Implementación de Manejadores de Dispositivo (Device Drivers)

Ing. José Ortega Bernal
Catedrático de la U. V. M.
Alumno de la Maestría del CIC-IPN
M. en C. Eduardo Vega Alvarado
Profesor del CIDETEC-IPN

Indudablemente, MS-DOS® (MicroSoft Disk Operating System) ha sido el sistema operativo de mayor éxito en la historia, estando presente en una amplia base de equipos de tipo personal. Si bien se le considera obsoleto después de la aparición de nuevos sistemas tales como Windows, DOS aún representa una herramienta flexible y sumamente poderosa como monitor para equipos dedicados, principalmente para labores de control. Así, mediante el desarrollo de la interfaz (circuitería) para control y un manejador de dispositivos adecuado, una computadora PC trabajando en DOS puede desempeñar funciones que difícilmente se programarían en Windows, dado el paradigma de programación de espera de eventos de este último.

INTRODUCCIÓN

Los manejadores o controladores de dispositivos (*device drivers*) constituyen uno de los aspectos más fascinantes (a la vez que complejos) de la programación de sistemas. Estos pequeños programas controlan el acceso y la comunicación con los dispositivos de entrada/salida externos, representando el nivel más bajo del sistema operativo; dadas sus características, estas rutinas normalmente deben programarse en lenguaje ensamblador.

En el caso especial del DOS, sus manejadores tienen la característica de incorporarse al kernel al momento de inicio de sesión, lo que proporciona una gran flexibilidad. Así, a diferencia de sistemas operativos como UNIX, donde la carga de nuevos controladores o la modificación de los ya existentes implica una recompilación del núcleo, en MS-DOS simplemente se requiere cambiar algunas líneas en el archivo de configuración CONFIG.SYS y reiniciar el equipo.

ESTRUCTURA BÁSICA

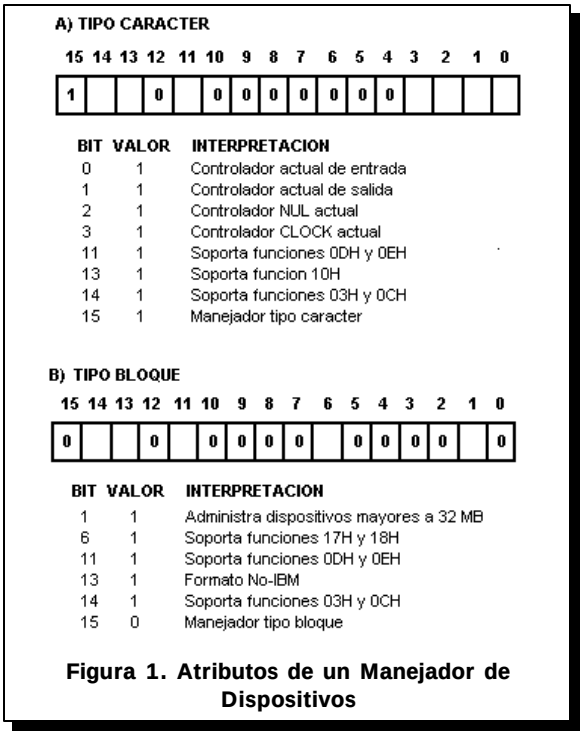
En MS-DOS se dispone de dos tipos de manejadores, correspondiendo a los dispositivos de carácter (*character*) y de bloque (*block*), respectivamente; los dispositivos de carácter transmiten información byte por byte, mientras que los de bloque comunican conjuntos de información, obviamente denominados bloques. Aunque los dos tipos difieren en varios aspectos, su estructura general es la misma. Cada controlador de dispositivos consiste básicamente de información de estado (*status*), y de una serie de rutinas para controlar al dispositivo; la comunicación se basa entonces en un protocolo de llamadas a dichas rutinas.

