EL MECANISMO DE PAGINACIÓN 9

9.1. INTRODUCCION

La paginación es un procedimiento de gestión de la memoria muy eficaz en los sistemas operativos multitarea que manejan memoria virtual. Divide y manipula los programas y los datos en trozos de tamaño fijo, llamados páginas.

A diferencia con los segmentos, las páginas no guardan relación con la estructura lógica con la que se ha construido el software.

La mayor ventaja de la paginación se obtiene en la transferencia e intercambio de elementos entre la memoria virtual y la física. El hecho de que las páginas tengan siempre el mismo tamaño facilita la ocupación de la memoria así como el rendimiento en su explotación.

Esto implica que los sistemas operativos que manejan la paginación sean muy simples, sencillos y rápidos debido a que los algoritmos de transferencia son muy simples ya que mueven cantidades de datos de igual tamaño.

El mayor inconveniente que presenta la paginación es el mal aprovechamiento de la memoria, puesto que las páginas tienen un tamaño fijo y puede que un objeto de código sea muy grande y requiere varias páginas o que un objeto de los datos ocupe menos espacio que una página, por lo tanto la paginación no se adapta a las necesidades de espacio que precisa el software.

Por otra parte, y puesto que la mayoría de los sistemas lógicos basan su flujo de procesamiento en el principio de vecindad, sólo es necesario que un reducido número de páginas de la tarea en curso esté ubicado en la memoria principal en cada momento.

El procesador Pentium siempre trabaja con segmentación y optativamente puede trabajar además con paginación.

9.2. MECANISMO DE PAGINACION

La Unidad de Paginación esta implantada en hardware dentro del Pentium.

El funcionamiento de la paginación es optativo y para su habilitación basta con poner a 1 un bit (PG) de uno de los registros de control (CR0), de los que maneja el programador de sistemas, para ello se utiliza la instrucción: MOV CR0, FFFF. Como a dicho bit solo se le puede modificar en Modo Protegido, la paginación solo opera en dicho modo.

Cuando está habilitada la paginación, se divide a cada segmento del espacio lineal creado por la Unidad de Segmentación, en páginas sucesivas de 4 KB de tamaño cada una. El Pentium también puede manejar páginas de hasta 4 MB, Luego, la Unidad de Paginación carga y distribuye, de forma aleatoria, las páginas que se precisan en cada momento, sobre el espacio de la memoria física.

Los algoritmos usados en la transferencia de bloques desde/hacia la memoria principal, son mucho más sencillos y efectivos que en la segmentación, puesto que manipular bloques de tamaño fijo y reducido, optimiza el aprovechamiento del espacio de memoria. Además, puesto que el Pentium dispone de una memoria caché de alta velocidad para guardar la traducción de direcciones lineales a físicas, dentro de la paginación, se puede acelerar extraordinariamente la velocidad de acceso.

En los procesadores Intel386 e Intel486 era necesario ejecutar una instrucción JMP después de activar la paginación para que no se ejecutaran las instrucciones que previamente ya habían sido buscadas y decodificadas. El Pentium utiliza un buffer (BTB) de predicción de bifurcaciones que evita la necesidad de utilizar instrucciones para eliminar las instrucciones que ya han sido buscadas y decodificadas como ocurría en los procesadores anteriores.

