carácter de escape no se introduce escribiendo las letras E, S y C, sino introduciendo el código ASCII 27 que aparece en pantalla simbolizado por una flecha a la izquierda ( ). Hace falta un segundo carácter, un simple corchete ([), cuyo valor es ASCII 91.

#### 4.2 Un artista del color en la pantalla

Para comenzar viendo el funcionamiento de ANSI.SYS, establezcamos los colores de una pantalla. La secuencia de escape que establece los colores es ESC[colorm donde ESC es el carácter de escape ASCII y color, un valor de los siguientes:

[Valor] [Color]

0 Letras blancas sobre fondo negro.

1 Alta intensidad o negrita. Azul para EGA color.

2 Verde para EGA color.

3 Cian para EGA color.

4 Subrayado (monitor monocromo). Azul sobre negro (monitor color). Rojo para EGA color.

5 Parpadeo de las letras. Magenta para EGA color.

6 Texto parpadeante (ruido). Marrón para EGA color.

7 Vídeo inverso. Blanco para EGA color.

8 Invisible (negro sobre negro). Gris para EGA color.

9 Azul claro para EGA color. Subrayado en alta intensidad para EGA monocromo.

10 Verde claro para EGA color. Alta intensidad para EGA monocromo.

11 Cian claro para EGA color. Alta intensidad para EGA monocromo.

12 Rojo claro para EGA color. Alta intensidad para EGA monocromo.

13 Magenta (granate) claro para EGA color. Alta intensidad para EGA monocromo.

14 Amarillo para EGA color. Alta intensidad para EGA monocromo.

15 Blanco en alta intensidad para EGA color. Alta intensidad para EGA monocromo.

16 Negro parpadeante para EGA color. Negro parpadeante, si el fondo es 7, para EGA monocromo.

17 0;133 0;135 0;137 0;139

F12 0;134 0;136 0;138 0;140

Ins \* 224;82 224;82 224;146 224;162

Inicio \* 224;71 224;71 224;119 224;151

ReP g \* 224;73 224;73 224;132 224;153

Supr \* 224;83 224;83 224;147 224;163

Fin \* 224;79 224;79 224;117 224;159

AvP g \* 224;81 224;81 224;118 224;161

Cursor arriba\* 224;72 224;72 224;141 224;152

Cursor izqda.\* 224;75 224;75 224;115 224;155

Cursor abajo \* 224;80 224;80 224;145 224;160

Cursor dcha. \* 224;77 224;77 224;116 224;157

Inicio 0;71 55 0;119 -

Cursor arriba 0;72 56 0;141 -

ReP g 0;73 57 0;132 -

Cursor izqda. 0;75 52 0;115 -

Cursor dcha. 0;77 54 0;116 -

Fin 0;79 49 0;117 -

Cursor abajo 0;80 50 0;145 -

AvP g 0;81 51 0;118 -

Ins 0;82 48 0;146 -

Supr 0;83 46 0;147 -

ImprPant - - 0;114 -

A 97 65 1 0;30

B 98 66 2 0;48

C 99 67 3 0;46

D 100 68 4 0;32

E 101 69 5 0;18

F 102 70 6 0;33

G 103 71 7 0;34

H 104 72 8 0;35

I 105 73 9 0;23

J 106 74 10 0;36

K 107 75 11 0;37

L 108 76 12 0;38

M 109 77 13 0;50

N 110 78 14 0;49

O 111 79 15 0;24

P 112 80 16 0;25

Q 113 81 17 0;16

R 114 82 18 0;19

S 115 83 19 0;31

T 116 84 20 0;20

U 117 85 21 0;22

V 118 86 22 0;47

W 119 87 23 0;17

X 120 88 24 0;45

Y 121 89 25 0;21

Z 122 90 26 0;44

1 49 33 - 0;120

2 50 64 - 0;121

3 51 35 - 0;122

4 52 36 - 0;123

5 53 37 - 0;124

6 54 94 - 0;125

7 55 38 - 0;126

8 56 42 - 0;127

9 57 40 - 0;128

0 58 41 - 0;129

Tecla !/§ 167 166 28 0;41

Tecla ?/' 39 63 31 0;130

Tecla ?/\* 173 168 - 0;131

Tecla ^/^/[ 96 94 27 0;26

Tecla \*/+[/] 43 42 29 0;27

Tecla // { 39 249 - 0;40

Tecla ‡/} 128 135 - 0;43

Tecla ;/, 44 59 - 0;51

Tecla :/. 46 58 OMPT ha de usar otro m, todo para generarlo. Quien genera el código Intro es el par metro \$ \_ dentro de la cadena valor.

Siguiendo el ejemplo anterior, ,ste quedaría de la siguiente forma:

**Código:**

PROMPT \$E[0;68;"|MORE\$ \_"p

- Puede redefinir alguna tecla con la orden externa "HELP" o con la cadena "/" seguida de un Intro. De esta forma solo tendrá que pulsar esta tecla antes o después de una orden del DOS (según utilice el HELP o el interrogante) para obtener la pantalla de ayuda de algún comando.

- Las teclas más fáciles de redefinir son las teclas de función, pero recuerde que hay algunas teclas que se utilizan para generar un carácter o que son usadas por DOSKEY. Por tanto, no reasigne las teclas F6, F7, F8 y F9 y ALT-F9, además de las teclas que tienen un significado como CTRL-C o CTRL-Z.

Vayamos ahora con la siguiente orden para activar/desactivar la reasignación de teclas repetidas en los teclados ampliados (101 teclas).

La orden 'q' de ANSI.SYS sirve para redefinir estas teclas repetidas. Se puede hacer de dos formas:

Esc[0q

Esc[1q

Empecemos por ver como trabaja el valor más alto. Si activamos Esc[1q nos permite reasignar independientemente las teclas repetidas del teclado ampliado. Si no activamos esta orden, entonces cuando redefinamos una tecla con otro valor (según el método, todo explicado con anterioridad, usando la orden 'p'), al cambiar el valor de una tecla también se cambiar la correspondiente en el teclado ampliado (si es que existe).

Si activamos Esc[1p, dos teclas iguales (en el teclado normal y teclado ampliado) pueden tener dos valores diferentes. Sin embargo, para el perfecto funcionamiento de esta orden conviene tener en cuenta lo siguiente:

- Si configura ANSI.SYS en el fichero CONFIG.SYS con el parámetro /K no funciona la orden 'q'. El parámetro /X tiene la misma función que la orden Esc[1p. Decida cuál de las dos formas es la que quiere utilizar. Por defecto, el controlador ANSI.SYS carga desactivada la orden de reasignación de teclas ampliadas.

- Hay que tener en cuenta que el controlador ANSI.SYS guarda un valor asignado, pese a que no tenga efecto si la redefinición del teclado ampliado no fue activada. Suponga un ejemplo:

Esc[224;71;"valor de Inicio en teclado ampliado"p

Esta orden no tendrá efecto como nuevo valor de la tecla Inicio del teclado ampliado si no está activada la reasignación del teclado ampliado, valga la redundancia. Esto

significa que cuando se active Esc[1q se activar el nuevo valor sin tener que repetir la secuencia.

La orden Esc[0q es muy sencilla de entender. Desactiva la reasignación de los valores dados para los teclados ampliados.

#### 4.5 La pantalla muestra sus atributos con otras órdenes

Como se ha visto, las ordenes PROMPT y ECHO permiten generar órdenes ANSI de formas diferentes. PROMPT utiliza el metacaracter \$E que indica el carácter de escape ASCII. ECHO utiliza el carácter ESC (ASCII 27) representado en pantalla por una flecha dirigida hacia la izquierda (<-). Las ordenes de ANSI.SYS se identifican mediante una secuencia de caracteres que se reconoce por sus dos primeros bytes. El primer byte sera el carácter de la secuencia de escape y el segundo, un simple corchete izquierdo ([), con valor ASCII 91.

Hasta este punto, queda claro como generar ordenes ANSI con lo aprendido, pero existe otro método con la orden TYPE

del DOS. La orden TYPE visualiza ficheros usando la salida estandar (el monitor CON). Por tanto, si se crea un fichero de texto con el código ANSI Esc[2J y luego se lee mediante la orden TYPE también se borra la pantalla.

