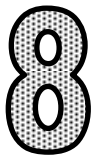
MEMORIA SEGMENTADA



8.1- ORGANIZACIÓN DE LA MEMORIA.

La memoria que controla en Pentium está organizada en bytes, palabras, dobles palabras y cuádruples palabras. El byte es el elemento básico de la memoria del Pentium. Las palabras se almacenan en dos bytes, quedando el byte de menor peso en la dirección inferior. Las dobles palabras ocupan cuatro bytes consecutivos, depositándose el byte inferior en la dirección más baja y el más significativo en la dirección superior (numérica). Una cuádruple palabra consta de ocho bytes de direcciones consecutivas.

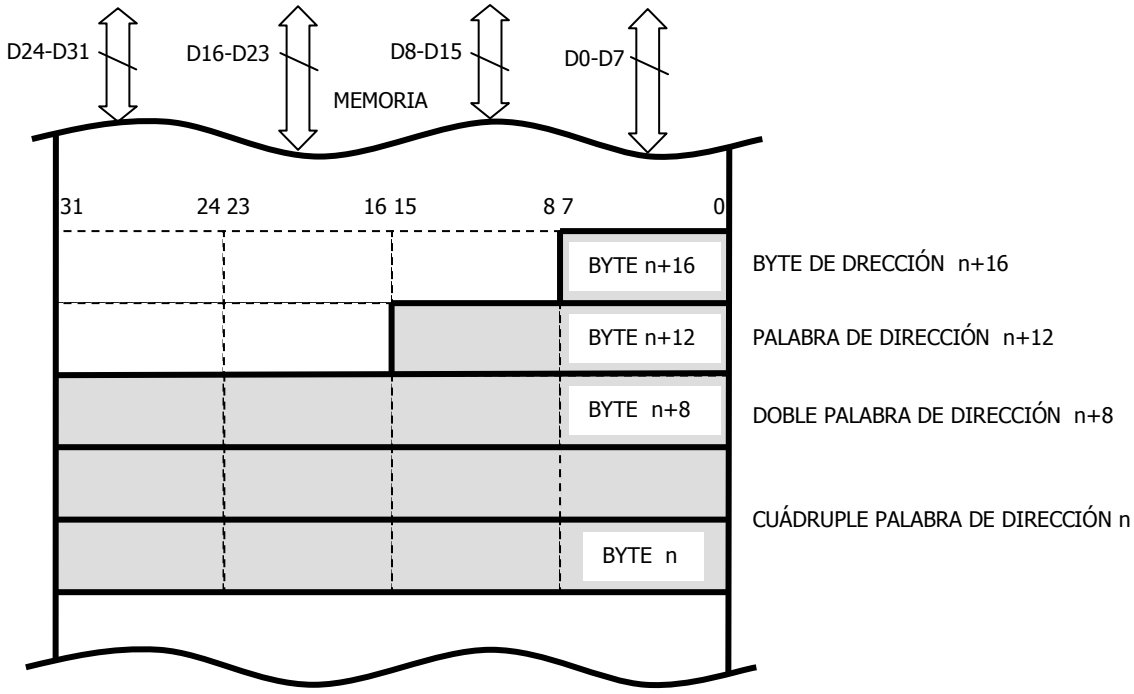


Figura 8.1. Distribución en la memoria de bytes, palabras, dobles palabras y cuádruples palabras.

Dado que el bus de datos es de 32 líneas divididas en 4 grupos de 8, los de dobles palabras conviene que comiencen en una dirección múltiple de 4, para que un ciclo en un ciclo los 32 bits que se componen se transfieran por el bus de datos. Si no cumple esta condición de transferencia de la doble palabra se realiza en dos ciclos. Para evitar estas penalizaciones de tiempo las cuádruples palabras deben ocupar direcciones de múltiplo de 4 y las palabras las direcciones pares

Para solucionar este problema la memoria debe estar "alineadas", es decir que las palabras se encuentren en dirección par y las dobles palabras en direcciones múltiplos de cuatro.

También existen instrucciones que reconocen los siguientes tipos de datos: enteros, ordinales, enteros BCD empaquetados y sin empaquetar, punteros de direcciones, campos, cadenas y números en coma flotante.

Además de los tipos básicos de estructuras de datos mencionadas, el Pentium maneja otros dos mucho más complejos:

1. Segmentos
2. Páginas.

Los segmentos son bloques de memoria de tamaño variable, que contienen información de la misma clase y constituyen el objeto principal sobre el que se basa el mecanismo de protección. La

segmentación es eficaz para organizar la memoria en módulos lógicos de similares características, que son el soporte de la programación estructurada. Es posible compartir recursos entre todas las tareas si se sitúan en el espacio global, o bien, ser exclusivos de una tarea concreta si están ubicados en el área local de la misma. Se puede asignar a cada segmento un determinado nivel de privilegio. Intel basa el control de la memoria en la segmentación por eso siempre está activada.

La paginación divide el espacio de memoria en trozos de longitud fija, llamados páginas, que, en el caso del Pentium, tienen un tamaño de 4 KB ó 4 MB. La paginación simplifica la labor del programador de sistemas al simplificar los algoritmos de intercambio entre objetos de la memoria física y la memoria virtual, al manejar elementos del mismo tamaño lo que supone un incremento de la velocidad en la localización y relocalización de páginas. La paginación es optativa y sólo funciona cuando la activa el programador.

La paginación y la segmentación son técnicas complementarias y ambas introducen ventajas particulares en la gestión de la memoria virtual. Los sistemas operativos UNIX y DOS son, respectivamente, ejemplos representativos de dichas técnicas.

El Pentium es capaz de combinar ambos métodos cuando funciona con segmentación paginada. De esta forma, el programador de aplicaciones estructura la memoria lógica en segmentos que soportan la multitarea, mientras el programador de sistemas emplea la paginación en el manejo y transferencia de bloques en la memoria física.

Cuando sólo se precisa trabajar con la paginación, al estar activa siempre la segmentación, es preciso que toda la memoria física se considere como un único segmento, que ocupa un espacio continuo o lineal. El modelo "plano" permite esta posibilidad.

En la segmentación paginada, el Pentium utiliza la paginación para descomponer los segmentos en páginas de 4 KB ó de 4 MB de tamaño, que se distribuyen aleatoriamente en la memoria física. No todas las que conforman un segmento tienen que residir en ella, sino sólo las que van a ser procesadas.

8.2- LA MEMORIA EN MODO REAL.

Cuando el procesador trabaja en Modo Real, lo hace de forma similar al 8086 con el fin de ser compatible con el software creado para dicho microprocesador de 16 bits.

Siempre que se realiza la operación RESET o de puesta en marcha, el Pentium comienza trabajando en Modo Real.

En la memoria en Modo Real, se encuentran 8 registros de propósito general de 16 bits (AX, BX, CX, DX, SP, BP, SI y DI), aunque también se pueden encontrar los registros extendidos de 32 bits (EAX, EBX, ECX, EDX, ESP, EBP, ESI y EDI) que son accesibles para programas que utilizan una operación para ignorar el tamaño de los operandos. Además de estos registros generales se encuentran otros que son los denominados registros de segmento (CS, DS, SS, ES, FS y GS). Aunque los registros FS y GS son para programas que explícitamente acceden a ellos.

En este modo se contempla un espacio de direcciones directamente accesibles por la CPU de 1MB (solo usa 20 líneas de direcciones), en el que sólo es posible aplicar la técnica de la segmentación. Se multiplica el contenido del registro segmento por 16 (se le añaden cuatro ceros) para hallar la base del mismo y luego para calcular la dirección a acceder, se suma al resultado obtenido, un desplazamiento de 16 bits, lo que implica que el tamaño máximo del segmento es de 64 KB. Como se ve en Modo Real los selectores de segmento son utilizados para formar la dirección lineal.

```
DIRECCIÓN EFECTIVA = RS x 16 + DESPLAZAMIENTO
```

Para direccionar una instrucción en el segmento de código, se usa como registro de segmento a CS y como desplazamiento el contenido de IP, que es la parte baja (16 bits) del Registro Puntero de Instrucciones (EPI).

Si se hace referencia a la pila, SS actúa como registro de segmento y SP como desplazamiento (palabra baja de ESP). Finalmente, si se desea direccionar un dato, se toma como registro a DS por defecto, ya que también puede actuar ES, FS o GS, si se explicita en la instrucción. Como desplazamiento se usa el que se indique en la propia instrucción del operando. Ejemplo:

```
MOV EBX, ES: DESPLAZAMIENTO
```

Esta instrucción mueve al registro EBX una información de 32 bits situada en la memoria en la dirección ES x 16 + DESPLAZAMIENTO.