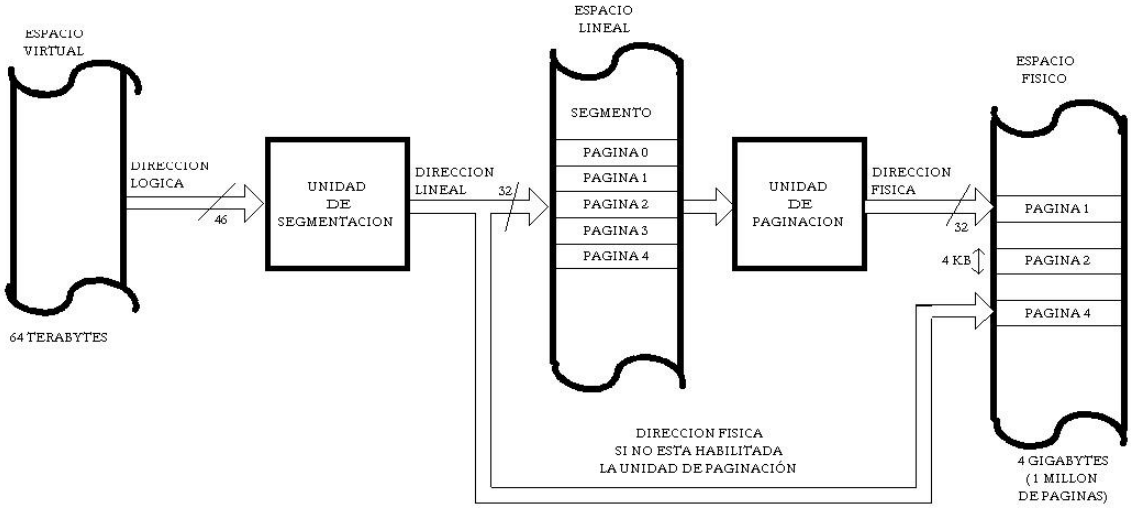

Figura 9.1 – Cuando funciona la Unidad de Paginación, los segmentos del espacio lineal se dividen en páginas y se trasladan al espacio físico solo aquellas páginas que se precisan en cada momento

De la figura 9.1 se deduce que la Unidad de Paginación traduce la dirección lineal a física. Teniendo en cuenta el espacio físico que puede alcanzar un máximo de 4 GB, la paginación lo descompone en un millón de páginas de 4KB, o sea, actúa como una gran tabla de un millón de entradas, una por página. En cada entrada se guardaría la dirección base de comienzo de la página y los atributos de la misma, algo similar a un descriptor de segmento, pero sin el campo LIMITE.

Para referenciar la base de la página bastan 20 bits, puesto que, como cada página tiene 4 KB, los 12 bits menos significativos de la dirección de 32 bits, serán siempre 0. De esta forma el descriptor de página proporciona los 20 bits de más peso de la dirección de la base, a los cuales se añaden los 12 bits de menos peso, que serán 0 si se referencia el comienzo y otro valor cuando se intenta acceder a una posición cualquiera de dicha página.

Con esta organización de la memoria, la Unidad de Paginación manejaría una tabla con un millón de entradas, conteniendo cada una la base (20 bits) y los derechos de acceso (12 bits) de cada página de la memoria principal. Luego si cada entrada consta de 32 bits de Tabla de Páginas tendría un tamaño de 1 M x 4 bytes = 4 MB (figura 9.2).

