La pelea por el modo protegido

El MS−DOS es un sistema operativo del modo real, lo que supone un gran atraso frente a la aparición de los nuevos procesadores con capacidades multitarea como en el caso de los 80386 y superiores. Estos procesadores desarrollan su plena capacidad una vez que han conmutado al modo protegido y es por ello por lo que muchos desarrolladores de software se han parado a pensar el cómo integrar este modo de funcionamiento bajo el DOS.

Utilidades que se ejecutan en el modo protegido bajo el DOS, sin que éste se de cuenta de ello, pueden encontrarse en la forma de emuladores EMS, gestores de memorias, DOS−Extender y Multitaskers.

El problema de utilizar el modo protegido bajo el DOS parece ya resuelto pero esto no es cierto. Existe otro problema que surge cuando varias utilidades del modo protegido quieren ejecutarse a la vez. Cuando un gestor de memoria, que suele ser el primer programa en ejecutarse al arrancar el ordenador, utiliza el modo protegido para lograr el acceso a direcciones de memoria más allá del primer Mbyte, conmutará al procesador al modo V86 para que los programas DOS puedan ejecutarse y utilizar las nuevas dimensiones de memoria. Cuando se ejecuta un programa que utiliza el modo protegido como un Multitasker bajo un gestor de memoria, el Multitasker intentará conmutar el procesador al modo protegido, pero esta acción violará las protecciones del procesador, ya que se encuentra en el Modo V86, y se denegará la ejecución al Multitasker.

En este capítulo vamos a resumir brevemente cada uno de los diversos tipos de programas que utilizan el modo protegido bajo el DOS y luego comentaremos los diversos interfaces que se han creado para que estos programas puedan ejecutarse a la vez.

- **Utilidades del modo protegido**

El modo V86 es especialmente adecuado para utilidades del modo protegido que se emplean bajo el DOS. Ya que gracias a este modo de trabajo del procesador, se pueden crear un especie de programas de monitoreo, que se ejecutan en el modo protegido, pero que controla varias máquinas DOS virtuales bajo el modo V86. Se puede decir que el DOS obtiene una especie de padre, que controla y conduce todos sus pasos, sin que se entere en absoluto de ello.

- **Programas de gestión de memoria**

Bajo el MS−DOS no es posible acceder más allá del primer Mbyte de memoria sin utilizar un programa de gestión de memoria. Estos programas, encontrados en la forma de QEMM, 386ToTheMax, EMM, etc, acceden al modo protegido y ejecutan a los programas DOS bajo el modo V86.

La utilización del modo protegido bajo estos programas es debido, como comentamos antes, a que en este modo de trabajo es posible de direccionar hasta 4 Gbytes de memoria física, lo cual es bastante tentador. El gestor de memoria pondrá a disposición de todos los programas DOS una interface o servicios para que se pueda acceder a la memoria por encima del primer Mbyte. Se han desarrollados dos interfaces estándar conocidos como XMS y EMS y estos son ofrecidos por todos los gestores de memoria para el DOS.

Bajo el estándar XMS se definen diversas funciones para permitir el acceso a la memoria extendida. Los programas DOS pueden reservar cualquier cantidad de memoria extendida libre a través de estas funciones, así como liberar la memoria alojada o modificar el tamaño de la memoria que ha reservado. Para utilizar la memoria extendida que han reservado los programas DOS, se han de utilizar funciones XMS para copiar el contenido de la memoria extendida a la memoria convencional y acceder a ella a través de la memoria

convencional. En la figura 7.1 podemos ver algunos de las funciones ofrecidas por el XMS.

En el estándar EMS la cosa cambia un poco, ya que los programas DOS no pueden alojar cualquier cantidad de memoria por encima de un Mbyte. Los programas DOS pueden acceder a la memoria expandida a través de un segmento específico de memoria convencional. El programa DOS indicará a través de funciones EMS que información de la memoria expandida quiere ver a través de esa ventana o segmento de memoria convencional. En la figura 7.2 se pueden observar algunos de los servicios ofrecidos por el estándar EMS.

Figura 7.1. Servicios de la XMS

- **Multitaskers**

Los Multitasker es otro de los tipos de programas que utilizan el modo protegido bajo el MS−DOS y que conmuta luego el procesador a modo V86 para permitir la ejecución paralela de varios programas DOS. Los Mutitaskers se han venido explicando a lo largo de los anteriores capítulos, por lo que prescindimos aquí de su explicación.