La estructura básica incluye 6 módulos principales:

1. Encabezado (*header*), área de datos con una longitud de 18 bytes, que proporciona la información necesaria para la activación del *driver*. Las 2 primeras palabras del encabezado (bytes 00H a 03H) indican la dirección relativa (desplazamiento - segmento) del controlador siguiente en la cadena interna del sistema. Al escribir el *driver*, el programador debe inicializar estos datos con el valor -1 (FFFFh); cuando se carga el manejador, DOS actualiza automáticamente estas referencias. La siguiente palabra corresponde al atributo, y en el se describe el tipo de manejador y sus características. Así, se tiene una distribución de los bits del atributo para los dispositivos de carácter, y otra para los de bloque; ambas interpretaciones se muestran en la figura 1, en la siguiente página.

Las dos palabras a continuación son apuntadores a los desplazamientos (*offset*) donde se encuentran las rutinas de estrategia y de interrupción, respectivamente. Finalmente, los últimos 8 bytes contienen el nombre del controlador, si este es tipo carácter, o el número de dispositivos lógicos soportados, para un manejador tipo bloque.

2. Tabla de salto a las funciones. Esta tabla contiene referencias al *offset* (dirección relativa) de cada



A) TIPO CARACTER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1			0		0	0	0	0	0	0	0				

BIT	VALOR	INTERPRETACION
0	1	Controlador actual de entrada
1	1	Controlador actual de salida
2	1	Controlador NUL actual
3	1	Controlador CLOCK actual
11	1	Soporta funciones 0DH y 0EH
13	1	Soporta funcion 10H
14	1	Soporta funciones 03H y 0CH
15	1	Manejador tipo caracter

B) TIPO BLOQUE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			0		0	0	0	0		0	0	0	0		0

BIT	VALOR	INTERPRETACION
1	1	Administra dispositivos mayores a 32 MB
6	1	Soporta funciones 17H y 18H
11	1	Soporta funciones 0DH y 0EH
13	1	Formato No-IBM
14	1	Soporta funciones 03H y 0CH
15	0	Manejador tipo bloque

Figura 1. Atributos de un Manejador de Dispositivos

una de las funciones (código) implantadas en el manejador.

3. Rutina de estrategia. Esta rutina, a pesar de su nombre, no plantea la forma de operar del controlador. Su función es establecer la comunicación entre DOS y el manejador, para inicializar a este último. Cuando DOS va a invocar a un controlador, se construye un encabezado de petición (*request header*) en un área reservada de memoria, y su dirección se transmite a la rutina de estrategia por medio de los registros ES:BX. El encabezado de petición sirve como espacio de comunicación, en donde DOS indica al controlador que debe hacer, y el manejador responde con el resultado de la operación. La figura 2 muestra una estructura típica del encabezado de petición.

+00H	Longitud en bytes del bloque de datos	1 word
+01H	Número de comunicación del dispositivo	1 word
+02H	Código de comando	1 word
+05H	Reservado	8 bytes
+0DH	Descriptor del medio	1 byte
+0EH	Dirección del buffer (offset)	1 word
+10H	Dirección del buffer (segmento)	1 word
+12H	Número	1 word
+14H	Sector de inicio	8 bytes

Figura 2. Estructura del Encabezado de Petición

4. Rutina de interrupción. Esta porción del controlador realiza todo el trabajo de control del dispositivo. Primeramente debe guardar el estado de todos los registros y banderas del procesador, para evitar su modificación, restituyéndolos al final de la operación. Posteriormente obtiene, del encabezado de petición, el código de comando a ejecutarse, verificando si este código es válido para ejecución con las rutinas del manejador. Si la operación es permitida, transfiere el flujo del programa a la rutina especializada y la procesa. En caso contrario, se detiene el curso del programa, desplegando un mensaje de error.