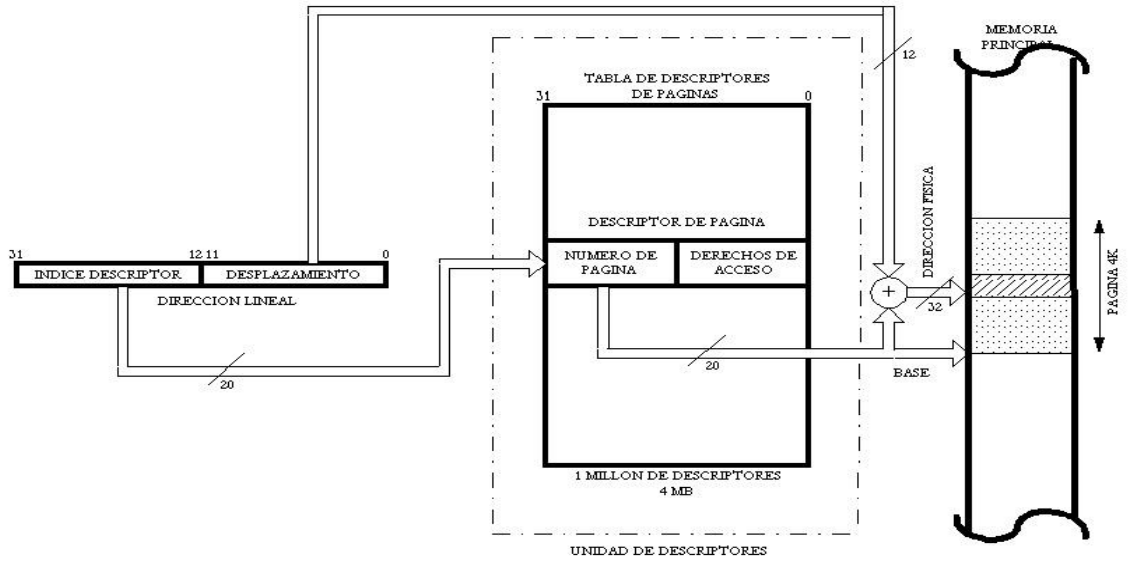

Figura 9.2 – "Aparentemente", la Unidad de Paginación se comporta como una tabla con un millón de entradas, que contienen los descriptores de las páginas en la memoria principal. La tabla traduce la dirección lineal en física.

Cada vez que la Unidad de Paginación detecta que la página no reside en la memoria principal, genera un fallo de pagina, que origina una excepción que llama a una rutina del S.O. que se encarga de trasladar dicha página desde la memoria virtual a disco hasta la memoria física o RAM.

Como la Tabla de descriptores de Páginas tiene un tamaño de 4 MB y debe residir en la memoria principal para poder ser manejada por la CPU, puesta en marcha del mecanismo de paginación hipotecaría un amplio espacio de memoria física. Para solucionar este problema, en vez de emplear una sola tabla con un millón de entradas, INTEL introdujo una traducción de direcciones a dos niveles, pero ahora se pueden producir dos fallos de página, uno por cada tabla.

En cada tarea, un primer nivel de traducción lo soporta la tabla denominada DIRECTORIO DE TABLAS DE PAGINAS, que consta de 1 K entradas de 32 bits cada una, es decir, ocupa una página de 4 KB. Esta tabla o directorio es fija para cada tarea y su base está cargada en el registro de control de la CPU, designado como CR3. Para posibilitar la paginación es imprescindible que el Directorio de Tabla de Páginas esté ubicado en la memoria principal.

El acceso a una entrada del Directorio de Tablas de Páginas se calcula sumando a la base (CR3) el valor de los 10 bits de más peso de la dirección lineal. La entrada seleccionada del Directorio contiene la dirección de la base de una página, que actúa como una segunda Tabla de Páginas, formada por 1 K entradas de 32 bits cada una. Se accede a una de estas entradas sumando a la dirección base de la Tabla de Páginas los 10 bits centrales de la dirección lineal. En la entrada seleccionada se almacena la dirección de la base de la página a acceder junto a sus atributos y para elegir, dentro de ella, la posición concreta, se suma a la base el valor de los 12 bits de menos peso de la dirección lineal. Véase la figura 9.3.