- **DOS−Extender**

Un DOS extender es un programa que suele ir ligado a un compilador y su función es la de preparar el procesador para ejecutar un programa generado por el compilador en el modo protegido. Cuando un programador realiza un programa bajo un compilador que posea un DOS−Extender, el compilador insertará al programa realizado el código correspondiente al DOS−Extender, para que al ser ejecutado se conmute el procesador al modo protegido y el resto de código del programador se ejecute en este modo.

Figura 7.2. Algunos servicios de la EMS

Todos los DOS−Extender están asociados a un compilador específico, ya que el compilador utilizará llamadas a su DOS−Extender para que le ofrezca diversos servicios desde el modo protegido. Además, el compilador ha de conocer las estructuras que ha creado el DOS−Extender en el modo protegido, como la GDT, para que pueda acceder y manipular a la memoria protegida y genere el código de una forma u otra. Uno de los servicios que ha de poseer el DOS−Extender es permitir al compilador llamadas al MS−DOS o la BIOS para que el programa de usuarios pueda solicitar estos servicios.

Existen DOS−Extender para programas que se realizan en ensamblador. El programador que quiera realizar un programa en modo protegido en ensamblador deberá de incluir el código fuente del DOS−Extender al principio de su programa, e implementar el resto del programa ensamblador teniendo en cuenta que se va a ejecutar en modo protegido, por lo que el modo de direccionamiento a memoria, por ejemplo, ya no es el igual que en el modo real.

- **DPMI y VCPI**

Cuando uno de los programas anteriores conmutan a modo protegido y ejecutan un programa bajo él, no es posible ejecutar un programa que necesite acceder al modo protegido. Por ejemplo, si tenemos un Multitasker y bajo el se quiere ejecutar un programa que haya sido realizado con la ayuda de un DOS−Extender, éste último intentará conmutar el procesador de modo V86 a modo protegido, pero el procesador le denegará el acceso. Esto es así debido a que los programas que se ejecutan bajo un gestor de memoria o un Multitasker, poseen un nivel de privilegio que no es el de supervisor, por tanto, un intento de ejecutar una instrucción privilegiada, como cambiar el procesador a modo protegido, no será permitido por el procesador. Para solucionar este problema, se han creado dos especificaciones, DPMI y VCPI, que permiten la coexistencia pacífica de varios programas para el modo protegido en memoria.

- **VCPI (Virtual Control Program Interface)**

Cuando se instala un gestor de memoria, tras arrancar el ordenador, se conmuta el procesador a modo V86 con nivel de privilegios 3 para aprovechar las propiedades de paginado del procesador y ofrecer de esta forma mayor cantidad de memoria a los programas. Para permitir que programas como un Multitasker o DOS−Extender puedan acceder al modo protegido, el gestor de memoria instalará un servidor VCPI.

El servidor VCPI ofrece un total de 13 funciones distintas, que cubren las distintas zonas en las que puede ocurrir una colisión entre las utilidades del modo protegido. Estas funciones se pueden dividir en 5 grupos:

- Tres funciones se dedican a la inicialización del VCPI.
- Cuatro funciones ayudan en la gestión de la memoria extendida.
- Tres funciones permiten el acceso al primer registro de control (CR0) y a los registros de depuración del procesador.
- Dos funciones ayudan a la programación del controlador de interrupciones
- Una función es la responsable de la conmutación entre el modo protegido y el modo V86.

En la figura 7.3 se muestra cada uno de los servicios ofrecidos por un servidor VCPI. Para más detalle sobre cómo utilizar estas funciones remítase a [VCPIspec].

El servidor VCPI, como ha podido observarse, ofrece posibilidades para el acceso al modo protegido, no obstante, los servicios ofrecidos son pobres y el programador se ha de preocupar de muchos detalles de implementación, como pueden ser la emulación de servicios DOS desde el modo protegido.

Respecto a como se ha implementado DMT, al principio se intentó implementar bajo un servidor VCPI, pero se observó que VCPI necesita ciertas restricciones que deberían de conservar los programas que usaran VCPI (restricciones como no cambiar las primeras entradas de la GDT). Estas restricciones no permiten la realización de DMT o un Multitasker en general, por tanto para utilizar VCPI bajo un Multitasker, se ha de violar las restricciones de VCPI, lo que resulta bastante arriesgado para el funcionamiento global del sistema.