El metodo para escribir ordenes TYPE es el descrito para la orden ECHO. Para introducir el carecter de escape puede

hacerlo con la combinación ALT-27 del teclado num,rico, o bien pulsando CTRL-P y luego la tecla ESC desde el editor EDIT del DOS.

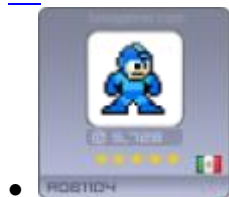
Ejecutada una orden ANSI ,sta se generar , si bien el indicador del PROMPT se habra desplazado una fila hacia abajo.

Para evitar esto (un retorno de carro, el resultado de pulsar una vez Intro) conviene eliminarlo introduciendo el código de fin de fichero (el carácter CTRL-Z). De esta forma, todo lo demas no se tiene en cuenta. Para introducir este código desde el editor EDIT del DOS, pulse CTRL-P y luego CTRL-Z.

El símbolo que aparecer en pantalla ser el de una flecha hacia la derecha (->).

[vitrox2007](#), [24/2/08](#)

[#1](#)



[rob1104](#) *Super Moderador Super Moderador*

Muy bueno, gracias. Lo paso a Manuales y Tutoriales.

Saludos

[rob1104](#), [24/2/08](#)  
[#2](#)



[vitrox2007](#) *:::Omnipresente::: Miembro*

Dale!

Aquí la segunda parte..

-----

El porque de la orden TYPE viene explicado a continuación, pues esta orden no visualiza en pantalla ningún carácter, dado que las ordenes ANSI se filtran de la entrada y salida estándar. Es un método muy práctico para averiguar si hay algún error en un fichero de texto con Ordenes ANSI.

Piense que si al leer el fichero mediante TYPE apareciese algún carácter en pantalla, significaría que este carácter no se ha interpretado como orden ANSI y que, por consiguiente, existe algún error en la programación del archivo. Es el método más semejante a un depurador de ordenes de un compilador.

Ahora, ello no quiere decir que las ordenes del fichero sean correctas si no se produce ninguna salida en la pantalla. Tampoco hay salida si no se han introducido todos los parámetros de la orden.

Un último consejo, antes de entrar con los ejemplos.

Reasigne alguna de las teclas de función de su teclado con el carácter Esc[. De este modo, siempre que llame a la tecla de función y añadiendo los parámetros adecuados de la orden, podrá ejecutar ordenes ANSI desde la línea de comandos.

Con todo lo estudiado, podemos aventurarnos a crear un fichero batch complejo. Este archivo por lotes que hará uso de las secuencias de escape utilizando para ello la orden PROMPT, convertirá el indicador de comandos del sistema en una visualización más agradable y con información diversa mostrada por pantalla.

@ECHO OFF

```
PROMPT $e[s $e[1;1H$e[41;1;33mDirectorio = $p$e[K$e[1;60HFecha =  
$e$d$e$u$e[40;37m$e[25;1HCOMANDO $e[K$e[37;40m
```



Escriba toda la longitud del listado en una sola línea lógica de su editor de textos.

El primer par metro especificado en el comando PROMPT (`$[s]`) graba la posición del cursor cuando se invoca un comando. El siguiente par metro (`$[1;1H]`) mueve el cursor a la fila 1, columna 1 de la pantalla, asegurando con esto que el directorio actual aparezca en la parte superior de la pantalla, sin importar donde estaba el cursor cuando se llamo a la orden. Las siguientes ordenes (`$[41;1;33m]`) definen la primera línea para una pantalla en color. "41" crea un fondo rojo; "1" activa la alta intensidad y "33" hace que el color de primer plano (letras) sea amarillo. Con todo, escribimos el texto "Directorio =" acompañado del par metro (`$p`) que nos informa del disco y directorio actuales. Pero existe un problema: la línea en la que queremos escribir tiene unos colores preestablecidos en la configuración por defecto del ordenador. Para arreglarlo y establecer los colores elegidos por nosotros para la primera línea completa, debemos borrar la línea actual desde la posición del cursor hasta el final (`$[K]`). Los parámetros siguientes (`$[1;60H]`) colocan la presentación de la fecha en la columna 60 de la primera fila.

El texto asociado, evidentemente, es "Fecha =". Para hacer aparecer la fecha debe estar el par metro "`$d`" presente.

La siguiente línea utiliza el par metro "`$[u]`" para devolver el cursor a la posición anterior a la ejecución del PROMPT, donde empezar a escribir el nuevo indicador de comandos. Los parámetros siguientes alteran los atributos de color para la segunda línea del indicador de comandos. "40" crea un fondo negro; y "37" hace que el color de primer plano (letras) sea blanco. Por norma, trabajamos en modo 25x80 (aunque si estamos trabajando en modo texto VGA o SVGA, utilizar el modo 50x80), con lo que debemos mover el cursor a la fila 25, columna 1. El texto asociado al PROMPT es "COMANDO".

`$[K]` borra de nuevo hasta el final de la línea, estableciendo el negro como color de fondo y el blanco como color de primer plano para esta línea.

Finalmente, "`$[37;40m]`" establece el blanco sobre fondo negro para cualquiera de las respuestas escritas en el indicador de comandos.

#### 4.6 Reduzca el trabajo pesado con otras herramientas

Para trabajar adecuadamente con las secuencias de escape y la programación gráfica de la pantalla, conviene hacerse con algunas herramientas. En el disquete que viene con las UTILIDADES+2 se incluyen varios programas.

KC-PAL y KCSETPAL permiten controlar las opciones de color por defecto para las tarjetas gráficas EGA y VGA desde el DOS. El primer paso consiste en pedir los servicios de KC-PAL.

Se obtendrá una visualización de los dieciséis colores en el estado de asignación por defecto. Usando las teclas del cursor se puede seleccionar cualquiera de ellos. De este modo, se puede cambiar la asignación por defecto con las teclas '+' y '-' o introduciendo el número del color mediante el teclado numérico. Cuando se han elegido las nuevas asignaciones se pulsa ESC y el programa se configura dejando las nuevas asignaciones en su lugar.

Esta herramienta es de utilidad para, por ejemplo, cuando un programa como WordPerfect llame al color 1 (azul) de la pantalla, aparezca la nueva opción asignada por nosotros. Es una forma de personalizar el ordenador, que bien se puede incluir dentro de un fichero por lotes como AUTOEXEC.BAT.

Veámoslo.

Cuando se sale de KC-PAL se da información sobre un conjunto de parámetros para utilizar con su programa compañero, KCSETPAL. Este programa es el que restaura los colores preferidos por el usuario desde un fichero batch o una macro.

Actúa, por ejemplo, eligiendo un color para el fondo de la pantalla del ordenador. Tras elegir un nuevo color para el fondo desde KC-PAL y si elegimos un color verde claro, se ver el mensaje "KCSETPAL 24". Este mensaje es el comando necesario para incluirlo dentro del fichero AUTOEXEC.BAT y que dejar un color de fondo verde claro en la pantalla.

Otro tema interesante es la posibilidad de disponer de menús en la pantalla y otras informaciones en recuadros. Para presentar estos menús basta con crear un archivo batch que realice un ECHO de un menú creado con varias líneas gráficas del juego de caracteres ASCII extendido, o bien un TYPE de un fichero gráfico ASCII o ANSI.

Para crear un archivo ANSI se debe utilizar un editor ANSI. El editor realiza un archivo de texto con todo tipo de efectos: parpadeo, negrita, colores e incluso animación. En el disquete de las UTILIDADES+2 se incluye un programa de shareware, TheDraw, de TheSoft Programming Services.

Por un módico precio, la versión registrada incluye macros, ayuda y ejemplos de aprendizaje. El entorno gráfico del programa UTILIDADES++ está realizado con el ejemplo del PROMPT que antes hemos estudiado, donde el fichero por lotes que incluye este PROMPT hace después un TYPE de un fichero de texto gráfico realizado con este editor ANSI. En el disquete de las UTILIDADES+2 se encuentra este ejemplo de entorno gráfico.

Las técnicas de menús gráficos descritas hasta ahora son muy útiles y funcionan tanto en monitores color como en monocromo. No obstante, existe un procedimiento para visualizar menús cinco veces más rápido que usando el comando TYPE o ECHO. La clave consiste en utilizar menús que se hayan grabado en formato binario. Se pueden crear desde el editor ANSI TheDraw o bien desde un programa como Menu Maker, de Andrew Fluegelman. Los menús se visualizan llamándolos desde la función BASIC: BLOAD.