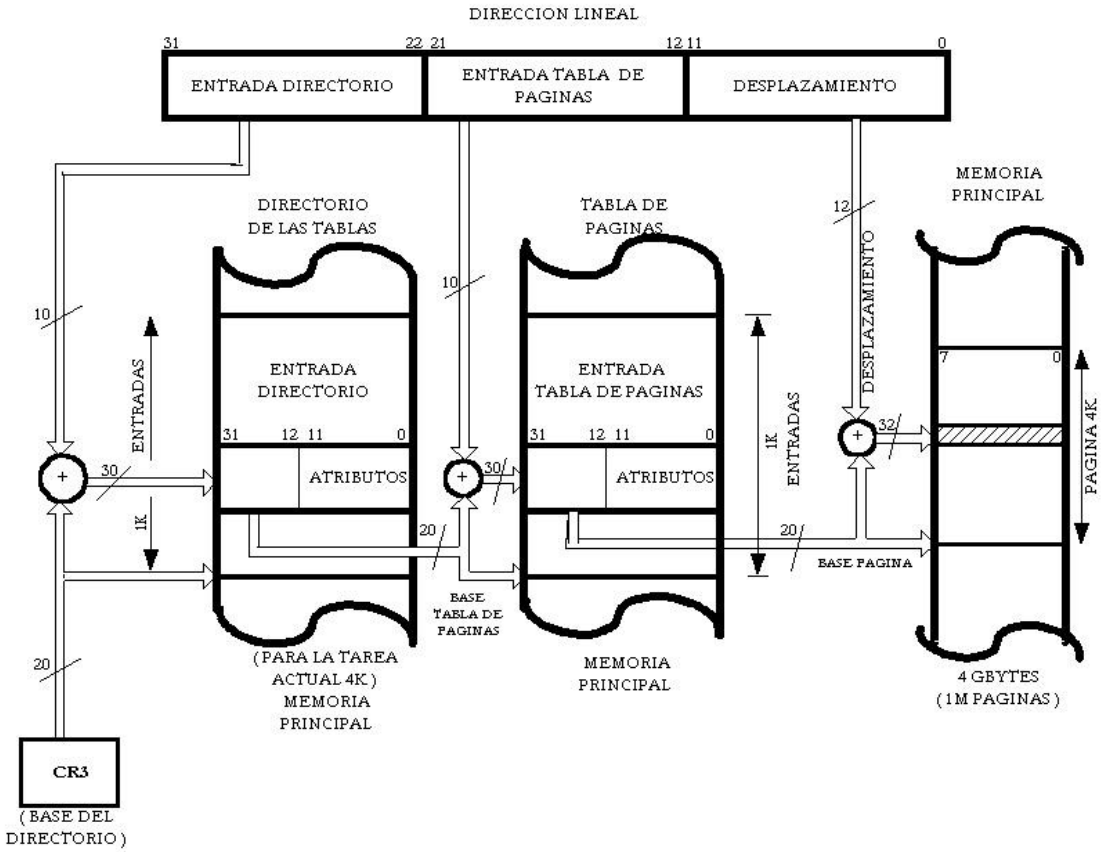

Figura 9.3 – Esquema sobre el mecanismo de paginación. El directorio de las Tablas de Páginas es propio de ada tarea y su base es apuntada por el registro de control CR3, el cual se modifica en cada conmutación de tarea.

Apréciese que, para seleccionar entradas en el Directorio de Tablas de Páginas y en las Tablas de Páginas de segundo nivel, solo se usan 10 bits, para cada caso, aunque el tamaño de ambos elementos es el de una página de 4 KB. La razón estriba en que cada entrada posee cuatro bytes y, por tanto, los dos bits de menos peso de la dirección de cada entrada serán 0, lo que equivale a multiplicar por cuatro el valor de 10 bits usado en la selección de la entrada (figura 9.4).