Figura 7.3. Servicios del VCPI

- **DPMI (DOS Protected Mode Interface)**

El Interface DPMI nace tras el desarrollo de Windows 3.0 con el objetivo de hacer posible la ejecución de varios programas Windows en la memoria extendida. DPMI cubre un gran espectro de funciones que están disponibles para la ejecución de varios programas que fueron creados para el modo protegido. Podemos dividir este gran conjunto de funciones en los siguientes grupos:

- Gestión de tablas de descriptores de un programa para el modo protegido.
- Gestión y alojamiento de la memoria extendida.
- Gestión de interrupciones y excepciones.
- Comunicación con programas del modo real desde modo protegido.
- Acceso a los diferentes registros del procesador.
- Virtualización del DMA.

En la figura 7.4 puede observarse algunas de las funciones ofrecidas por el interface DPMI. Como puede observarse el DPMI es muy superior a VCPI, ya que ofrece muchas más funciones para la utilización del modo protegido y aprovecha al máximo todas las características de este modo de trabajo.

Figura 7.4. Algunos servicios del DPMI

El DPMI tampoco se ha utilizado para la implementación de DMT, debido a que no encontrado ninguna forma de implementarlo bajo este interface. El DPMI es más que nada para programas que quieren ejecutarse en el modo protegido teniendo acceso a mayor cantidad de memoria, pero no está pensado para crear un Multitasker, ya que no contiene funciones para crear nuevas tareas y ponerlas todas en ejecución paralela.

La pelea por el modo protegido **53**

47

Número de función	**Descripción**
00h	Obtener número de versión de XMS
01h	Obtener posesión de la HMA
02h	Liberar HMA
03h	Activación globlal de la linea A20
04h	Cierre de la línea A20
05h	Liberación local de la línea A20
06h	Bloqueo local de la línea A20
07h	Obtener estado de la línea A20
08h	Obtener tamaño de la memoria extendida libre
09h	Aloja un Extended Memory Block (EMB)
0Ah	Liberación de un Extended Memory Block
0Bh	Copia Memoria
0Ch	Bloquea un Extended Memory Block
0Dh	Desbloquea de nuevo un Extended Memory Block
0Eh	Obtener informaciones sobre un EMB
0Fh	Aumentar o reducir un EMB alojado
10h	Alojar Upper Memory Block (UMB)
11h	Liberar de nuevo un UMB alojado

Número de función	**Descripción**
40h	Obtener estado del EMM
41h	Obtener dirección base del segmento de paginas
42h	Obtener número de páginas
46h	Obtener versión del EMM
47h	Salvar contexto de una página
4Eh	Obtener/Establecer mapa de páginas parcial
53h	Obtener/Establecer nombre de Handle
5Ah	Alojar Páginas
57h	Mover/Intercambiar región de memoria

Número de función	**Descripción**
DE00h	Detectar si VCPI está presente
DE01h	Obtener Interfaz con el modo protegido
DE02h	Obtener dirección máxima de memoria física
DE03h	Obtener número de páginas de 4 KB libres

DE04h	Alojar una páginas de 4 KB
DE05h	Liberar una página de 4 KB
DE06h	Obtener dirección de una página de 4 Kb en el primer Mbyte de memoria
DE07h	Leer registro de estado CR0
DE08h	Leer registros de depurado
DE09h	Cargar registros de depurado
DE0Ah	Obtener mapa del 8259A
DE0Bh	Establecer mapa del 8259A
DE0Ch	Pasar de V86 a modo Protegido

Número de Función	**Descripción**
1687h	Determinar di DPMI está disponible
0000h	Alojar descriptor de segmento LDT
0001h	Liberar descriptor de segmento LDT
0004h	Bloquear segmento contra descarga
0005h	Permitir descarga de un segmento
0006h	Obtener dirección base de un segmento
0008h	Fijar longitud de un segmento
000Ah	Crear Alias para segmento de código
000Dh	Pedir un selector determinado
0100h	Pedir memoria DOS
0101h	Liberar memoria DOS
0102h	Modificar tamaño de un bloque de memoria
0200h	Obtener dirección del controlador de interrupciones desde el modo real
0201h	Fijar controlador de interrupciones
0203h	Instalar controlador de excepciones
0900h	Bloquear bandera de interrupción virtual
0901h	Liberar bandera de interrupción virtual
0902h	Controlar bandera de interrupción virtual
03000h	Simular interrupción del modo real
0301h	Llamar rutina del modo real
0400h	Obtener número de versión DPMI