Otra programa de la recopilación de utilidades, RESPUEST, realiza una pausa en el proceso del fichero de lotes, esperando la pulsación de una tecla. Cuando ésta se produce, se puede examinar la secuencia de escape de la tecla para producir una acción.

Estas y otras herramientas para la programación de los archivos por lotes se encuentran dentro del disquete de las UTILIDADES+2. Estudie los ficheros de documentación y

vea las posibilidades y el rendimiento que puede extraer a estos programas en sus ficheros batch. Adóptelos para sus necesidades.

#### 4.7 Música, maestro

Cuando se implantó el controlador ANSI.SYS en la versión 2.0 del DOS no se consideró la posibilidad de añadir música y sonido a esta potente herramienta. Con el paso del tiempo, los artistas de los gráficos ANSI buscaron nuevas opciones extendidas para este controlador de dispositivo del DOS. Los CD-ROM de utilidades de dominio público están plagados de nuevas versiones ANSI (hay que andarse con cuidado, pues algunas de estas versiones no siguen el estándar de las rutinas del controlador). Los resultados, cada vez mas espectaculares, han llevado a los artistas ANSI (en su mayoría programadores de rutinas de instalación de CD-ROM bajo DOS y sysops de BBS) a la concepción de una fabulosa idea: la posibilidad de escuchar sonido y música a través del altavoz del PC o la tarjeta de sonido, orada y que imita a un teclado. Todas sus funciones derivan a la generación de código musical que pueda ser grabado en un fichero. Tal fichero puede sumarse, mediante una orden adicional, a cualquier fichero ANSI estándar, para que al ejecutarse, produzca sonidos junto al resto de instrucciones. Merece la pena hacerse con este programa, aunque sólo sea para ver la pantalla de presentación ANSI y que muestra letras alejándose en el espacio.

Todo lo dicho está muy bien, pero hay un problema en la nueva emulación ANSI, o mejor dicho, un conflicto. El actual estándar usa la secuencia Esc[M para borrar una línea de la pantalla. Una solución a este inconveniente la han aportado algunos programas de comunicaciones que ya incorporan las secuencias ANSI musicales. Telemate, Procomm Plus para Windows y BananaCom soportan el nuevo controlador de dispositivo sustituyendo la secuencia Esc[M por Esc[N, al tiempo que proponen cambiar la secuencia de borrado de una línea por Esc[Y. Esto último dejaría las puertas abiertas a la implantación de la secuencia correcta, cuando aparezca el nuevo estándar. Piénsese que el mundo de las BBS es el que más uso hace de las pantallas gráficas ANSI y que, por eso, algunos programas de emulación de terminal han querido ser los primeros en incorporar las nuevas posibilidades musicales.

Baste decir que la norma propuesta aportaría una nueva dimensión, la del sonido, a las ventajas de las facilidades de los códigos ANSI. Además, las secuencias musicales reconocerían la presencia de tarjetas de sonido como la Sound Blaster y Adlib o el altavoz interno del PC, emitiendo los sonidos por uno u otro canal.

#### 4.8 Ejecutando reclusos ANSI

Este capítulo le resultará imprescindible a la hora de elaborar sus diseños de pantalla o en la reasignación y control de teclas. Por lo tanto, repasemos algo de importancia y que quizás no haya quedado suficientemente claro.

Hay varios métodos para activar las órdenes ANSI.SYS.

Desde el DOS se suele utilizar la orden TYPE seguida de un fichero de texto con órdenes ASCII, o en su defecto la orden PROMPT.

Una secuencia de caracteres ANSI será reconocida como parámetro de la orden

PROMPT, dado que cuando se termina de escribir el comando y se pulsa Intro, se est produciendo una entrada y es entonces cuando ANSI supervisa la gesti3n.

Recuerde que el s3mbolo \$[E detr3s de la orden PROMPT representa el c3digo Esc.

La orden TYPE del DOS visualiza ficheros utilizando la salida del monitor. Por tanto tambi3n es gestionado por ANSI.

Para leer c3digos ANSI, se tiene que crear un fichero con c3digos ASCII que contenga las secuencias del controlador de dispositivo. El 3nico problema est en c3mo generar el

car3cter Esc; tan f3cil como pulsar ALT y teclear 27 en el teclado num3rico. Sin embargo en el editor EDIT deber utilizar la combinaci3n de teclas CTRL+P y soltar para que aparezca el car3cter ^P; luego pulse la tecla Esc de su teclado. Normalmente, ver que el car3cter Esc se visualiza en pantalla mediante una flecha a la izquierda ' '. As3 que no le extra3e ver listados en libros, manuales o revistas utilizando indistintamente cualquiera de los caracteres de representaci3n para las secuencias de escape.

Como ha podido ver se pueden efectuar todo tipo de operaciones y utilidades gr3ficas muy potentes desde los archivos batch del DOS. Si desea mejorar las posibilidades, a3n no esta dicho todo. En el siguiente cap3tulo usaremos el comando DEBUG para escribir programas muy sencillos (no se trata de estudiar ensamblador, ni de infundir ese respeto que muchos padecen ante este comando), sino de proporcionar una utilidad para programadores que permita encontrar errores en los programas, para beneficio de nuestros ficheros por lotes.

## Cap3tulo 5

### USO DEL DEBUG PARA PREGUNTAS INTERACTIVAS

-----

5.1 El depurador, un tipo duro que trata a los archivos como series de bytes DEBUG es una orden externa del DOS que sirve como depurador binario para visualizar la memoria del sistema o el contenido de un fichero. Se gestiona con 3rdenes propias cuya explicaci3n esta fuera de las posibilidades de este libro, aunque s3 que resumir3 brevemente las mismas y sus diferencias para cada microprocesador, am3n de conocer lo mas b3sico para crear programas sencillos, compa3eros de nuestros ficheros por lotes.

Antes de conocer su uso en la programaci3n de los ficheros por lotes, convendr3a un estudio b3sico de las instrucciones b3sicas para el microprocesador, lo que se entiende por ensamblador. Para que lo entienda, la redacci3n de un programa con DEBUG requiere que se conozca c3mo act3an sus 18 comandos.

DEBUG se usa, normalmente, para volcar en pantalla contenidos de programas y realizar peque3os programas en ensamblador.

Conviene saber las 3rdenes m3s esenciales de DEBUG si se quieren aprovechar los recursos de los ficheros por lotes y todas las ventajas del DOS. Por ejemplo, para entender los ejemplos que vienen a continuaci3n y que no le suene a chino, conviene

saber que el comando 'A' significa "assembler" y que todos los programas escritos con DEBUG deben comenzar en la dirección 100H. De todas formas, si éste no es su caso (no es ninguna aberración no conocer los secretos de DEBUG; muchos programadores de los que se dicen profesionales jamás han trabajado directamente con él) no se desespere. Tome los ejemplos aquí descritos para su uso personal y extraiga una conclusión: el ensamblador le puede sacar de algunos apuros en aquellas aplicaciones excesivamente complejas. De todas formas, buscando encontrar multitud de programas y herramientas que le harán el mismo servicio que la rutina pensada en ensamblador.

Cuando se carga el programa DEBUG, aparece un indicador propio, el carácter '-' (guión). Si a continuación se introduce la orden ? se visualizará una pantalla de ayuda con todas las órdenes que acepta DEBUG. Para salir del programa y regresar al DOS ha de ejecutar la orden Q.

## 5.2 "Cómo desnudar a un programa?"

Para empezar, crearemos un programa, IMPRIMIR.COM que imprime el contenido de la pantalla. Supóngase, por ejemplo, que un programa de los ejecutados en un archivo batch presenta en pantalla unos datos de interés para el operador. Para asegurarse de que el usuario dispone de esta información, el archivo batch puede ejecutar el programa IMPRIMIR después de que el programa ejecutado complete su ejecución.

Ejecute DEBUG, indicando en la línea de comandos el nombre del archivo a crear:

```
C:\>DEBUG IMPRIMIR.COM
```

DEBUG responderá en su programa que el archivo no ha sido encontrado, lo que quiere decir que no existía previamente y que éste se creará por primera vez.