En la figura 8.2 se muestra un ejemplo de direccionamiento de la memoria en Modo Real. En cada momento, la CPU tiene activados a un segmento de código (referenciado por CS), otro de pila (que selecciona con SS) y hasta cuatro datos, referenciados por DS, ES, FS y GS.

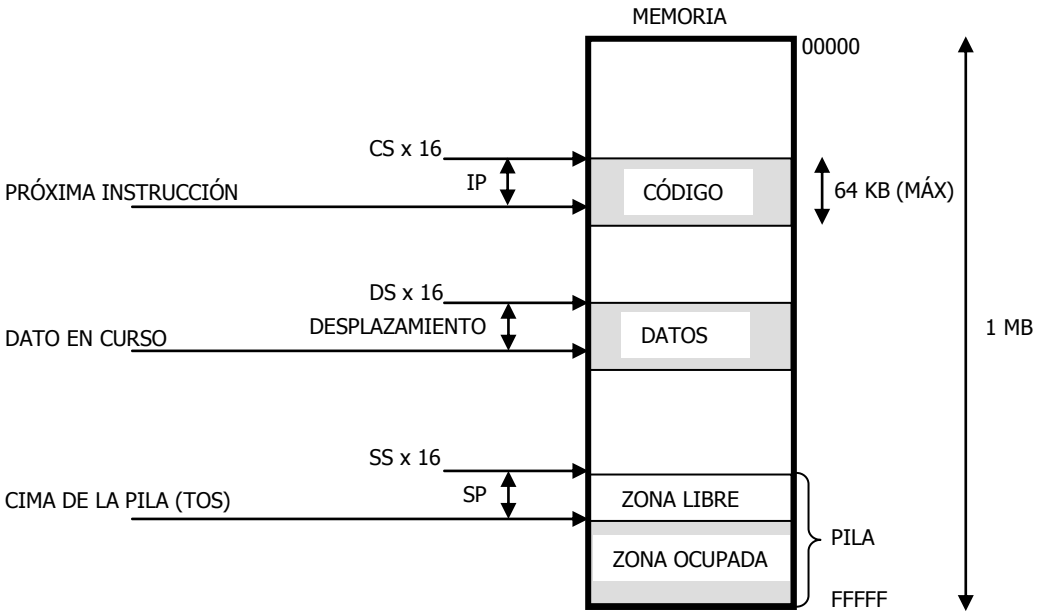


Figura 8.2. La manipulación de los registros de segmento y los desplazamientos permite referenciar en cada momento a las posiciones de memoria que contienen la próxima instrucción, el dato en curso y la cima de la pila.

Aunque en Modo Real el Pentium manipula, por defecto, operandos y desplazamientos de 16 bits, es posible usar elementos de 32 bits de longitud. Además, en este modo existen nuevas instrucciones respecto al 8086 y se han potenciado otras. Todo ello implica el uso de un "compilador" de lenguaje Ensamblador más complejo si se desea alcanzar el máximo rendimiento. Por otra parte, el Pentium precisa menos ciclos de reloj que el 8086 para ejecutar la mayoría de las instrucciones, de esta manera el tiempo de ejecución del software para el 8086 se reduce notablemente cuando se usa el Pentium.

En Modo Real, todas de las instrucciones específicas de protección de modo como LLDT (load local descriptor table) son tan inútiles como los códigos innecesarios.

Como el acceso a las posiciones de los segmentos se hace mediante direccionamiento relativo (desplazamiento) se obtiene como ventaja adicional la facilidad en la relocalización, característica esencial para soportar la portabilidad de los programas sobre diferentes configuraciones físicas.

8.3- LA MEMORIA EN MODO PROTEGIDO.

El Modo Protegido fue originalmente implementado en el 80826 para proteger a las diferentes tareas cuando el sistema está operando en multitarea, contra los posibles accesos incorrectos o inválidos.

Dentro de la memoria contemplada por el Pentium, se pueden distinguir tres espacios:

- Espacio virtual o lógico.
- Espacio lineal.
- Espacio físico.

El espacio virtual abarca toda la dimensión de la memoria virtual y es el que maneja el programador de aplicaciones. Como la dirección virtual es de 46 bits el tamaño del espacio virtual (memoria de masa o disco) es de 2^{46} =64 TB. La Unidad de Segmentación, cuya activación siempre es obligada, traduce las direcciones virtuales a lineales, que reciben este nombre porque hacen referencia a segmentos que, al situarse sobre la memoria física, tienen dispuestas todas sus posiciones en orden consecutivo o lineal. Cuando la Unidad de Paginación, optativa, no está activada, la dirección lineal coincide con la dirección física, es decir, la de acceso a la memoria principal ligada directamente a la CPU. Si la Unidad de Paginación está activada, cada segmento se descompone en un número variable de páginas del mismo tamaño (4KB ó 4MB), y la Unidad de Paginación deposita a dichas páginas sobre la memoria física en los huecos que encuentra libres, no una detrás de otra, lo que significa que la dirección lineal se debe traducir a física de acuerdo con esta distribución aleatoria de las páginas (figura 8.3). La memoria física o principal (DRAM) al direccionarse con 32 bits admite un tamaño de 2^{32} = 4 GB.

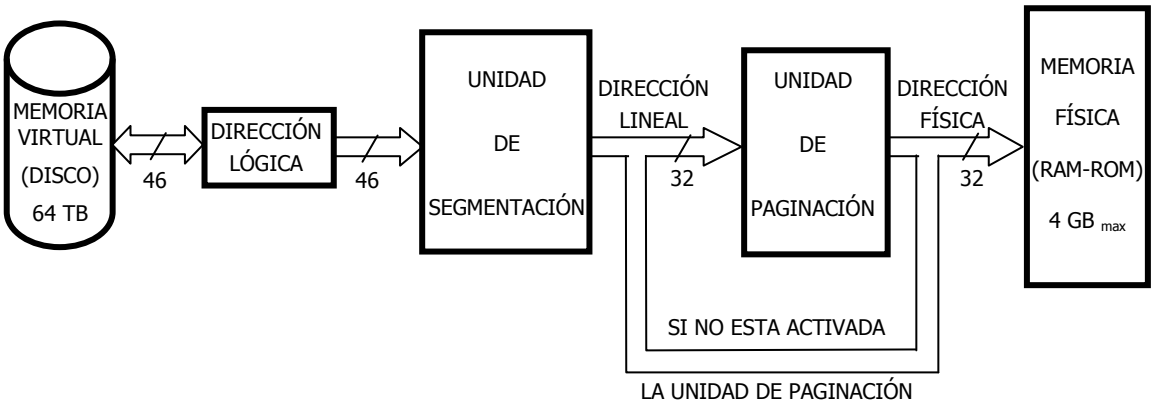


Figura 8.3.Traducción de la dirección lógica en dirección lineal y física.

Los programas de aplicaciones que procesa el computador sólo hacen referencia a direcciones virtuales y es la MMU (integrada en el hardware del Pentium) la que se encarga de traducirlas a direcciones físicas. Como la memoria física es mucho más pequeña que la virtual, la MMU también

deberá detectar la ausencia de los elementos que no estén cargados en la memoria física, para comunicárselo mediante una excepción al Sistema Operativo, el cual realiza el traslado de dichos elementos desde la memoria virtual a la física (figura 8.4).

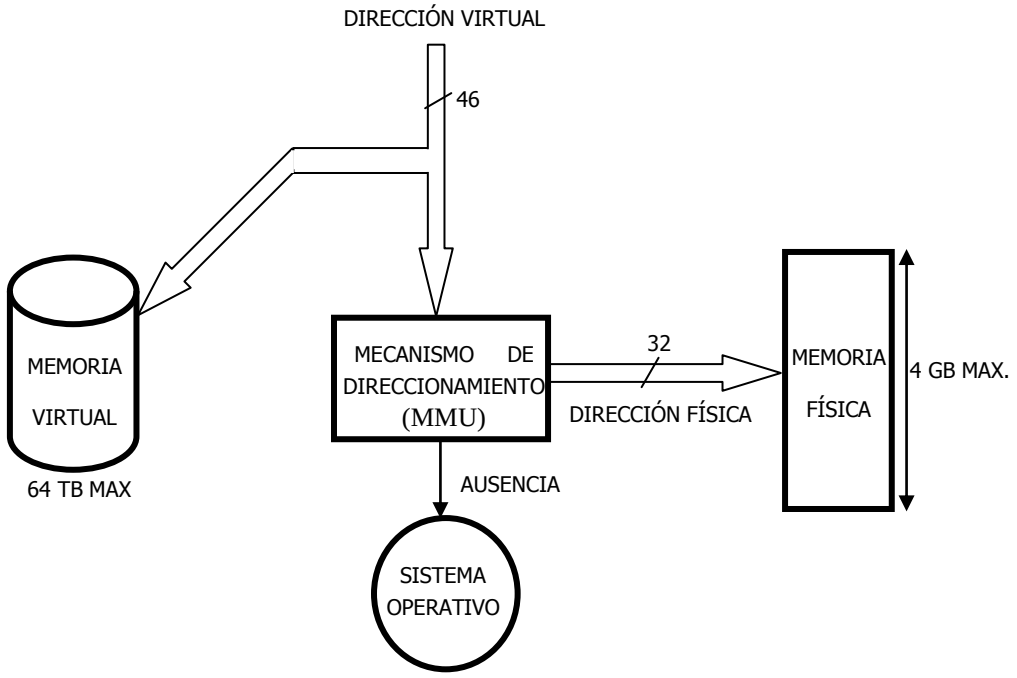


Figura 8.4. El mecanismo de direccionamiento traduce la dirección virtual a física, y, caso de no hallarse el elemento en la memoria, se detecta su 'ausencia' y se genera una excepción encargada de hacer la transferencia oportuna que es soportada por el Sistema Operativo.