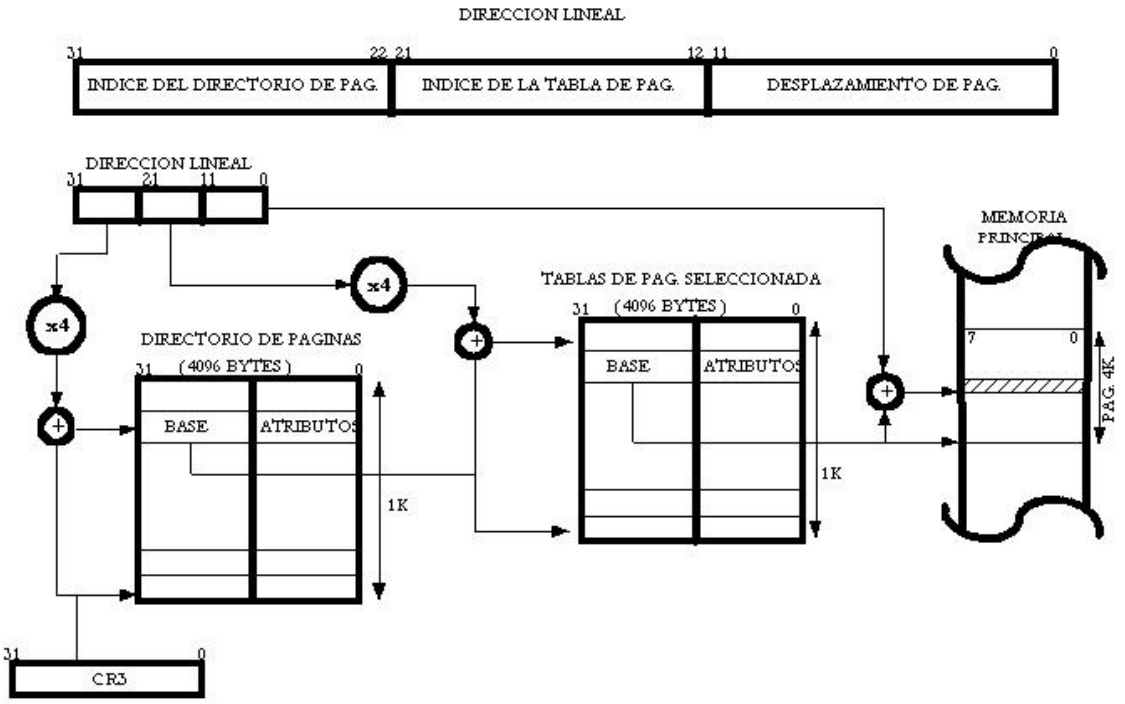

Figura 9.4 – Traducción de una dirección lineal a física.

En el caso de referenciar la dirección lineal una página de 4MB, cosa posible en el Pentium, los 10 bits de más peso direccional el directorio de páginas y los restantes 22 actúan como desplazamiento en la página.

Como las anteriores alternativas planteadas son lentas y degradan el rendimiento del computador, INTEL introdujo en el hardware, dentro del propio chip del procesador Pentium, una caché ultrarrápida de acceso por contenido (CAM) denominado TLB para optimizar el rendimiento del sistema, y que será explicado en los siguientes apartados.

Otra posible solución para el mecanismo de paginación es la Tabla de Páginas Inversa, que es utilizada por los equipos AS/400 de IBM y los RISC PowerPC (figura 9.5).

Esta técnica consiste, en que, en lugar de una entrada por cada página virtual tiene una entrada por cada marco de páginas de la memoria física asignada al proceso. Es decir la tabla inversa solo da información de las páginas guardadas en los marcos. Luego si se produce un fallo debe existir un mecanismo de traducción de la información a memoria secundaria.

Se utiliza una tabla hash para implementar la tabla. El número de página virtual se mapea a esta tabla mediante una función hashing sencilla. Recordamos que una función hash hace corresponder números que se encuentran en un rango 0-m a números que se encuentran en el rango 0-n siendo m>n. La salida de esta función se utiliza como índice de la tabla hash.

Es posible que más de una entrada apunte a la misma posición de la memoria principal. Cuando una posición está ocupada se crea un puntero a otra posición. Las cadenas que se crean con las técnicas hashing son habitualmente cortas. En estos casos para saber qué marco contiene una página hay que recorrer todos los enlaces hasta el final. Con lo que se demuestra que las cadenas que se forman son pequeñas. La tabla hash o tabla inversa, contiene la siguiente información:

- Página: indica la página almacenada en esa dirección.

- ETP: entrada da tabla de página que contiene el número de marco. Esta entrada de tabla es equivalente a la ya vista y contiene información similar.
- Puntero: a otra dirección de la tabla.