Ahas respuestas interactivas dentro de un fichero por lotes. Observe que en su archivo batch, primero debe llamar al programa SN.COM y luego realizar las comprobaciones correspondientes a los valores (tres líneas que deben situarse después de realizada la pregunta). Para que los valores resulten verdaderos se han de situar de mayor a menor, tal y como están en el listado. La comprobación resultante llevará la ejecución del programa a la etiqueta (por usted definida) SI o NO.

Para establecer el programa SN.COM

```
C:\>DEBUG SN.COM
```

Archivo no encontrado

-A 100

1CAE:0100 MOV AH,08 ; Rutina para obtener caracteres del ; DOS

1CAE:0102 INT 21 ; Llamada a la rutina

1CAE:0104 CMP AL, 53 ; "Se ha pulsado la tecla S?"

```

1CAE:0106 JZ 010E ; Letra v lida. Saltar al final del ; programa

1CAE:0108 CMP AL, 4E ; "Se ha pulsado la tecla N?

1CAE:010A JZ 010E ; Letra v lida. Saltar al final del ; programa

1CAE:010C JMP 0100 ; Bucle para insistir hasta obtener ; un carácter válido

1CAE:010E MOV AH, 4C ; Rutina para terminar un programa ; DOS

1CAE:0110 INT 21 ; Fin del programa

1CAE:0112

-R CX

CX 0000

:12

-W

Escribiendo 0012 bytes

-Q

C:\>

```

Este programa podría emplearse en el arranque de un ordenador, durante la ejecución del AUTOEXEC.BAT, a fin de liberar memoria o no ante la carga de determinados programas (tal y como hace la orden CHOICE de MS-DOS 6). Algunos juegos, sobre todo los más recientes, necesitan obligatoriamente 580 Kb de memoria convencional. Si nuestro arranque ejecuta varios programas residentes, utilidades, DOSKEY, un ratón, etc., puede que sea necesario preguntar si desea o no ejecutar estos programas a fin de liberar parte de la memoria convencional.

MS-DOS 6 y Novell DOS sí disponen de esta función, tan sencilla como introducir el carácter '?' delante de la línea de comandos que deba ejecutarse. El fichero por lotes se detendrá en este punto y preguntará al usuario si desea o no ejecutar esa línea de comandos.

Si bien los dos ejemplos de programación en ensamblador son muy elementales, no se ha pretendido en ningún momento dar una introducción al estudio del ensamblador. Simplemente, se ha querido ilustrar la capacidad de los archivos batch con otras herramientas, algunas de las cuales las podemos crear nosotros con programas en ensamblador. Si éste no es su caso, no se preocupe; tan sólo le interesa conocer estas posibilidades y buscar los programas adecuados para acogerlos en sus ficheros por lotes interactivos. Le garantizo que buscando -y no me cansar, de repetirlo- uno siempre encuentra lo que quiere.

### 5.3 Ensamblador para inmortales

Hay una parte del DOS que pertenece a los mortales y es aquella en la que hacemos uso de los comandos y órdenes del sistema operativo, ya sea para movernos por éste o para programar nuestros propios ficheros por lotes. Sin embargo, cuando se menciona DEBUG o el BASIC del DOS parece que se está hablando para los dioses del ciberespacio, esos geniecillos que realizan auténticas proezas con estas herramientas.

Aunque la realidad es bien distinta, he dejado para este apartado algunos trucos adicionales para quienes deseen entrar en el limbo de los dioses.

Para empezar, existe un truco que evita tener que teclear órdenes desde el propio DEBUG. Consiste en crear un listado de texto con las órdenes en ensamblador. Por ejemplo, utilice el editor del DOS para escribir lo que vea en cualquier listado de ensamblador (tal como está, incluyendo líneas en blanco) y grábelo como SCRIPT.TXT. Seguidamente haremos uso de una de las órdenes para redireccionar entradas con DEBUG <

SCRIPT.TXT; lo que generará el fichero ejecutable SCRIPT.

Sustituya SCRIPT por el nombre del programa.

Con lo sabido, experimente con estos ejemplos y que le serán de gran utilidad para sus ficheros batch.

El primer ejemplo está basado en crear un programa que sustituya al controlador de dispositivo ANSI, prescindiendo de utilizar cualquier sustituto ejecutable.

```
:HAZESC
```

```
ECHO OFF
```

```
ECHO Creando ESC.COM en el directorio activo
```

```
DEBUG ESC.COM <HAZESC.BAT > NUL
```

```
GOTO TEST
```

```
A100
```

```
MOV SI,80
```

```
MOV CH,0
```

```
MOV CL, [SI]
```

```
MOV BYTE PTR[SI+1],1B
```

```
MOD DX,81
```

MOV BX,1

MOV AH,40

INT 21

INT 20

RCX

17

W

Q

:TEST

IF EXIST ESC.COM GOTO OK. ECHO ESC.COM no ha sido creado

ECHO DEBUG.COM debe encontrarse en el directorio activo

ECHO o estar especificado en la ruta del PATH

GOTO FIN

:OK. ECHO ESC.COM ha sido creado

:FIN

Lo mejor de este programa es que no es un listado ensamblador de los habituales, sino que mezcla técnicas de programación batch con el ensamblador. Para ello deberá teclear y guardar el listado anterior en su editor preferido con el nombre de HAZESC.BAT

A continuación, desde el DOS, teclee

HAZESC (Intro)

lo que generará el ejecutable ESC.COM

Para utilizar ESC.COM como orden ANSI, sólo tiene que agregar la orden ESC antes de la secuencia de escape (ejemplo:

ESC [2J).

El siguiente ejemplo prueba la presencia de una unidad duplicada. Le servirá de ayuda para determinar si sus ficheros batch van a ser ejecutados sobre una unidad DoubleSpace o no, actuando en consecuencia.

Cree el listado TESDS.SCR desde su editor de textos.



E 100 B4 19 CD 21 88 C2 B8 11

E 108 4A BB 01 00 CD 2F 09 C0

E 110 75 09 F6 C3 80 74 05 B0

E 118 01 EB 02 B0 00 B4 4C CD

E 120 21

P gina 53

RCX

21

N TESTDS.COM

W

Q

Utilice el metodo DEBUG < TESDS.SCR para crear el fichero ejecutable.

El siguiente listado batch hace uso de TESDS.COM para supervisar la presencia de una unidad duplicada.

@ECHO OFF

REM DOBLESN.BAT

TESTDS

IF ERRORLEVEL 1 GOTO SI

ECHO La unidad ha sido comprimida con DoubleSpace

GOTO EXIT

:SI

ECHO La unidad ha sido comprimida con DoubleSpace

:FIN

El código de error 1 es el que determina si el disco duro ha sido duplicado o no. Cualquier otro código de error diferente indicar que la unidad no ha sido compactada.

Aparte de las funciones que pueden desempeñar, los más expertos deberían considerar

el estudio de las técnicas usadas en estos ejemplos, pues contienen rutinas avanzadas para aplicar en sus desarrollos.

## Capítulo 6

### CODIGOS DE CONTROL PARA IMPRESORA

-----

#### 6.1 El lenguaje de la impresora no es nada vulgar

La tabla de los códigos de control de impresión que viene a continuación, combina estos códigos con otros caracteres de control. Por ejemplo, para enviar un texto con cursiva a las impresoras Epson, debe teclear el código de escape <Alt-27> (recuerde, pulsando la tecla ALT junto al número correspondiente del teclado numérico) seguido del número 4.

Una excepción a esta norma es que algunas impresoras utilizan el código de escape FS (Alt-28).