8.4- EL ESPACIO VIRTUAL O LÓGICO.

La dirección lógica que usan los programadores de aplicaciones, consta de dos partes:

1. **Selector:** Es un campo de 14 bits, que selecciona un determinado segmento del espacio virtual.
2. **Desplazamiento:** Es un campo reducido de 32 bits que determina una posición de un segmento cuya capacidad máxima es de 4 GB. Este campo puede tener una longitud de 16 bits para mantener la compatibilidad con el software procedente de los microprocesadores de 16 bits (8086-80286), en cuyo caso los segmentos admiten un tamaño máximo de 64 KB.

En realidad, el puntero de direcciones virtuales tiene el siguiente formato:

```
SELECTOR(14): DESPLAZAMIENTO(32),
```

El campo selector está contenido en los 14 bits de más peso del registro de segmento y los dos restantes sirven para referenciar al nivel de privilegio del peticionario del segmento y no se utilizan en el direccionamiento propiamente dicho. Dicha pareja de bits forman el campo RPL. El campo RPL puede tener 4 niveles, teniendo un valor máximo cuando vale 00 y un valor mínimo cuando vale 11.

En el campo selector está contenido también el TI (Indicador de Tabla), que indica que la Tabla de Descriptor es local (TI=1) o global (TI=0), ayudando a localizar el segmento determinado en la memoria.

Como selector, actúa uno de los registros de segmento de 16 bits, en donde los 14 bits de más peso son discriminantes y direccionan un máximo de 16 K descriptores de segmento.

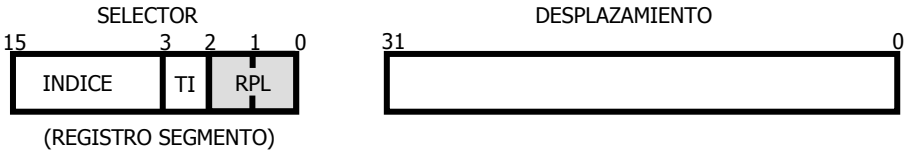


Figura 8.5. Una dirección lógica consta de un campo SELECTOR de 16 bits contenido en un registro segmento, y un campo DESPLAZAMIENTO de 32 bits.

El desplazamiento es un campo de 32 bits cuyo valor se suma a la base del segmento para hallar la dirección a acceder. Cuando se accede a código y CS actúa como selector el desplazamiento lo conforma el contenido de EIP. Si se accede a la pila, SS hace de selector y ESP de desplazamiento. Finalmente, cuando se accede a datos, uno de los registros de segmento DS, ES, FS o GS, actúa como selector, y el desplazamiento se calcula de acuerdo con el modo de direccionamiento utilizando en la instrucción en curso.

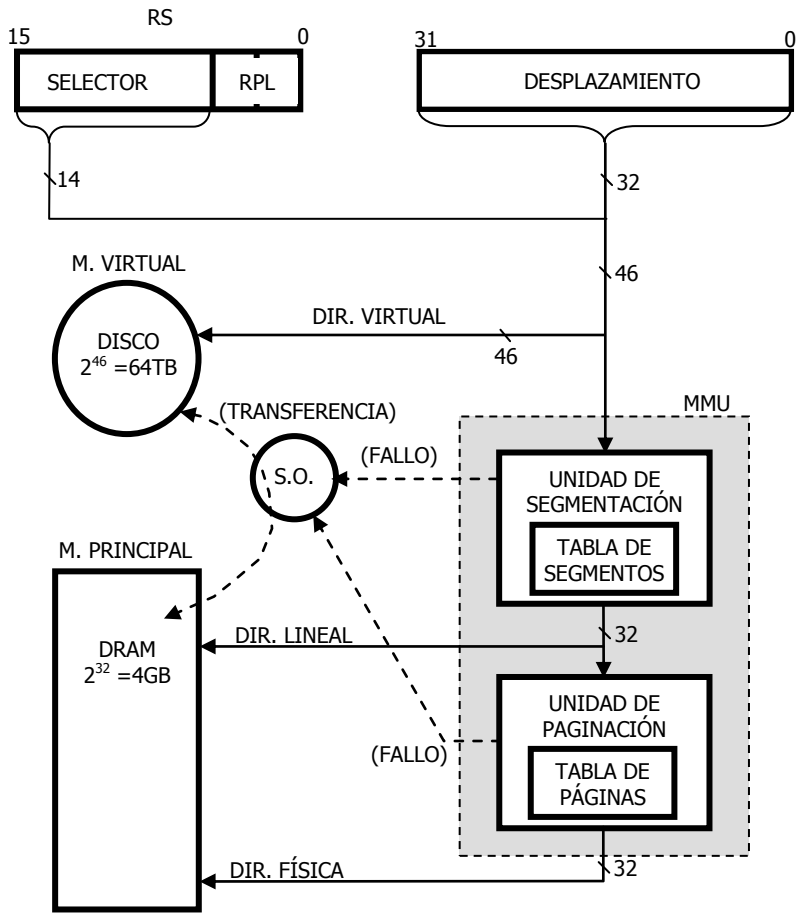


Figura 8.6.

Los 14 bits de más peso del Registro de Segmento (RS) determinan el selector del segmento en la memoria virtual y sus dos últimos bits indican el nivel de privilegio del peticionario (RPL), es decir, el nivel de privilegio del segmento que ha solicitado usar ese selector.

El Pentium dispone de un mecanismo hardware llamado MMU, Unidad de Manejo de Memoria, por el cual convierte la dirección virtual de 46 bits a dirección física de 32 bits. La MMU recoge la dirección virtual de 46 bits y lo introduce en la Unidad de Segmentación, que funciona siempre. La Unidad de Segmentación contiene la Tabla de Segmentos, que determina y registra los segmentos que están en la memoria principal y qué posición ocupan. Si el segmento solicitado está en la memoria principal, se traduce la dirección virtual a dirección lineal de 32 bits y se accede a dicha posición de memoria. Si por el contrario, no está presente (fallo), el Sistema Operativo realiza una transferencia que consiste en coger el segmento del disco y llevarlo a la memoria principal, actualizando también la Tabla de Segmentos, para dar curso a la traducción de dirección virtual a dirección lineal.

En el caso de que funcione la paginación, la dirección lineal que sale de la Unidad de Segmentación pasa a la Unidad de Paginación. La Unidad de Paginación contiene la Tabla de Páginas, que soporta la ubicación de las páginas en que se han dividido los segmentos y que están en memoria principal. Si la página está presente en memoria, la dirección lineal se traduce a dirección física de 32 bits. Si no está presente (fallo), avisa al Sistema Operativo para que este realice la transferencia correspondiente y actualice las tablas.

8.5- EL ESPACIO LINEAL.

La Unidad de Segmentación siempre se halla activada en el Pentium y se encarga de traducir la dirección lógica de 46 bits en dirección lineal de 32 bits. La dirección lineal coincide con la física correspondiente a la memoria principal si la Unidad de Paginación no está activada.

Se supone que no funciona la Unidad de Paginación dejando para el siguiente capítulo la explicación de su actuación.

Los objetos con los que opera la Unidad de Segmentación son los segmentos, por lo tanto en la memoria principal sitúa y mueve segmentos completos. De ahí proviene el nombre de dirección lineal, que significa que la segmentación referencia a bloques (segmentos) que tienen todas sus posiciones ordenadas consecutiva o linealmente.