5. Funciones soportadas por el manejador. En esta zona se implementan todas las rutinas de operación del controlador. Por ejemplo, un controlador de consola debe contener rutinas de lectura y almacenamiento del teclado, carácter por carácter, así como de despliegue de esta información. Todo manejador debe soportar al menos 13 funciones básicas, numeradas de 00H a 0CH, aún y cuando su única acción sea activar la bandera de REALIZADO, en la palabra de estado. Además, se cuenta con funciones adicionales que corresponden a alguna versión específica de DOS o al tipo de controlador (bloque o carácter); la siguiente lista muestra las funciones principales:

FUNCION	SIGNIFICADO
00H	Inicialización del controlador
01H	Verficación del medio
02H	Construir bloque de parámetros de BIOS
03H	Lectura IOCTL
04H	Lectura
05H	Lectura no destructiva
06H	Estado (*status*) de entrada
07H	Buffers de flujo de entrada
08H	Escritura
09H	Escritura con verificación
0AH	Estado de salida
0BH	Buffers de flujo de salida
0CH	Escritura IOCTL
0DH	Abrir
0EH	Cerrar dispositivo
0FH	Medio removible
10H	Salida hasta ocupado
11H	IOCTL genérico
13H	IOCTL genérico
17H	Obtener dispositivo lógico
18H	Determinar dispositivo lógico

Todas estas funciones reciben sus argumentos del encabezado de petición, y almacenan sus resultados dentro del mismo. Este tipo de mecanismo libera al sistema operativo de la necesidad de conocer la dirección de cada función del manejador; el DOS simplemente utiliza las rutinas de estrategia y de interrupción como interface a las diversas funciones implementadas en el controlador.

Para entender mejor lo anterior, a continuación se describe a los parámetros de mayor relevancia solicitados por el controlador, para procesar la Función 01h (verificación del medio, *media check*); en la figura 3 , se observa la estructura correspondiente al encabezado de petición para esta función.

Esta operación la utilizan los controladores de dispositivos de bloque, para determinar cuando el medio (disco) ha cambiado. Estos cambios se determinan comparando la información actual en memoria después del último acceso a disco, con la correspondiente al acceso anterior. Si hubo cambios, el sistema operativo debe releer la información del

PARAMETROS DE ENTRADA

OFFSET	CONTENIDO	TIPO
01H	Número del dispositivo	Byte
02H	Número de función	Byte
13H	Descriptor del medio	Byte

PARAMETROS DE RETORNO

OFFSET	CONTENIDO	TIPO
03H	Palabra de estado	Word
14H	Cambio en el medio? FFH=si, 00H=No sé, 01H=No	Byte
15H	Dirección del buffer con el nombre del volúmen (solo si el apuntador indica cambio del medio)	Apuntador

Figura 3. Función 01H: Verificación del Medio

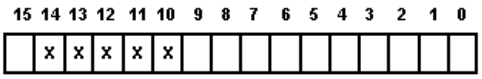
disco, lo cuál retardará la ejecución; en caso contrario, el contenido de memoria es correcto. Sin embargo, en medios tales como el disco flexible (*floppy disk*), la contestación a la pregunta *¿Hay cambios en el medio?* **es más complicada, por lo que DOS permite a la función contestar** *si* **o** *no*, **pero también** *no sé*.

Las posibles contestaciones dependen del manejo que realiza el sistema operativo con la información. Cuando el medio no ha cambiado, el acceso al mismo es inmediato; si hay cambios, DOS cierra los buffers relacionados con el dispositivo lógico actual, ocasionando la pérdida de los datos que debían ser transmitidos. Si la función contesta *no sé*, **la operación que realice el sistema operativo depende del estado del buffer relacionados con el dispositivo lógico actual; si está vacío asume que el medio ha cambiado y contesta entonces** *si*; **pero si contiene datos que debían ser transmitidos al medio, MS-DOS considera al medio como intacto y escribe en él, pudiendo ocasionar daños a la estructura del medio nuevo. En la figura 4 , se observa la estructura de la Palabra de Estado dentro del encabezado de petición, en la zona relativa a los códigos de error.**

6. Identificador. El nombre del controlador tiene doble uso, ya que además de proporcionar una identificación sirve para que, en caso de tener el nombre duplicado (esto es, otro manejador tenga el mismo nombre), el último en ser dado de alta sea el que se ejecute; lo anterior es sumamente útil cuando se requiere sustituir a un manejador por otro propio. En caso contrario, de tratarse de un controlador autónomo nuevo, éste se anexa a la cola de controladores del sistema.