[Código [Carácter ASCII] de control [Descripción] /(en decimal)]

14 Modo ampliado (sólo una línea)

20 Cancela el modo ampliado

27 W1 (87 1) Modo ampliado

27 W0 (87 0) Modo ampliado

15 Modo condensado

18 Cancela el modo condensado

27 4 (52) Cursiva o it lica (Epson)

27 5 (53) Cancela it lica (Epson)

27 -1 (45 1) Subrayado

27 -0 (45 0) Cancela subrayado

27 E (69) Negrita

27 F (70) Cancela negrita

27 G (71) Doble pasada

27 H (72) Cancela doble pasada

27 M (77) Letra Elite (Epson)

27 : (58) Letra Elite (IBM)

27 x1 (120 1) Letra NLQ (de calidad)

27 P (80) Letra Pica Normal (Epson)

27 S0 (83 0) Superíndice

27 S1 (83 1) Subíndice

27 T (84) Cancela superíndice y subíndice

27 @ (64) Resetea todos los códigos

27 g (103) Microimpresión (15 cpi)

27 p1 (112 1) Escritura proporcional (Epson)

27 p0 (112 0) Desactiva esc. proporcional (Epson)

27 P0 (80 1) Escritura proporcional (IBM)

27 P1 (80 1) Desactiva esc. proporcional (IBM)

27 w1 (119 1) Altura doble de caracteres (Epson)

27 w0 (119 0) Des. altura doble caracteres (Epson)

27 q1 (113 1) Outline o contorno (Epson)

27 q2 (113 2) Shadow o sombra (Epson)

27 q3 (113 3) Contorno y sombreada (Epson)

27 q0 (113 0) Desactiva contorno y/o sombra (Epson)

Recuerde que si el código de escape 27 no funciona en su impresora, deberá probar con el código de escape 28.

Como es evidente, se pueden combinar dos o más códigos para obtener el estilo de letra deseado. Por ejemplo, doble pasada con negrita o ampliado con doble pasada y elite.

Si todavía no lo ha visto muy claro, nada mejor que un programa que hace uso de todos los modos de impresión, a fin de que compruebe el funcionamiento de cada uno de los códigos de control de impresión.

El programa prepara la impresora para el envío de un fichero de texto. Según la opción que hayamos elegido, obtendremos un resultado u otro cuando enviemos el texto a imprimir. El programa, llamado SETTIPO.BAT espera a que le pasemos el par metro

correcto (uno o varios) para preparar la impresora.

@ECHO OFF

CLS

REM Programa SETTIPO.BAT

IF "%1"==" " GOTO AYUDA

ECHO Preparando estilos de impresi n

GOTO %1

:BUCLE

SHIFT

IF "%1"==" " GOTO FIN

GOTO %1

:AYUDA

ECHO Use cualquiera de estos nombres como par metro de SETTYPE

ECHO.

ECHO AMPL (modo ampliado activado)

ECHO COND (modo condensado activado)

ECHO SUBR (modo subrayado activado)

ECHO NEGR (modo negrita activado)

ECHO BOLD (modo doble pasada activado)

ECHO RESET (resetea todos los c digos de control)

GOTO FIN

:AMPL

ECHO <Alt-27>W1 > LPT1

GOTO BUCLE

:COND

ECHO <Alt-15> > LPT1

GOTO BUCLE

:SUBR

ECHO <Alt-27>-1 > LPT1

GOTO BUCLE

:NEGR

ECHO <Alt-27>E > LPT1

GOTO BUCLE

:BOLD

ECHO <Alt-27>G > LPT1

GOTO BUCLE

:RESET

ECHO <Alt-27>@ > LPT1

:FIN

Quizás ahora comprenda cómo es posible que las impresoras lleven consigo estilos de impresión, que no dejan de ser los mismos que los descritos anteriormente. Un simple fichero batch da el mismo resultado y es igualmente efectivo.

Una impresora láser puede hacer mucho más que su equivalente de matriz de puntos, con la excepción de que incorpora más comandos para controlar ciertas características, como los gráficos de trescientos puntos por pulgada (300 ppp), tipos de letra escalables, estilos y tipos de letra recargables. El lenguaje más habitual (después de PostScript) que controla estas características es el PCL, al que antes se ha hecho referencia. A diferencia de la impresora de matriz de puntos, una impresora láser no imprime hasta que ha recibido una página completa de texto; es obvio que los comandos de la impresora HP deben estar contenidos en un archivo.

A continuación y como en el caso de las impresoras matriciales, adjunto una lista con algunos de los comandos principales del lenguaje de descripción de páginas PCL.

Consulte siempre la documentación de su impresora para ver el conjunto completo de las secuencias de escape especificadas para su utilización (tanto para impresoras matriciales como láser o inyección de tinta).

[Código [Carácter ASCII] de control] [Descripción]

27 (10U) Conjunto de símbolos PC-8

27 (8U) Conjunto de símbolos Roman-8

27 (s1P Espaciado (proporcional)

27 (s0P Espaciado (fijo)

27 (sanchoH Ancho (en caracteres por pulgada)

27 (spuntosV Altura

27 (s1S Cursiva

27 (s4S Condensada

27 (s8S Comprimida

27 (s24S Ampliada

27 (s32S Fuera de línea

27 (s64S En línea

27 (s128S Sombreada

27 (s0S Desactiva todos los estilos a normal

27 (s-3B Impresión de borrador

27 (s0B Impresión media

27 (s3B Negrita

27 (s0T Letra normal

27 (s1T Letra Pica

27 (s2T Letra Elite

27 (s3T Letra Courier

27 (s4T Letra Helv,tica

27 (s5T Letra Times Roman

27 &d0D Subrayado

27 &d@ Cancela subrayado

7 Suena pitido impresora

27 E Resetea impresora y expulsa una

Existen otro tipo de c digos de control de impresi n que son los que hacen referencia a los comandos de formato, es decir, aquellos que controlan caracter sticas como el espacio interlineal, los saltos de l nea o p gina, el tama o de p gina, los m rgenes y los tabuladores; los comandos de gr ficos, que controlan la impresi n de 911

17 968

18 1025

19 1082

20 1139

21 1196

22 1253

23 1310

24 1367

25 1424

26 1481

27 1538

28 1595

29 1652

30 1709

31 1766

32 1823

33 1880

34 1937

35 1994

No creo que sus documentos vayan m s all de las dos mil l neas, pero si es as , s lo tiene que ampliar la plantilla aplicando el modificador de +57 a cada nueva p gina.

No olvide a adir en la primera l nea un comando de impresi n de salto de p gina. Se

hace servir para una primera prueba de avance de página; si ve que funciona correctamente, el resto de la documentación se imprimirá en su totalidad.

Con toda probabilidad, lo que más utilizará en la programación de sus ficheros por lotes será la pequeña tabla de control de la impresora, aparte de la plantilla para ver en qué lugar de la hoja se produce el avance de página. Ahora bien, conviene conocer el resto de comandos de impresión para aquellas ocasiones en que tenga que programar aplicaciones más complejas.

## Capítulo 7

### MÁS ALLA DEL DOS 5

-----

#### 7.1 La batalla de los titanes del DOS

A finales de 1980, IBM contactó con Bill Gates para revelarle que estaban considerando la producción de un ordenador de ocho bits. IBM quería que Microsoft diseñase una versión del Basic para esta nueva máquina, incluida en la memoria ROM. Gates recomendó a IBM la posibilidad de diseñar un ordenador de 16 bits. Sugirió, entonces, que IBM se pusiese en contacto con Gary Kildall, de Digital Research, para diseñar un sistema operativo de 16 bits. No obstante, Digital se sobrepasó con IBM y se asoció con la competencia para diseñar aplicaciones para el sistema operativo CP/M.

En cualquier caso, Gates e IBM compraron los derechos del sistema operativo 86-DOS de Patterson. El 12 de agosto de 1981 se presentó el primer PC y el sistema operativo DOS 1.0 (al cual IBM llamó PC-DOS). En marzo de 1983, IBM anunció su PC-XT y la versión 2.0 de DOS. Con la llegada del AT llegaron las versiones 3.x. El resto es historia. Una versión 4.0x que dio muchísimos quebraderos de cabeza. Una sorprendente versión 5, más completa y agradable. Una versión 6.x que añadió programas adicionales y un duplicador de discos...

Toda esta historia se complica cuando Digital Research decide diseñar su propio sistema operativo compatible con DOS.

DR-DOS 5 y DR-DOS 6 calaron hondo en el mercado europeo por sus características: un interfaz gráfico agradable, utilidades para transferir ficheros, soporte de redes, comandos avanzados... Por su parte, Microsoft decide divorciarse de IBM para casarse con Intel. IBM continúa su carrera y continúa diseñando su propio sistema operativo PC-DOS. A su vez, Novell compra en 1993 a Digital y absorbe su sistema operativo.