Aunque la segmentación sea beneficiosa dada la adaptación de la memoria a las modernas técnicas de programación estructurada, tiene el inconveniente de trabajar con segmentos de tamaño variable, lo que complica el mecanismo de transferencia entre las memorias virtual y física, así como la optimización en el comportamiento del espacio de esta última (figura 8.7).

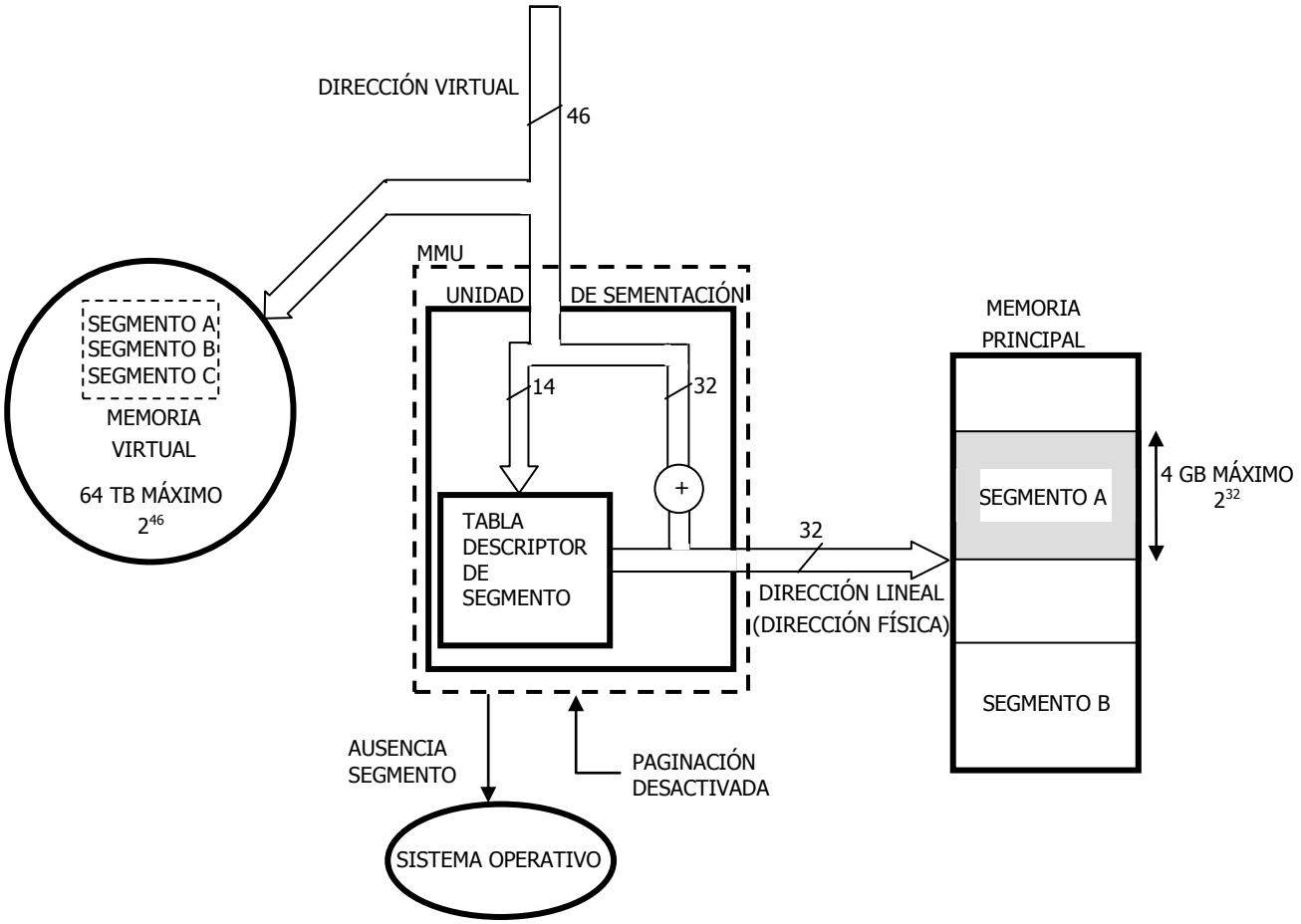


Figura 8.7. La Unidad de Segmentación, cuando está desactivada la Paginación, traduce las direcciones virtuales en lineales (físicas) y opera con segmentos completos, de forma que en todo momento la memoria principal contiene una parte de los segmentos existentes en la memoria virtual.

Cuando la dirección virtual hace referencia a un segmento que no está cargado en la memoria principal (puesto que no caben todos los existentes en la memoria virtual) la Unidad de Segmentación detecta su ausencia, o sea, el fallo del segmento. En esta situación, provoca una excepción, que pone en marcha una rutina del Sistema Operativo que se encarga de trasladar dicho segmento desde la memoria virtual a la memoria física. Al tener que manejar esta rutina transferencias de bloques de tamaño variable los algoritmos que la configuran son largos y complejos. Tras cargar el segmento en la memoria principal la rutina actualiza la tabla de segmentos y se da curso al acceso.

8.5.1- Descriptores de segmento.

En el modo protegido un segmento queda especificado por tres parámetros: Base de 32 bits, Límite o tamaño de 20 bits y Atributos de 12 bits. Al conjunto de estas informaciones se llama "Descriptor" que es una estructura de 8 bytes.

Los 14 bits de más peso del Registro Segmento (RS) conforman el Selector del Segmento que actúa como una entrada en la Tabla de Descriptores, que referencia a los segmentos que maneja la MMU en donde se encuentra la información necesaria para definir al segmento referenciado.

Un descriptor de segmento es una estructura de datos compuesta por ocho bytes, que contiene los parámetros que definen completamente el segmento referenciado (base, límite y derechos de acceso o atributos). Los Descriptores son típicamente creados por compiladores, vínculos, cargadores, o por las operaciones del sistema, pero no aplicaciones de programa.

En la figura 8.8 se muestra el formato que se utiliza en la memoria para contener un descriptor de segmento.

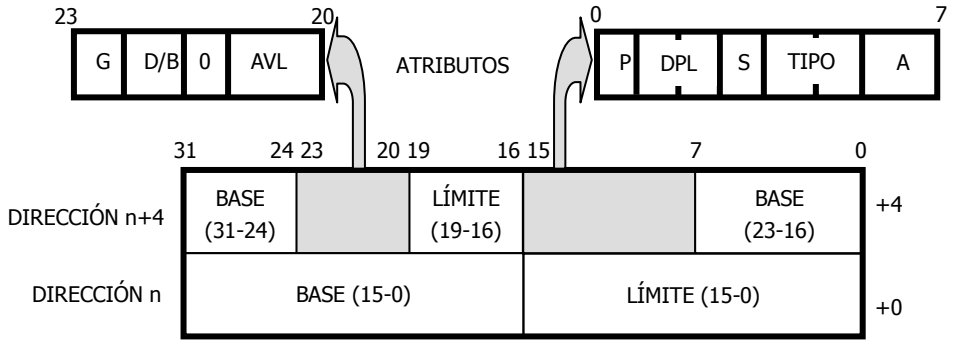


Figura 8.8. Formato de un descriptor de segmento compuesto por ocho bytes, que ocupan dos posiciones de memoria de 32 bits cada una.

Al analizar la estructura del descriptor, se comprueba que en ella residen los tres parámetros que determinan un segmento:

1. **Base**: Es un campo de 32 bits que contiene la dirección lineal donde comienza el segmento.
2. **Límite**: Campo de 20 bits que expresa el tamaño del segmento. Como con 20 bits el tamaño máximo es de 1 MB, hay otro bit complementario en el campo de atributos, llamado de granularidad, G, que indica si el límite está expresado en bytes (G=0) o en páginas de 4 KB (G=1). En este último caso, el tamaño máximo del segmento sería:

$$1\ M \times 4\ KB = 4\ GB$$

3. **Atributo o derechos de acceso**: Se trata de un campo de 12 bits, que proporciona las características relevantes del segmento.

A continuación se describe la función de diversos bits que componen el campo de atributos.

- **Bit de presencia (P):** Indica si el segmento al que referencia el descriptor está cargado, o sea, se halla presente en la memoria principal (P=1), o bien, está ausente (P=0).

  Si P=1, la Unidad de Segmentación accede a la tabla de los Descriptores de Segmento de la MMU y carga la base, el límite y los atributos en el registro caché "invisible" asociado al registro de segmento correspondiente procediendo a continuación al acceso en la memoria principal.

  Si P=0, la Unidad de Segmentación genera una excepción que pone en marcha una rutina del Sistema Operativo encargada de transferir el segmento referenciado desde la memoria virtual a la física. Luego, actualiza el valor del descriptor en la Tabla de Segmentos y da paso a la carga de los parámetros en el registro caché y su posterior acceso.