En las tablas de páginas normales se tienen tantas entradas como páginas virtuales luego pueden crecer enormemente. En la tabla de páginas inversa hay tantas entradas como marcos de página asignados al proceso el sistema operativo. Su principal ventaja es que el tamaño de la tabla es constante, independientemente del tamaño del proceso. Como inconveniente tiene que la paginación es un poco más compleja pues hay que aplicar previamente la función hash.

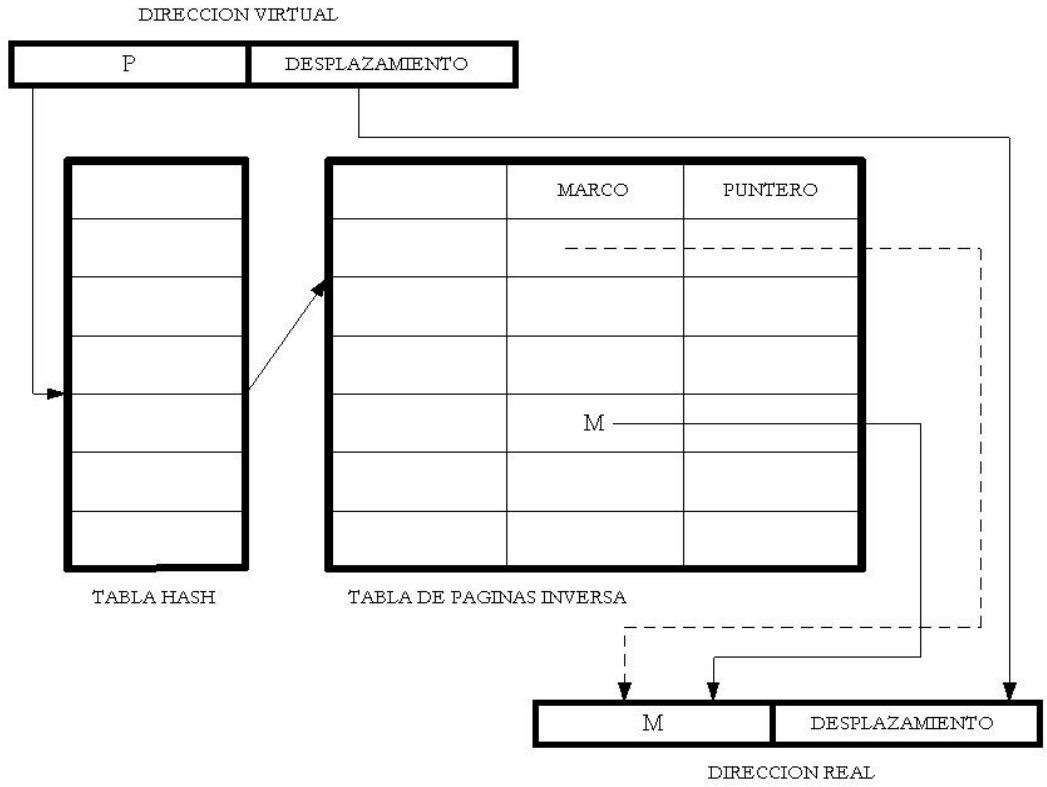

Figura 9.5. Paginación mediante la Tabla de Páginas Inversa.

9.3. FORMATO DE LAS ENTRADAS AL DIRECTORIO Y A LAS TABLAS DE PÁGINAS

El formato de una entrada al Directorio es similar al de una entrada a una Tabla de Páginas. Ambas constan de 32 bits de los cuales los 20 de más peso proporcionan los 20 bits más significativos de la dirección de la base de la página de la siguiente estructura a la que hacen referencia. Los 12 bits de menos peso de la dirección de la base son ceros y por eso en la entrada se utilizan para definir los atributos.

En cuanto a los 12 bits restantes de las entradas que se comentan, sirven para definir los atributos tal como se refleja en la figura 9.6.