Consecuencia de todo ello (a la hora de escribir este libro electrónico) es que mientras que Microsoft ha llegado hasta la versión MS-DOS 6.21 (de la que se ha extraído el duplicador de discos, por problemas jurídicos con Stac Electronics), Novell acaba de presentar Novell DOS 7, mientras que IBM ha presentado su PC-DOS 6.3 (con un salto extraño de dígitos para evitar confusiones a los usuarios). De todas estas versiones hablaremos ahora y de las herramientas incluidas para la programación de archivos batch.



## 7.2 "MS-DOS 6? \*No es para tanto!

Durante un tiempo, todos los medios de comunicación especializados en informática, coincidieron en afirmar que las nuevas prestaciones del más reciente sistema operativo de Microsoft no superan las de ciertos programas de dominio público, y mucho menos las expectativas creadas a su alrededor.

A la vista de las innovaciones incluidas (la mayor parte aplicaciones de otras empresas de reconocido prestigio), los críticos creyeron injustificado hablar de la subida de un punto en la numeración de la nueva versión.

Pese a estas consideraciones, MS-DOS 6 y versiones 6.x posteriores, incluyeron las siguientes características:

- DoubleSpace, un programa integrado para comprensión de discos (duros o flexibles) que aumenta el espacio disponible mediante la compactación de archivos.
- MemMaker, un programa de optimización de memoria.
- Comandos mejorados, como es el caso del modificador /C para el comando DIR que presenta información de las tasas de comprensión; el comando MEM que ahora proporciona más detalles acerca de la memoria; LOADHIGH y DEVICEHIGH que incluso permiten especificar el sector de memoria donde se desee cargar un programa.
- Un nuevo programa de copias de seguridad (Backup) para Windows.
- Microsoft Anti-virus, programa que identifica y elimina unos mil virus electrónicos.
- El programa Restablece (Undelete) mejorado, con tres niveles de protección y versión para Windows.
- MS-DOS Defragmentar, un nuevo programa que reorganiza los archivos del disco duro.
- El programa SMARTDrive mejorado y que ahora almacena información que vaya a ser escrita en el disco duro.
- Microsoft Diagnostics (MSD) que presenta información técnica del sistema.
- Interlink, programa para transferir ficheros entre PC.
- Comando POWER que desactiva la energía de la batería de los ordenadores portátiles.
- Nuevos comandos: MOVE, para mover ficheros de un directorio a otro; y DELTREE, que elimina un directorio y sus subdirectorios y archivos subordinados.
- Un controlador para el CD-ROM (MSCDEX.EXE).
- SCANDISK, un chequeador del disco para corregir posibles errores, incluyendo los

derivados de la duplicación con DoubleSpace.

- Mejoras de órdenes y controladores tales como CHKDSK.COM, DOSKEY.COM, DEBUG.COM, ANSI.SYS, EDIT.COM, DOSSHELL.EXE, FDISK.COM, FORMAT.COM, QBASIC.ESE, UNDELETE.EXE, HIMEN.SYS, XCOPY.EXE y UNFORMAT.COM.

La verdad es que las novedades son bastantes.

Personalmente, y pese a los comentarios adversos de la crítica especializada, he encontrado bastantes características que convencer de por sí solas para cambiarse de versión en cualquier ordenador.

En cuanto a lo que nos interesa, las facilidades para los archivos batch, ésta es la síntesis de las nuevas mejoras:

- Capacidad para definir más de una configuración en el sistema en un archivo CONFIG.SYS, a través de menús agradables y vistosos.
- Resolución de problemas en los programas de arranque del ordenador. Se pueden desviar los comandos de inicio cuando se enciende el PC con algunas teclas de función.
- El nuevo comando CHOICE que espera información del usuario y devuelve un ERRORLEVEL.
- El nuevo comando NUMLOCK para, siempre y cuando después del nombre de elemento, coloque una ',' (coma). Aunque es posible indicar un nombre de bloque largo, es recomendable que éste sea corto, puesto que el mismo se almacena en una variable de entorno, ocupando espacio. Cada bloque de menú puede contener hasta un máximo de 9 opciones MENUITEM. Si se desea crear más configuraciones, deber utilizarse la orden SUBMENU para crear nuevos bloques de menú, cada uno de los cuales podrá tener sus 9 opciones. Si se pulsa F5 cuando aparezca en pantalla el menú no se procesará ninguno de los bloques de menú, mientras que si se pulsa F8 se pedirá confirmación para cada una de las órdenes a ejecutar en el menú elegido.

Sintaxis: MENUITEM = bloqueOrdenes,título

MENUDEFAULT Especifica el elemento de menú predeterminado. Especifica el bloque de menús que se arrancará por defecto si el usuario no hace ninguna elección. Este comando es opcional; si no se especifica, la configuración predeterminada se establecerá en el elemento 1. Este comando, seguido de una coma ',' permite añadir el valor de tiempo opcional en el que el ordenador esperará tantos segundos como haya indicado, antes de arrancar la configuración por defecto.

Sintaxis: MENUDEFAULT = bloqueOrdenes,segundos

MENUCOLOR Establece los colores para el texto de arranque y el fondo del menú. Los valores numéricos pasados como parámetros de estas ordenes son los siguientes:

[Número] [Color]

0 Negro

1 Azul

2 Verde

3 Cíen

4 Rojo

5 Magenta

6 Marrón

7 Blanco

8 Gris

9 Azul brillante

10 Verde brillante

11 Cíen brillante

12 Rojo brillante

13 Magenta brillante

14 Amarillo

15 Blanco brillante

Sintaxis: `MENUCOLOR = colorTexto,colorFondo`

No se puede dejar ningún espacio en blanco entre la coma que separa los dos parámetros.

**SUBMENU** Especifica un elemento de menú que presenta otro juego de opciones. El comando especifica una desviación a otro bloque del menú con nuevas elecciones. Como puede comprobar, MS-DOS abre ramificaciones de menús en forma de árbol.

Sintaxis: `SUBMENU = bloqueMenú, título`

Un bloque de configuraciones puede contener cualquier comando que normalmente situaríen en su archivo `CONFIG.SYS` con cualquier versión del DOS. Pero además, MS-DOS 6 añade, aparte de los comandos ya descritos, otros dos más, particularmente útiles en los bloques de configuraciones:

**INCLUDE** Este comando indica al DOS que ejecute los comandos de otro bloque de

configuraciones, dentro del bloque de configuraciones en el que se encuentre.

Sintaxis: INCLUDE bloqueOrdenes

[COMMON] Nombre del bloque de configuraciones que ser común a todas las configuraciones de arranque. Microsoft recomienda añadir este bloque al final del archivo CONFIG.SYS, puesto que algunas aplicaciones y programas modifican este archivo agregando nuevas órdenes al final, y que debieran ser comunes a todas las configuraciones. Si no se especificase este nombre de bloque, las órdenes nuevas se situarían en el último bloque, activándose únicamente para éste.

Veamos qué significado tiene todo lo explicado hasta ahora, con otro ejemplo:

[MENU]

menuitem=Windows, Configuración para Windows

menuitem=DOS, Configuración para DOS

menuitem=Juegos, Configuración para juegos con memoria

expandida

menucolor=15,1

menudefault=Windows, 20

[WINDOWS]

set path=c:\windows;c:\dos

set temp=c:\windows\temp

[DOS]

path=c:\;c:\dos

[JUEGOS]

include=windows

device=c:\dos\emm386.exe 2048

[COMMON]

device=c:\dos\ansi.sys

Este archivo CONFIG.SYS define un bloque [COMMON] que incluye comandos comunes a todas las configuraciones; en este caso, el controlador de dispositivo ANSI.SYS.

También incluye tres configuraciones: Windows, DOS y Juegos. El bloque de configuraciones [JUEGOS] incluir los comandos del bloque de configuraciones de [WINDOWS], dado que algunos juegos, como es lógico, se ejecutarán bajo Windows.

El comando MENUCOLOR establece el color del texto en 15 (blanco brillante) y el color del fondo en 1 (azul).

El comando MENUDEFAULT especifica a [WINDOWS] como la configuración y el elemento de menú predeterminados, dando un valor de intervalo de 20 segundos antes de que se inicie el menú por defecto.

Veamos otro ejemplo que activa una opción de submenú.