- **Nivel de privilegio (DPL):** Indica el nivel de privilegio del segmento al que referencia el descriptor. Su valor puede variar entre el 0 y el 3 y es un campo que consta de dos bits.

- **Tipo de segmento (S):** Si S=1, el segmento correspondiente al selector es "normal", o sea, un segmento de código, de datos o de pila.

  Si S=0, se refiere a un segmento del sistema, que referencia un recurso especial del sistema, como puede ser una puerta de llamada, un segmento TSS, etc. Estos segmentos los maneja el programador de sistemas.

- **Tipo:** Los tres bits de este campo distinguen en los normales si se trata de uno de código, de datos o de pila. Además, determinan el acceso permitido: lectura/escritura/ejecución. Posteriormente se estudian los bits de este campo.

- **Accedido (A):** Este bit se pone automáticamente a 1 cada vez que el procesador accede al segmento. Este bit también podría considerarse dentro del Tipo de segmento.

  Muchos S.O. se encargan de borrar periódicamente a este bit y llevar la cuenta del número de veces que es accedido cada segmento, con el fin de implementar adecuadamente los algoritmos que controlan las transferencias entre la memoria virtual y física. En muchos casos, al estar llena la memoria física, hay que establecer qué segmento se ha de eliminar para dejar sitio a uno nuevo. Para esta labor se suele aplicar el algoritmo LRU, que selecciona al segmento que menos veces se haya usado recientemente. Para conocer este dato, el S.O. lleva la cuenta del número de accesos a cada segmento.

- **Granularidad (G):** Los 20 bits del campo LÍMITE del descriptor indican el tamaño del segmento, que estará expresado en bytes si G=0, y en páginas de 4 KB si G=1. En el primer caso, el tamaño máximo del segmento es de 1 MB y en el segundo, de 4 GB.

- **Defecto/grande (D/B):** En los segmentos de código el bit D (Defecto) y en los segmentos de datos de este mismo bit, llamado B (Grande), permiten distinguir los segmentos nativos de 32 bits para el Pentium. Así se mantiene una compatibilidad total con el software creado para el 80286, sin penalizar las nuevas instrucciones

que aporta el Pentium. Si D=1 tanto las direcciones efectivas como los operandos son de 32 bits. Si D=0 se toman sólo 16.

Por ejemplo, la instrucción DEC CX, que decrementa el registro CX, tendrá el mismo código en un segmento perteneciente al 80286, que la instrucción DEC ECX de un segmento del Pentium. Además, la instrucción DEC ECX en un segmento del 80286 y DEC CX en otro del Pentium, estarán precedidas por un prefijo de tamaño byte, que indica que el tamaño del operando es el contrario al tamaño por defecto del tipo de segmento que lo utiliza.

En resumen, con este bit es posible manejar conjuntamente segmentos pertenecientes al 80286 con otros del Pentium.

- **Disponible (AVL):** Este bit está en disposición del usuario para poder diferenciar ciertos segmentos que contengan un tipo determinado de información o que cubran alguna función específica.

Todos los componentes del descriptor son empleados por la Unidad de Segmentación para comprobar si se satisface el conjunto de reglas que protegen a los segmentos, tales como:

1. Si la dirección está dentro del tamaño del segmento.
2. Si el segmento está presente en la memoria.
3. Si el nivel de privilegio del segmento hace posible su acceso.
4. Si se puede leer, escribir o ejecutar.

8.5.2- Tipos de segmentos normales.

Dentro de los segmentos normales, con el bit S =1 en el campo de los atributos, hay dos grandes tipos, cada uno con diversas posibilidades.

- **Segmento de código:**
  1. Sólo ejecutable.
  2. Ejecutable y leíble.
  3. Ajustable o conforming.

- **Segmento de datos:**
  1. Se puede leer y escribir.
  2. Sólo se puede leer.
  3. Es un segmento de pila.

De los tres bits que componen el campo TIPO, dentro de los atributos del descriptor, el bit de más peso E (Ejecutable), diferencia los segmentos de código (E=1), de los segmentos de datos que no se pueden ejecutar (E=0). Figura 8.9

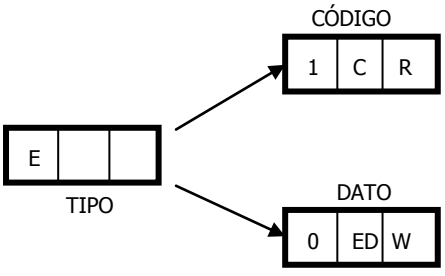


Figura 8.9. El bit E del campo TIPO, distingue entre segmentos de código (E=1) y segmento de datos (E=0). Los otros dos bits toman distinto significado según el tipo de segmento que se trate.

En caso de que E=1, o sea, se referencia a un segmento de código, los otros dos bits de este campo tienen el siguiente cometido:

- **Ajustable o Conforming (C):** Si C=0, al ser accedido el segmento no cambia su nivel de privilegio.

  Si C=1, se llama "segmento ajustable", porque, cuando se accede a él, su nivel de privilegio toma el del segmento que lo ha pedido. Es decir pueden ejecutarse y compartirse por programas con diferentes niveles de privilegio.

- **Leíble (R):** El segmento de código se puede leer si R=1. En ningún caso se puede escribir un segmento de código.

Cuando E=0 y se hace referencia a un segmento de datos, los otros dos bits del campo TIPO tienen el siguiente significado:

- **Expansión decreciente (ED):** Si ED=0, se trata de un segmento de datos normal, datos puros, lo que supone que el crecimiento del mismo se realiza incrementando el valor de la dirección.

  Cuando ED=1, se trata de un segmento de pila pues su crecimiento se efectúa decrementando el valor de la dirección que apunta a su cima, es decir, disminuyendo el valor de la dirección(figura 8.10).

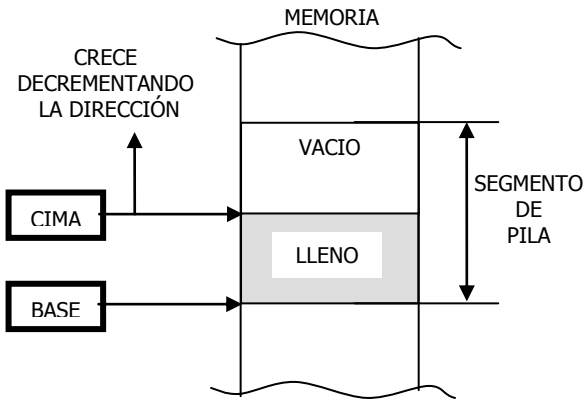


Figura 8.10. Los segmentos de datos con el bit ED = 1, referencian a los segmentos de pila, los cuales crecen disminuyendo el valor de la dirección que apunta su cima.

- **Escribible (W):** Si W=1, el segmento de datos se puede leer y escribir, mientras que, si W=0 sólo se puede leer.

8.6- MANEJO DE LOS DESCRIPTORES

A los descriptores de los segmentos sólo los maneja el procesador automáticamente. Ni el programador de aplicaciones ni el de sistemas tienen acceso a ellos. Los descriptores determinan las características del segmento.

Los descriptores están agrupados en Tablas disponibles en la memoria principal. Cuando en un programa se desea acceder a un nuevo segmento se ejecuta una instrucción. Por ejemplo si se quiere cambiar de segmento de código se ejecuta una " jmp CS':DESPLAZAMIENTO ". Dicha instrucción carga el valor de CS' en el registro de segmento CS. Inmediatamente que se modifica un valor de cualquiera de los seis registros de segmento (CS, SS, DS, ES, FS y GS) el procesador toma como puntero los 14 bits de más peso de dicho registro y direcciona una entrada de la Tabla de descriptores de segmentos. El contenido de dicha entrada que son los 8 bytes de un descriptor los carga en un registro caché ultrarápido de 64 bits asociado al registro de segmento modificado y a partir de ese momento el procesador toma los valores de la Base, Límite y Atributos del descriptor cargado en el registro oculto para direccionar el segmento. En el caso de ser un segmento de código el desplazamiento se carga en el puntero de instrucciones EIP y su valor se suma al de la Base para determinar la siguiente instrucción a ejecutar (ver figura 8.11). Cuando se cambie segmento de código se cambia el de pila en los que el desplazamiento lo determina el valor de ESP.

Como se ve cada registro de segmento tiene:

- Una parte visible, que se puede manejar con una instrucción que puede ser directa (MOV, POP, LDS,…) o implícita, que carga el contenido del registro CS,
- y otra parte invisible que es cargada por el propio procesador y que a la que no se puede acceder .

La carga de los registros de segmento de datos se producen cuando se ejecute en una instrucción un operando ubicado en un nuevo segmento de datos.

