-

Modo Real

El modo real es el único modo de trabajo que posee el 8086, por lo que el MS−DOS al querer mantener la compatibilidad con estos procesadores, trabaja únicamente en modo real. El 80386 puede correr en este modo como si se tratará de un 8086 de altas prestaciones. Este modo de trabajo da la máxima libertad al programador para realizar cualquier operación y controlar todo el sistema a su antojo, lo que supone una gran ventaja para el programador pero un gran inconveniente para el sistema operativo, ya que no puede controlar los pasos seguidos por el programador.

En este capítulo describiremos brevemente este modo de trabajo, debido a su relativa sencillez y a que es el menos utilizado por DMT.

- **Gestión de la memoria en modo real**

En modo real no existe ninguno de los mecanismos mencionados anteriormente, por lo que desaparecen los conceptos de direcciones lógicas y lineales. Toda dirección dada por un programa en modo real es una dirección física.

La memoria en modo real se divide en segmentos de longitud fija (64 Kb) y para acceder a una dirección específica hay que indicar el segmento en el que se encuentra el dato y el desplazamiento dentro de ese segmento. Por tanto, todas las direcciones poseen el formato *segmento:desplazamiento*. En la figura 6.1 se muestra cómo se divide la memoria en segmentos dentro del modo real. Para formar la dirección física, el procesador multiplica el valor del segmento por 16 y suma el desplazamiento.

Figura 6.1. División de la memoria RAM en un PC

Como puede observarse en la figura 6.1, bajo el modo real sólo existen 16 segmentos de 64 Kb, lo que supone un Mbyte de memoria física. Esto es así debido a que el 8086 posee registros de 16 bits y a través del direccionamiento *segmento:desplazamiento* no es posible acceder a más de un Mbyte de memoria. Bajo el modo real, por tanto, no es posible acceder más allá de la barrera de un Mbyte, lo que supone un gran atraso usar hoy en día dicho modo bajo el 80386 y superiores.

- **Una sola tarea ejecutándose en modo real**

Bajo el modo real no existe ningún mecanismo para tener varias tareas ejecutándose concurrentemente, aunque si se pueden tener varios programas cargados en memoria al mismo tiempo, pero inactivos.

En el modo real se puede lograr un pequeño grado de multitarea si interceptamos algunos de los vectores de interrupción que se ejecutan periódicamente, como el temporizador, e insertamos una rutina de usuario bajo esa interrupción. Como puede verse, este tipo de multitarea no tiene ni punto de comparación con la que ofrece el modo protegido del 80386.

- **Escasa protección en modo real**

En el modo real no se puede controlar que un programa, por ejemplo, no pueda acceder a una zona específica de memoria. Un programa en modo real puede acceder a cualquier dirección de memoria direccionable por el procesador, con lo que un programa maligno puede modificar estructuras del sistema operativo o redireccionar vectores de interrupción y tener un control total sobre el sistema.

Debido a esta escasa protección que se ofrece en el modo real, aparecen los temibles *virus informáticos* y otros programas malignos. Tras conocer el modo protegido y haber trabajado con él, me atrevo a afirmar que es imposible realizar un virus para un sistema operativo que trabaje en modo protegido, siempre y cuando ese sistema operativo no deje ninguna puerta de entrada libre al usuario para controlar el sistema.

- **Acceso a puertos de E/S**

El mecanismo que se utiliza en el modo protegido para controlar los accesos a puertos de E/S desaparece bajo el modo real. Cualquier programa ejecutado en modo real puede acceder a cualquier puerto de E/S direccionable por el procesador, lo que supone otra forma de violar el sistema por parte de un programa mal intencionado.

- **Interrupciones en el modo real**

Todas las interrupciones hardware y software son controladas por el procesador en el modo real leyendo de una tabla de interrupciones que se encuentran en las primeras posiciones de memoria física. Cuando se produce una interrupción, se lee de la tabla anterior la dirección donde encontrar su rutina de tratamiento. Como cualquier programa pude acceder a cualquier dirección de memoria, pude manipular este área de memoria e interceptar diversos vectores de interrupción y apuntarlas a rutinas propias, con lo que cualquier programa puede controlar el sistema según desee.

- **Solicitud de servicios del sistema operativo**

Para solicitar alguno de los servicios que ofrece el sistema operativo, se ha de ejecutar una instrucción del tipo INT *n* con algunos registros inicializados. Esta instrucción leerá la dirección del servicio desde la tabla de interrupciones mencionada arriba, y se pasarán a ejecutar las instrucciones que se encuentren en la dirección obtenida.

Modo Real **46**

44

Bloque	**Dirección**	**Contenido**
15	F000:0000 − F000:FFFF	BIOS−ROM
14	E000:0000 − E000:FFFF	Libre para cartuchos ROM
13	D000:0000 − D000:FFFF	Libre para cartuchos ROM
12	C000:0000 − C000:FFFF	BIOS−ROM adicional
11	B000:0000 − B000:FFFF	Video RAM
10	A000:0000 − A000:FFFF	Video RAM (VGA/EGA)
9	9000:0000 − 9000:FFFF	RAM de 576 KB a 640 KB
8	8000:0000 − 8000:FFFF	RAM de 512 KB a 576 KB
7	7000:0000 − 7000:FFFF	RAM de 448 KB a 512 KB
6	6000:0000 − 6000:FFFF	RAM de 384 KB a 448 KB
5	5000:0000 − 5000:FFFF	RAM de 320 KB a 384 KB
4	4000:0000 − 4000:FFFF	RAM de 256 KB a 320 KB
3	3000:0000 − 3000:FFFF	RAM de 192 KB a 256 KB
2	2000:0000 − 2000:FFFF	RAM de 128 KB a 192 KB
1	1000:0000 − 1000:FFFF	RAM de 64 KB a 128 KB
0	0000:0000 − 0000:FFFF	RAM de 0 KB a 64 KB