[MENU]

menuitem=dospañol, Ejecutar DOS 6 en castellano

submenu=otros, Ejecutar DOS 6 en otros idiomas

menudefault=dospañol, 5

[OTROS]

menuitem=dosingl,s, Ejecutar DOS 6 en inglés

menuitem=dosfranc,s, Ejecutar DOS 6 en francés

menuitem=dosalem n, Ejecutar DOS 6 en alemán

menudefault=dosinglés

[DOSESPAÑOL]

country=34,,c:\dos\country.sys

[DOSINGLES]

country=1,,c:\dos\country.sys

[DOSFRANCES]

country=33,,c:\dos\country.sys

[DOSALEMAN]

country=49,,c:\dos\country.sys

[COMMON]

files=30

buffers=20

shell=c:\dos\command.com /e:512 /f /p c:\dos

Este fichero CONFIG.SYS mostrar en pantalla un menú principal con dos opciones, donde la primera se asume por defecto y se activa a los 5 segundos si el usuario no indica lo contrario.

Si se escoge la segunda opción del menú principal aparece el submenú [OTROS], donde se asume por defecto la opción de ejecutar el teclado en inglés y sin límite de tiempo para que el usuario tome una decisión.

“Se va viendo para qué pueden servirnos las múltiples configuraciones de arranque?

Conviene señalar que cuando se usan múltiples configuraciones, puede hacerse que el DOS ejecute comandos diferentes de AUTOEXEC.BAT diferentes para cada configuración.

Se pueden crear distintas configuraciones en el archivo AUTOEXEC.BAT usando comandos como IF y GOTO.

Ver , cuando se selecciona una configuración en el menú de inicio de CONFIG.SYS, MS-DOS 6 define una variable de entorno llamada CONFIG. Podr instruir al DOS para que compruebe la variable y realice una serie de órdenes en virtud de lo que haya en la variable.

Por ejemplo, si inserta el siguiente comando en el archivo AUTOEXEC.BAT

GOTO %CONFIG%

podrá indicar bloques de configuraciones que deberán coincidir con los mismos nombres de los bloques de configuraciones de CONFIG.SYS.

:CESAR

Este nombre de etiqueta correspo

2 prompt \$p\$g

3 path c:\dos;c:\;c:\4dos;c:\windows;d:\wpwin;c:\norton;d:\dbase

4 set blaster=a220 i2 d1 t2

5 set temp=c:\dos

6 set sound=c:\sbpro

```
7 goto %config%

8 :Normal

9 set comspec=c:\4dos\4dos.com

10 c:\qemm\loadhi /rf nlsfunc c:\dos\country.sys

11 mode con cp prepare=((437,850) c:\dos\ega.cpi)

12 c:\qemm\loadhi /rf c:\dos\keyb sp,,c:\dos\keyboard.sys

13 chcp 437

14 c:\qemm\loadhi /rf c:\dos\mouse

15 c:\qemm\loadhi /rf c:\dos\smartdrv 512 512

16 goto end

17 :Red

18 set comspec=c:\4dos\4dos.com

19 c:\qemm\loadhi /rf nlsfunc c:\dos\country.sys

20 mode con cp prepare=((437,850) c:\dos\ega.cpi)

21 c:\qemm\loadhi /rf keyb sp,,c:\dos\keyboard.sys

22 chcp 437

23 c:\qemm\loadhi /rf c:\dos\mouse

24 call c:\nwlite\startnet

25 goto end

26 :end
```

## CONFIG.SYS

```
1 [common]

2 device=c:\qemm\dosdata.sys

3 shell=c:\c:\4dos\4dos.com /p /e:300

4 set loadhidata=c:\qemm\loadhi.rf
```

5 [Menu]

6 menuitem=Normal, Memoria extendida adaptada a Windows

7 menuitem=Red, Memoria EMS y XMS; adaptada para red

8 menucolor=14

9 menudefault=Normal, 20

10 [Normal]

11 device=c:\qemm\qemm386.sys frame=none ems:n maps=0 norom

x=b000-b7ff rf ram

12 device=c:\qemm\dos-up.sys @c:\qemm\dos-up.dat

13 dos=high

14 device=c:\qemm\loadhi.sys /rf c:\dos\eansi.sys

15 device=c:\qemm\loadhi.sys /rf c:\dos\display.sys con=(ega,437,2)

16 device=c:\qemm\loadhi.sys /rf c:\dos\dblspace.sys /move

17 break=on

18 files=30

19 buffers=30

20 country=034,,c:\dos\country.sys

21 stacks=0,0

22 lastdrive=f

23 [Red]

24 device=c:\qemm\qemm386.sys ram x=b000-b7ff st:m rf

25 device=c:\qemm\dos-up.sys @c:\qemm\dos-up.dat

26 dos=high

27 device=c:\qemm\loadhi.sys /rf c:\qemm\st-dbl.sys

28 device=c:\qemm\loadhi.sys /rf c:\dos\eansi.sys



29 device=c:\qemm\loadhi.sys /rf c:\dos\display.sys con=(ega,437,2)

30 break=on

31 files=40

32 buffers=40

33 country=034,,c:\dos\country.sys

34 stacks=0,0

35 lastdrive=m

La numeración de las líneas no debe teclearse. Se han puesto a efecto orientativo.

Este ejemplo de dos configuraciones define un menú, con los elementos Normal y Red, en color amarillo sobre fondo negro. El elemento Normal ser el predeterminado, si el usuario no hace su elección antes de 20 segundos.

El menú arranca un bloque de configuraciones con memoria extendida adaptada para Windows. La primera línea de este bloque del archivo CONFIG.SYS excluye parte de la memoria de vídeo y la memoria expandida. No se fije mucho en estos parámetros y otros similares; los establece el gestor de memoria QEMM en su configuración. Aparte, observar que se ejecuta un controlador de dispositivo ANSI extendido, se preparan dos páginas de código, se activa el duplicador de disco de MS-DOS 6 y se ahorra memoria convencional dejando la pila de los programas en cero.

El segundo menú gestiona tanto la memoria expandida como la extendida, y arranca los controladores y drivers de una red local. Las órdenes son similares, a excepción de que el gestor de memoria añade su propio controlador de dispositivo para el duplicador de discos, ahorrando otros 40K más de memoria.

En el archivo AUTOEXEC.BAT, después de haber definido las trayectorias y variables de entorno (para tarjeta de sonido), introduce en la sección de Normal un caché, de disco duro para acelerar las aplicaciones bajo Windows, mientras que en la sección de Red se llama al fichero batch STARTNET (que contiene los drivers de dispositivos y la configuración de la red Netware de Novell). Por cierto que el gestor de memoria QEMM también mueve los ficheros por lotes asociados con AUTOEXEC.BAT a los bloques de memoria superior.

Algunas de las técnicas realizadas para este ejemplo pueden ser de interés. \*Estúdielas! A saber, el SHELL que instala en memoria el procesador de órdenes que controla el sistema no es el COMMAND.COM, sino un procesador compatible con el DOS muchísimo más potente. Por otro lado, esta misma línea define un entorno para variables de 300 bytes, ahorrando más memoria. Por último, el sistema se inicializa con dos páginas de código preparadas (437, 850), correspondientes a Estados Unidos y Multilingüe. Si sabe lo que esto significa, podrá activar una u otra con sólo teclear:

No es tan difícil. Tómese su tiempo para estudiar el ejemplo anterior, dado que no es ninguna tontería. Este caso activa toda una serie de dispositivos y configuraciones en un simple menú de arranque, realizado con MS-DOS 6 y QEMM 7, para un ordenador 386 con tan solo 4 MB de memoria adicional. Puedo asegurarle que esta configuración funciona perfectamente y que arranca las aplicaciones de Windows o red con una rapidez asombrosa.

Si no dispone de MS-DOS 6 no se desanime. Puede realizar lo mismo con líneas ECHO para visualizar la pantalla deseada, y haciendo uso de IF y GOTO, siempre y cuando compruebe el ERRORLEVEL devuelto. Además, el disquete de utilidades contiene algunos comandos que le pueden resolver la papeleta.

### 7.3 Novell DOS 7, la última frontera

Novell DOS 7 es el más duro competidor para cualquiera de las versiones MS-DOS 6.x. Novell DOS 7 simplemente es impresionante. Ofrece multitarea preemptiva (traducida como prioritaria), a fin de que múltiples aplicaciones DOS puedan trabajar simultáneamente, siempre y cuando el ordenador sea 386 o superior y disponga de 2 MB de RAM como mínimo.