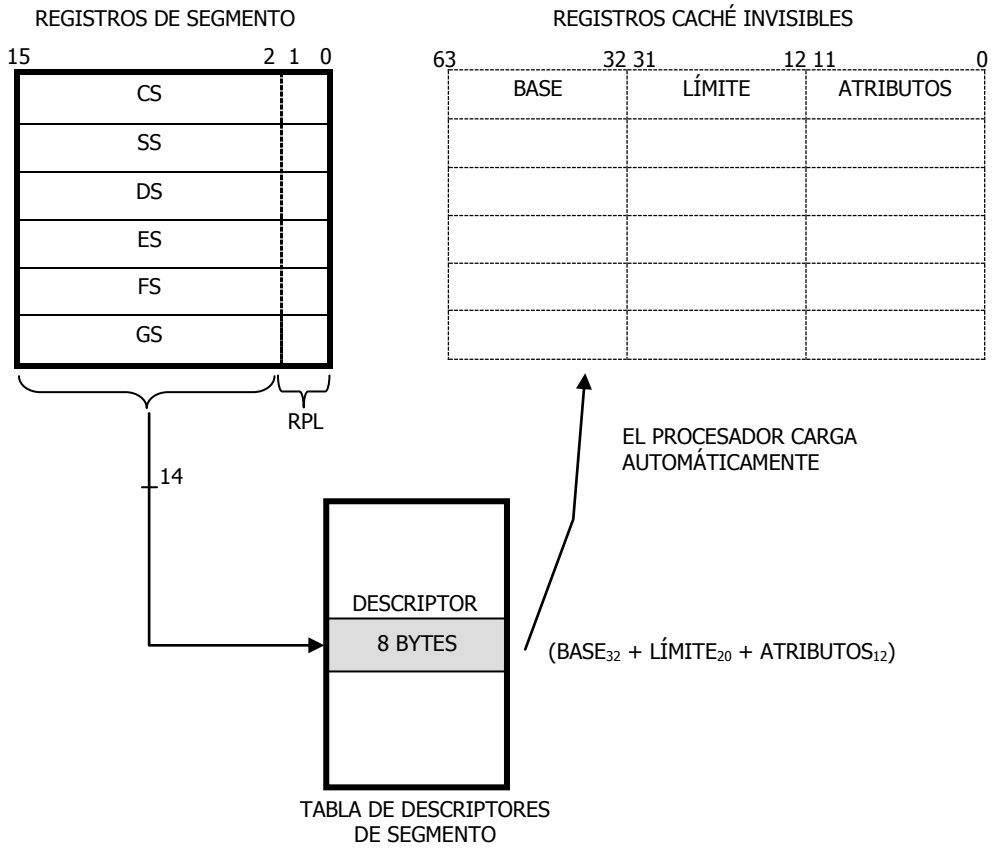


Figura 8.11. Con los 14 bits de más peso del registro de segmento modificado se accede a un descriptor de la Tabla y su contenido (Base, Límite y Atributo) lo carga automáticamente el procesador en el registro caché asociado.

8.7. - TABLAS DE DESCRIPTORES.

Al entrar al Modo Protegido, deben estar residiendo en la memoria principal las tablas de descriptores, que contienen las referencias precisas para los segmentos que va a usar el procesador.

En los Pentium se trabaja en un ambiente multitarea, en el que hay varias tareas atendidas por la CPU, y cada una de estas tareas está formada por segmentos. Un sistema multitarea se compone de un área global, en la que residen los segmentos comunes a todas las tareas y de un área local exclusiva de cada tarea, con los segmentos propios de cada una (figura 8.12).

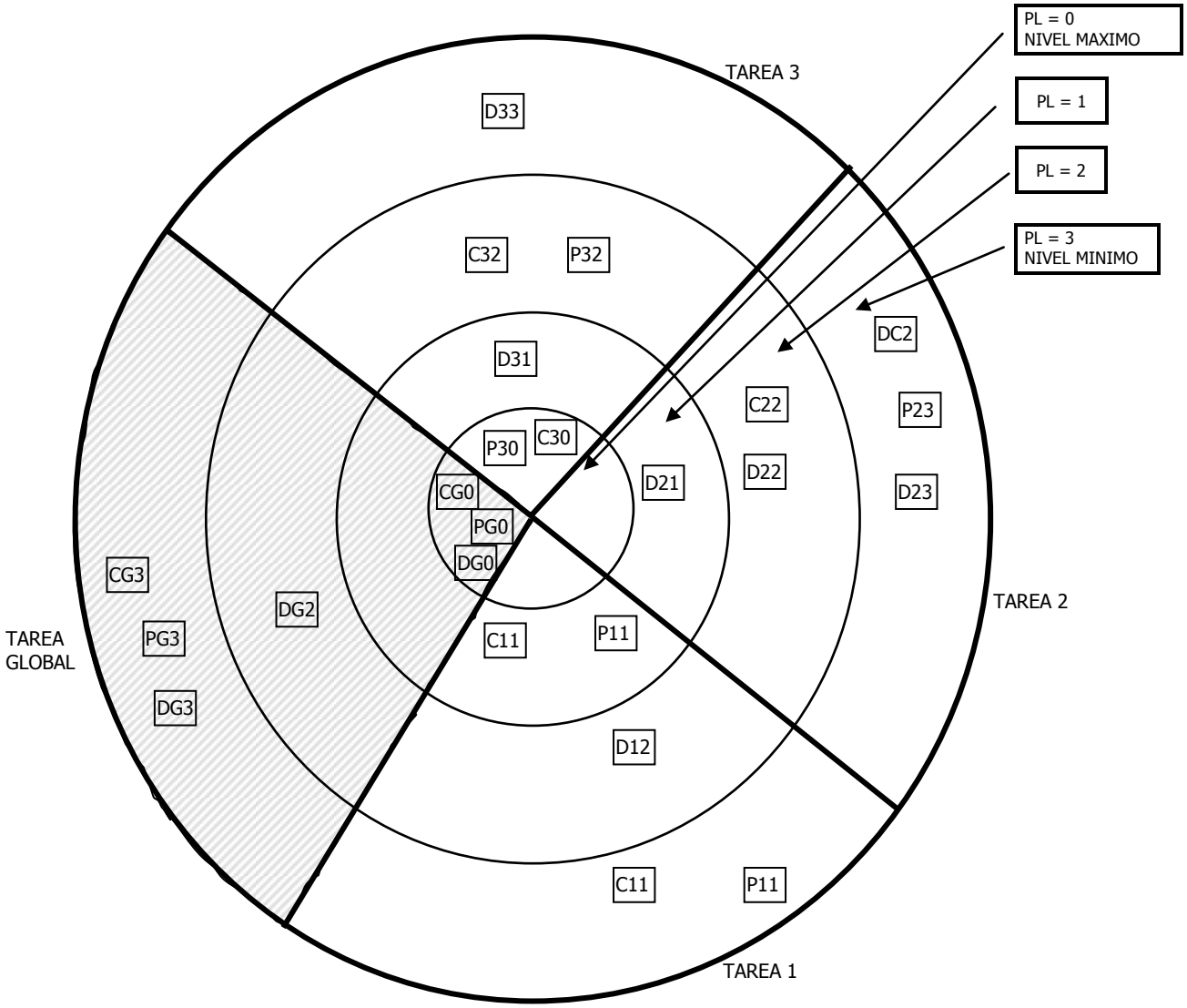


Figura 8.12. Ambiente multitarea en los Pentium.

Cada segmento del área global está definido por un descriptor, existiendo una tabla, llamada Tabla de descriptores globales y también Tabla Global de Descriptores, (GDT), que contiene todos los descriptores del área global.

Asimismo, existe una tabla para cada tarea, que recoge todos los descriptores de los segmentos de cada una de ellas. Se trata de las Tablas de descriptores locales o Tablas Locales de Descriptores, (LDT). Existirán tantas LDT como tareas soporte el sistema.

En un momento determinado el Pentium estará ejecutando una tarea concreta y tendrá activas a la GDT y a la LDT correspondiente a la tarea en curso. Dos registros internos de la CPU, manejados por el programador de sistemas, apuntan a la base de la GDT y a la base de LDT activa, denominándose GDTR y LDTR, respectivamente (figura 8.13).