IMPLEMENTACIÓN

Una vez diseñado el controlador de dispositivos, su implementación se basa en el manejo de herramientas de programación en lenguaje ensamblador, tales como TurboAssembler (Borland) y MacroAssembler (Microsoft). A continuación se incluye el listado de un manejador ejemplo, el cual reemplaza al controlador de consola estándar (CON); en el se indica claramente cada uno de los componentes de cualquier controlador.



BIT	VALOR	INTERPRETACION
8	1	Llamada ejecutada (ready)
9	1	Manejador ocupado (busy)
10-14	X	No usados
15	1	Sin error
15	0	Error

Si se presenta error, los bits 0 a 9 indican el tipo de fallo:

CODIGO	ERROR
0	Medio protegido contra escritura
1	Dispositivo desconocido
2	Manejador no listo
3	Comando desconocido
4	Error de lectura del CRC
5	Bloque de datos del parámetro
6	Error de búsqueda
7	Medio desconocido
8	Sector no encontrado
9	Impresora sin papel
10	Error de escritura
11	Error de lectura
12	Error común
15	Cambio ilegal de medio

Figura 4. Códigos de Error en la Palabra de Estado

Para probar algún manejador recién programado, lo recomendable es incluirlo en un disco flexible de arranque, para evitar daños en el sistema instalado en las unidades de disco duro. Como ya se mencionó, el archivo CONFIG.SYS constituye la vía de inclusión para los controladores, por lo que se debe agregar la línea

```
Device = X
```

donde X representa tanto la trayectoria donde está situado el archivo del controlador como la especificación del mismo (nombre y extensión, normalmente SYS).

CONSIDERACIONES FINALES

Los controladores de DOS tienen un inconveniente: dada sus estructura especial, solo pueden escribirse en lenguaje ensamblador. Sin embargo, esta aparente deficiencia se compensa con la mayor velocidad de operación de las rutinas resultantes, comparadas con programas equivalentes desarrollados en lenguajes de alto nivel. Las características de este tipo de programas los asemejan (en reglas de estructura) con los programas con extensión COM, pero se diferencian en que los controladores inician su código en el desplazamiento 0h, mientras que los programas COM lo hacen en el *offset* **100h.**

También debe mencionarse que en ocasiones la comunicación entre el sistema y un manejador puede resultar sumamente complicada para el programa-

dor, debido a la falta de documentación sobre las diversas funciones soportadas para un dispositivo. Este problema puede ser aún más importante si se considera la existencia de muchas características de MS-DOS no documentadas, por lo que su aplicaciòn puede producir resultados inesperados.

A pesar de estos aparentes inconvenientes, los manejadores de MS-DOS constituyen una alternativa flexible y poderosa para implementaciones de control por computadora. La facilidad para integrarlos al sistema permite una actualización permanente, sin efectos negativos sobre el rendimiento.

BIBLIOGRAFÍA

[1] **Abel, Peter.** *"IBM PC Assembly Language and Programming"***. Prentice-Hall, 1995.**

[2] **Tischer, Michael.** *"PC Intern"***. ABACUS 1995.**

[3] **Barkakati; Nabajyoti & Hyde, Randall.** *"Microsoft Macro Assembler Bible"***. SAMS 1992.**

[4] **Dettmann, Terry.** *"DOS Programmer´s Reference"***. QUE 1990.**

[5] **Schulman, Andrew.** *"El DOS no Documentado"***. Addison Wesley 1995.**