Desafortunadamente, sólo funciona con su propio gestor de memoria (EMM386), aislando a los usuarios de cualesquiera de las versiones QEMM de Quarterdeck.

DOS 7, siendo un producto de Novell, está pensado para su uso en redes. \*Seamos justos! Cuando un cliente pide a su distribuidor que le instale una red, sólo quiere Novell Netware. Por muchas cifras que muestre Microsoft, la implantación de su red LAN Manager nunca podrá competir contra el monopolio de mercado de Novell. Su único hueco de mercado podría estar en el sector de las redes entre pares o igualitarias, lo que convierte a cada PC en servidor y cliente al mismo tiempo. Microsoft consiguió enlazar aplicaciones Windows a través de la red con sus productos estrella, Windows para Grupos de Trabajo 3.11 y Windows NT (el futuro Windows 95 también incorpora protocolos para redes igualitarias). Y parece que se habrá hecho con el mismo, \*hasta que apareció Novell DOS 7!

Personal Netware es una red entre pares (incluida en DOS 7) muestra el número de versión del DOS del sistema operativo, merced a la variable OS de Novell DOS 7.

```
if not %os%=nwdos goto otrosdos
```

```
echo Sistema trabajando bajo %os% %ver%
```

```
exit
```

```
otrosdos
```

```
echo Sistema corriendo bajo MS-DOS o PC-DOS
```

Advierta que la orden EXIT finaliza el proceso por lotes en Novell DOS 7, no así en

MS-DOS o PC-DOS.

Aparte de las variables ordinarias, DOS 7 provee de un nuevo tipo de variables llamadas variables de sistema. Estas variables proporcionan información adicional, según la siguiente lista:

[Variable] [Valor]

DAY Día comprendido entre 1 y 31

DAY\_OF\_WEEK Día de la semana

NDAY\_OF\_WEEK Día de la semana numerado entre 1 y 7 (el 1 corresponde al domingo, primer día de la semana para los americanos)

MONTH Mes numerado entre 1 y 12

MONTH\_NAME Nombre del mes

YEAR Año

SHORT\_YEAR Últimos dos dígitos del año (ej: 93)

HOURL Hora entre 1 y 12

HOURL24 Hora entre 0 a 24

MINUTE Minutos entre 0 y 59

SECOND Segundos entre 0 y 59

AM\_PM AM (antes del mediodía) o PM (después del mediodía)

GREETING\_TIME Espacio de tiempo: mañana, tarde o noche

LOGIN\_NAME Nombre de login de usuario (para red)

P\_STATION Número de estación física (para red)

STATION Número de estación (para red)

Estas variables nos pueden servir, por ejemplo, para añadir la siguiente línea al fichero AUTOEXEC.BAT:

```
if %nday_of_week%==2 echo Acuérdate de que hoy toca hacer una copia de seguridad
```

DR-DOS 6 ya incorporaba órdenes propias para su uso con ficheros batch. DOS 7, aparte de conservar las de versiones anteriores, añade nuevos comandos.

CHOICE realiza una pregunta y responde con un código de error, tal cual como su homónimo en MS-DOS 6.x (de hecho es totalmente compatible).

SWITCH establece una pregunta con una respuesta alfabética, desviando el proceso a una etiqueta.

GOSUB/RETURN implementa subrutinas de la misma forma que el BASIC.

? Requiere confirmación para ejecutar un comando precedido de este carácter.

CPOS coloca en una posición la localización del cursor en la pantalla, siempre y cuando se tenga activado ANSI.SYS.

TIMEOUT espera un tiempo determinado de segundos antes de ejecutar órdenes como ? o SWITCH. Si el tiempo expira sin una respuesta por parte del usuario se ignora el comando precedido por ? o se asume la primera opción de menú SWITCH.

CHAIN comprueba la existencia de un fichero adicional de configuración (sólo para CONFIG.SYS). Si existe transfiere el control a ese segundo archivo.

#### 7.4 PC-DOS 6.3, recogiendo las migas

En la batalla de los titanes del DOS, después del divorcio con Microsoft, IBM perdió el liderazgo en favor del imperio de Bill Gates. Ha sido entonces cuando IBM se ha lanzado a una guerra sin cuartel en la que no ha dudado en unirse hasta con su peor enemigo (Motorola, los creadores del microprocesador de los Macintosh) para arrebatarse el puesto de líder a Microsoft. Su versión del sistema operativo DOS tampoco es una versión más, sino que se ha dado cuenta de que no todo es OS/2 en esta vida y de que al DOS todavía le quedan muchas cosas que decir. PC-DOS 6.3 es el único de los sistemas operativos DOS que contempla las nuevas extensiones y dispositivos que al PC se le han ido acoplando y sumando.

Por lo pronto es el único DOS que integra las extensiones pen-computer para poder utilizar este dispositivo con forma de bolígrafo en lugar del ratón. Recordemos que los pen-computer son aquellos ordenadores portátiles, similares a un notebook, pero sin teclado alguno. Las extensiones PenDOS de IBM permiten navegar por la versión de Windows (y sus aplicaciones), adaptada para este dispositivo.

Asimismo, el PC-DOS 6.3 incluye software para las tarjetas PCMCIA III, un nuevo tipo de tecnología que se viene adaptando últimamente a los notebook para sustituir a los disquetes y discos duros actuales.

Aparte de las extensiones, pocas son las diferencias principales respecto al MS-DOS, a excepción de las utilidades incorporadas. IBM ha basado su optimización de la memoria en el gestor de Central Point, RAMBoost, que supera con creces a MemMaker de Microsoft, al menos en la creación de memoria superior. Además, el gestor detecta cuando se han realizado modificaciones en los ficheros AUTOEXEC.BAT y CONFIG.SYS.

En esta misma línea, cómo no, el programa que realiza las copias de seguridad no es otro que el CPBACKUP, licenciado a Central Point.

IBM ha incluido el sistema de compresión de datos de Addstor Inc. La utilidad SuperStore/DS es un duplicador de discos compatible con DoubleSpace, que incluso puede poner contraseñas a la compresión. SuperStore cuenta con la tecnología UDE (Universal Data Exchange), que no es otra cosa que un programa residente para leer diskettes comprimidos en ordenadores que no estén equipados con SuperStore. No obstante, este duplicador presenta muchas incompatibilidades con algunas aplicaciones y la recuperación de ficheros borrados.

Además de todos estos avances, PC-DOS 6.3 dispone de un fabuloso antivirus, capaz de detectar más de 2.000 virus y mutaciones, ocupando sólo de 1 a 4 KB de RAM. Otras novedades, recogen mejoras en varios mandatos, soporte de CD-ROM y defragmentador avanzado.

Aparte de estas utilidades, PC-DOS 6.3 guarda una total compatibilidad con las órdenes y comandos batch de MS-DOS 6.2; no así con las órdenes adicionales de Novell DOS 7. Es importante tener esto en cuenta a la hora de programar nuestros ficheros por lotes en ordenadores en los que desconozcamos la versión de sistema operativo instalada.

#### 7.5 4DOS, todo lo que siempre quiso el DOS y gratis

Existen algunos procesadores de órdenes, compatibles con el DOS, bastante más potentes y eficaces que el tradicional COMMAND.COM. Recuerde que esta orden carga en memoria las rutinas de las órdenes internas del DOS (COPY, CLS, DATE, VER, etc.). El fichero COMMAND.COM se carga automáticamente en memoria gracias al valor por defecto de la orden SHELL del fichero CONFIG.SYS.

Atención, observe, que he mencionado que existen unas órdenes internas, lo que significa que también hay unas órdenes externas, almacenadas cada una de ellas en un fichero independiente. Por ejemplo, la orden FORMAT es externa porque existe un fichero en alguna parte (normalmente en C:\DOS) llamado FORMAT.EXE. Pues bien, ¿qué nos impide sustituir el

COMMAND.COM por un procesador de órdenes a DOSKEY, de forma que el usuario puede recuperar los últimos comandos.

Este editor de comandos es mucho más versátil y flexible.

- Las variables de entorno para el usuario pueden definirse de 512 a 32.000 bytes. Se pueden editar y modificar estas variables con un comando.
- Incluye una orden para renombrar los comandos del DOS y para realizar macros (DIR como D, etc.). Estos sinónimos pueden tener archivos por lotes como a