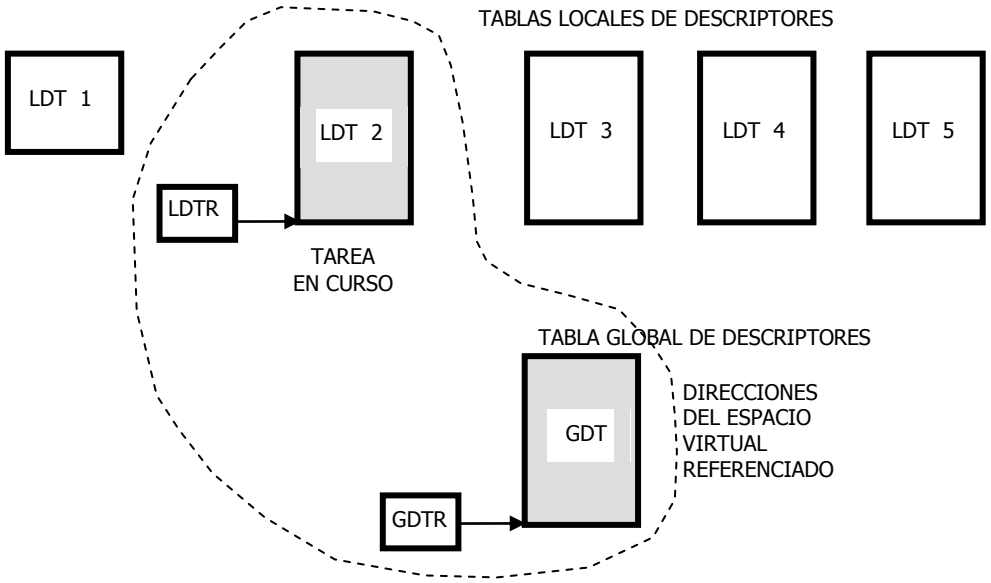


Figura 8.13. En cada instante el Pentium tiene activados a la GDT y a una LDT, la correspondiente a la tarea en curso de procesamiento.

La GDT y la LDT actúan como segmentos del sistema y sólo son accesibles por el sistema de explotación.

La estructura interna de una tabla de descriptores se muestra en la figura 8.14 pudiendo contener un máximo de 8 K descriptores de ocho bytes de tamaño cada una. La LDT es una tabla local propia de la tarea en curso y una conmutación de tarea provoca automáticamente el cambio de la LDT a través de la modificación del valor en el registro LDTR.

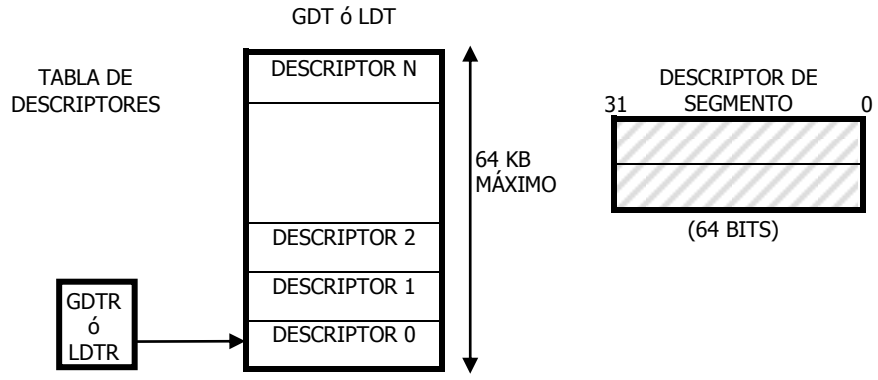


Figura 8.14. Estructura de tablas de descriptores.

Como la CPU tiene activadas dos tablas de descriptores, GDT y LDTn, hay un bit en el selector de segmento del registro de segmento, que indica a cual de ellas se refiere. Es el bit TI: Indicador de Tabla. Cuando TI=1, se selecciona la LDTn, mientras que si TI=0, se hace referencia a la GDT. Los 13 bits más significativos del selector, a los que se denomina ÍNDICE, apuntan una de las entradas a la tabla de descriptores seleccionada con TI. En realidad, el valor de ÍNDICE se multiplica por ocho para apuntar la dirección concreta de inicio del descriptor, puesto que cada uno consta de ocho bytes (figura 8.15). Las Tablas de Descriptores tienen un tamaño máximo de 64 KB y contiene un máximo de 8 K descriptores.

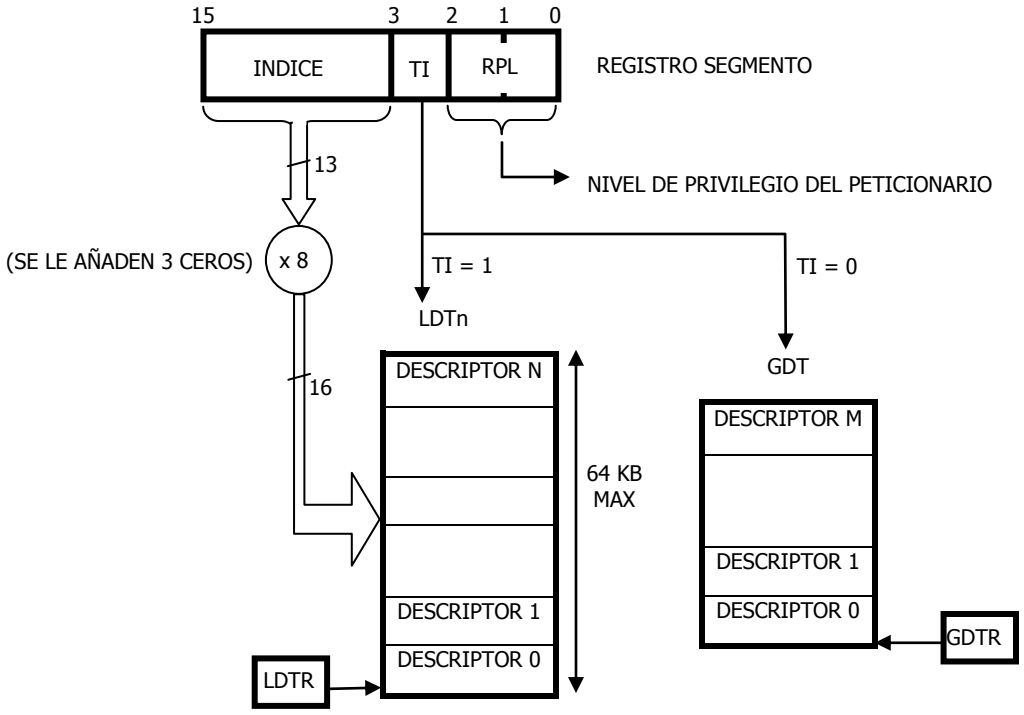


Figura 8.15. El bit TI selecciona la tabla de descriptores, mientras que el campo INDICE, multiplicado por 8, la dirección del descriptor seleccionado.

El Pentium no usa al descriptor de la entrada 0 de la GDT, que es un selector nulo, o sea, con todos los bits a 0 (INDICE =0 y TI =0). Esta circunstancia posibilita que el sistema cargue en cualquier registro de segmento un selector nulo sin que se origine una condición de excepción. De esta forma se puede borrar el contenido de un registro de segmento cargándolo con ceros.

El descriptor 0 de las LDT puede usarse, puesto que no responde a un selector nulo, ya que el bit TI=1.

A partir del selector y a través de las tablas de descriptores, la Unidad de Segmentación localiza la base del segmento a la que suma el desplazamiento para obtener la dirección lineal, que se convierte en la dirección física cuando se halla desactivada la Unidad de Paginación (figura 8.16).

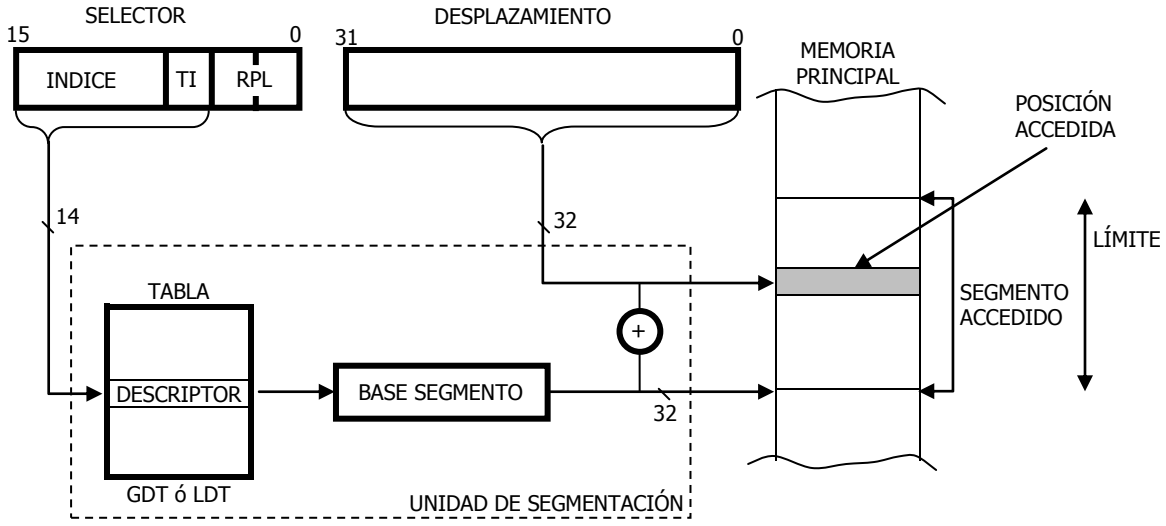


Figura 8.16. Obtención de la dirección lineal o física por parte de la Unidad de Segmentación, cuando está inhibida la Paginación