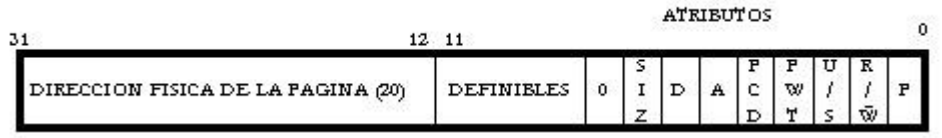

Figura 9.6. Estructura de una entrada del Directorio o de la Tabla de Páginas. Hay tres bits a disposición del S.O. que pueden usarse para guardar información auxiliar sobre la página.

A continuación se describe la función de los bits significativos del campo de los atributos de una entrada del Directorio o de las Tablas de Páginas.

- **P** (BIT DE PRESENCIA): Es el bit de menos peso de una entrada y, si está a 1, indica que la página está cargada en la memoria física y, en consecuencia, los restantes bits de dicha entrada son operativos.
  Si P = 0, significa que la página no está cargada en la memoria física. La CPU genera una excepción de fallo de página, que activa una rutina del S.O. que trae la página desde la unidad de almacenamiento externo (disco) a la RAM. Una vez cargada la página en la memoria física, el S.O. pone P = 1, escribe los 20 bits de más peso de la entrada con el valor correspondiente a la dirección de la base de la página y actualiza los restantes bits de los atributos. Posteriormente, la CPU vuelve a acceder a la entrada y como P = 1, procede a la lectura o escritura de la posición accedida.
- **A** (BIT ACCEDIDO): Se pone a 1 cada vez que se accede a dicha página. Este bit lo maneja el S.O. para llevar la cuenta del número de accesos que tiene cada página. Cada poco tiempo el S.O. lee este bit y, si vale 1, lo pasa a 0 e incrementa el contador que tiene asociada la página. El número de accesos es empleado por el algoritmo LRU, que sirve para eliminar de la memoria la página que menos se haya usado recientemente.
- **SIZ**: Este bit sólo existe en CR3 y si esta activado el Pentium trabajará con páginas de tamaño 4MB. En las entradas del directorio de p´sginas y de las tablas de páginas este bit vale 0 ya que no tiene ningún valor. Los anteriores procesadores al Pentium sólo podían manejar páginas de 4 KB.
- **D** (BIT SUCIO): Indica si se ha escrito en la página. Si D = 1, significa que se ha escrito y, cuando se quiera eliminar esta página de la memoria principal, será preciso salvarla previamente en la memoria virtual para mantenerla actualizada.
  Si D = 0, se puede sobrescribir en esta página cuando se decide sustituirla por otra, ya que no ha sido modificada durante su estancia en la memoria principal.

- **R/W** (BIT DE LECTURA/ESCRITURA): Si R/W = 1, la página es accesible en lectura y escritura, mientras que si vale cero, sólo se puede leer.
- **U/S** (BIT USUARIO/SUPERVISOR): Indica el nivel de privilegio correspondiente a dicha página. La paginación también aplica reglas de protección, aunque menos potentes al existir solo dos niveles de privilegio: Nivel Supervisor, equivalente al nivel 0 de la segmentación y Nivel Usuario, equivalente al nivel 3.

  Si U/S = 1, la página tiene nivel Supervisor y, por tanto, el mayor grado de confianza y seguridad. En dichas páginas pueden existir todo tipo de instrucciones, mientras que las páginas con el nivel de Usuario (U/S = 0) tienen ciertas restricciones.
  Al activarse la paginación, cada acceso a memoria soporta tres controles por parte del sistema de protección:

  1. A nivel de descriptor de segmento.
  2. A nivel de entrada al Directorio.
  3. A nivel de entrada a la Tabla de Páginas.

  El nivel resultante en la paginación se considera como el más restrictivo entre el correspondiente a la entrada del Directorio y el de la entrada de la Tabla de Páginas.

- **PCD** (BIT DE ACTIVACION DE LA CACHE): Indica si la página es cacheable, si se puede meter o no en la memoria caché
- **PWT** (BIT DE ESCRITURA OBLIGADA): Indica que la página además de ser cacheable funciona en modo de Escritura Obligada.
- **Definibles**: Son tres bits a disposición del S.O. que pueden usarse para guardar información auxiliar sobre la página.