Como desplazamiento del segmento de código se utiliza EIP. ESP actúa como desplazamiento del segmento de pila. Cuando se accede a datos y se emplea como selector a uno de los registros DS, ES, FS ó GS el desplazamiento viene dado por la definición del operando y su modo de direccionamiento dentro de la instrucción a la que pertenece. En el caso más complejo, este desplazamiento puede venir formado por cuatro parámetros:

1. Un registro base.
2. Un registro desplazamiento de hasta 32 bits, incluido en la propia instrucción.
3. Un registro índice.
4. Un factor de escala (x1, x2, x4 o x8), según el número de bytes del operando.

El organigrama de la figura 8.17 muestra la obtención de la dirección física correspondiente a una posición de un segmento de datos, con un modo de direccionamiento en el que participan un registro base, un desplazamiento, un registro índice y un factor de escala.

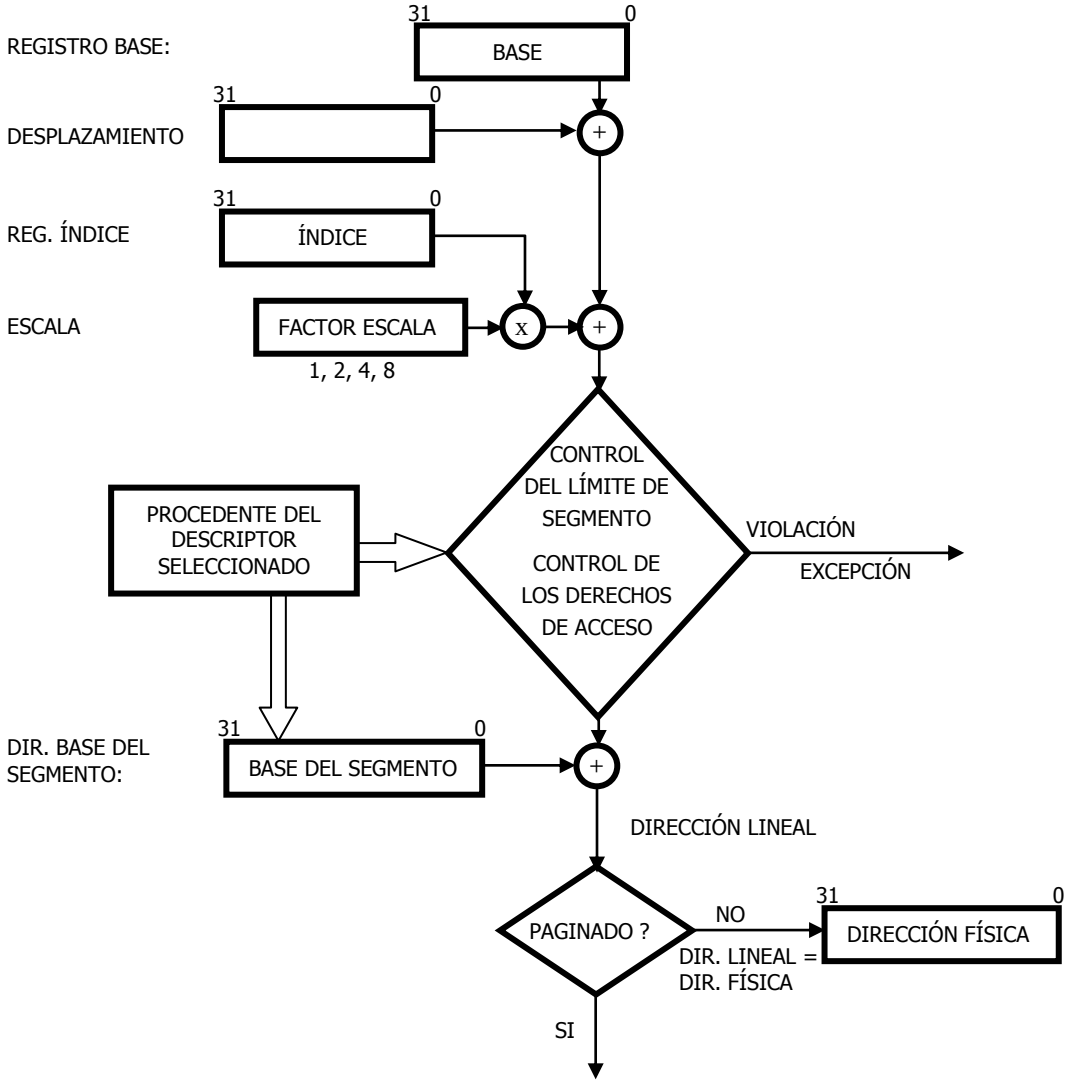


Figura 8.17. Procedimiento por el que se obtiene la dirección física de una posición en un segmento de datos, definida por un direccionamiento complejo.

8.8- EL MODELO PLANO

El mecanismo de segmentación es intrínseco al Pentium y no puede desactivarse. En aquellas aplicaciones y sistemas en los que no se use la segmentación hay un procedimiento para simular su inhabilitación, llamándose plano al modelo de memoria al que se tiene acceso en esta situación.

Se cargan todos los registros de segmento con selectores que apuntan en las tablas a descriptores caracterizados porque el valor de su base es 00000000H y el límite FFFFFFFFH. De esta manera, la CPU sólo maneja un único segmento, que abarca todo el espacio lineal, o sea, los 4 GB (figura 8.18). No alterando el valor de los registros de segmento, el procesador maneja seis segmentos que se solapan en el espacio lineal, en base a los diferentes desplazamientos, que puesto que admiten 32 bits, pueden alcanzar los 4 GB.

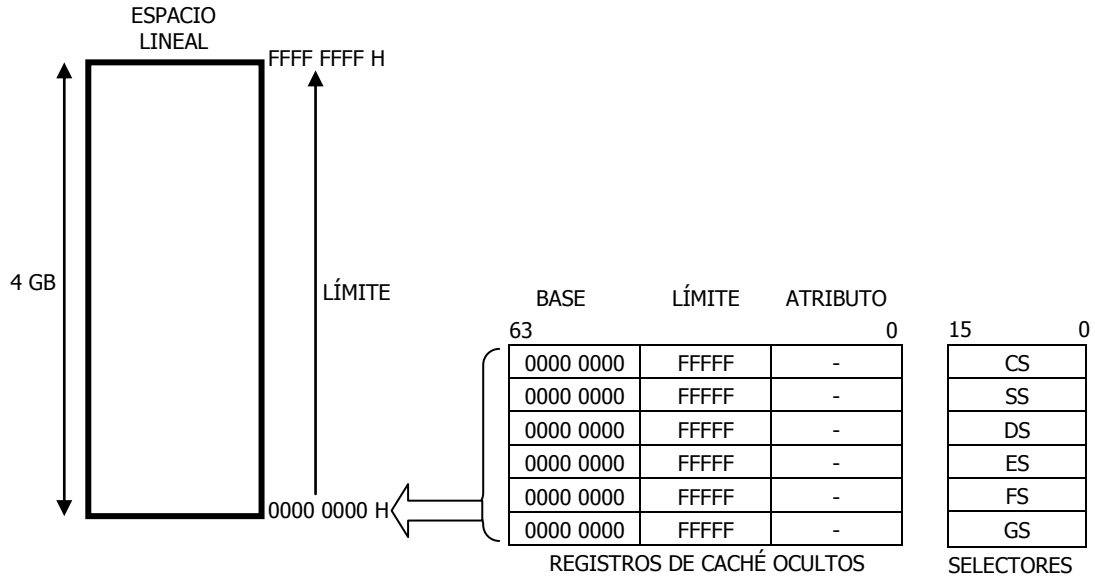


Figura 8.18. En el modelo "plano" el procesador maneja un único segmento en el que se solapan los segmentos de código, datos y pila, mediante los desplazamientos de 32 bits correspondientes a cada uno.

Usando el modelo plano se puede activar la Paginación con objeto de conseguir un entorno protegido a nivel de páginas.