Cualquier fallo en la protección que detecta la Unidad de Paginación origina una excepción, que, entre otras cosas, guarda el valor de la dirección lineal que lo ha provocado en uno de los registros de control (CR2).

9.4. TABLA DE TRADUCCION DE DIRECCIONES LINEALES (TLB)

El mecanismo de traducción de direcciones en la paginación es bastante lento porque debe realizar dos accesos a memoria, uno al Directorio y otro a la Tabla de Páginas, para localizar la página y, en un tercer acceso, efectuar la lectura o escritura solicitada por la CPU. Además, en dichos accesos se debe realizar la suma del valor de la base y el desplazamiento en la página referenciada, con lo cual la CPU tarda cierto tiempo en realizar la traducción de una dirección lineal a física.

Para acelerar el proceso de traducción, INTEL, introdujo dentro del propio chip una pequeña memoria caché ultrarrápida, que se denomina TLB: "TRANSLATION LOOKSI DE BUFFER".

La TLB guarda la traducción de las direcciones lineales a físicas, correspondientes a las 32 últimas páginas que se han manejado. Se trata de un mecanismo parecido al empleado en la segmentación, donde a cada registro segmento se asociaba un registro caché invisible, que guardaba el valor del descriptor con el fin de evitar tener que acceder a las tablas de descriptores en cada acceso a memoria.

Con la paginación en marcha, la CPU consulta a la TLB en cada acceso a memoria y, en el caso de que la página referenciada se halle almacenada junto a su traducción (de dirección lineal a física), se tardan un tiempo despreciable en obtener la dirección física. Si la página no esta en la

TLB, el mecanismo de la paginación accede al Directorio y, luego, a la Tabla de Páginas adecuada, cargando el valor de la dirección física hallada en la TLB. Después, la CPU vuelve a buscar en la TLB para efectuar el acceso, con lo que se originará un retardo suplementario, que no es mucho si se compara con el ahorro que se logra si, como es de esperar, la mayor parte de los accesos se encuentran en la TLB al seguir las reglas de la vecindad de la construcción del software.

Cada vez que la dirección lineal está contenida y traducida en la TLB, se obtiene la dirección física en escasos nanosegundos.

9.5. ESTRUCTURA Y FUNCIONAMIENTO DE LA TLB

Con las 32 entradas residentes en la TLB se controlan 32 páginas, que suponen un espacio total de 128 KB de memoria, que, en muchas ocasiones, es suficiente para contener el área de trabajo de un proceso. Se ha comprobado experimentalmente que, para programas de proposito general, una TLB de 32 entradas proporciona "ACIERTO" en más del 97% de los accesos a la memoria. También hay que considerar que en cada cambio de contexto, hay que limpiar la TLB, lo que puede ser barato, pero hay que considerar un costo indirecto, pues si los cambios de contexto son muy frecuentes, la tasa de aciertos se puede reducir.

Las 32 entradas de la TLB están organizadas en cuatro grupos de ocho entradas cada uno, que operan en paralelo. La gran velocidad en este tipo de memorias se alcanza por su método de acceso, que es por contenido (CAM: Memoria de Acceso por Contenido).

Cada entrada en una CAM se compone de una etiqueta y un dato asociado. Cuando se quiere obtener una información se suministra un valor, que se compara con los campos de etiqueta de todas las posiciones. La comparación se hace en paralelo y a gran velocidad. En el caso de que el circuito comparador hardware detecte ACIERTO o PRESENCIA, es decir, que el valor suministrado a la CAM coincida con alguna etiqueta, la información asociada a la misma se obtiene como salida. En la TLB, se proporciona como entrada la dirección lineal y, si la contiene, el resultado es la dirección física correspondiente.

Si el comparador no encuentra una etiqueta igual a la información suministrada, la CAM no contiene la traducción que se busca y señaliza AUSENCIA o FALLO.

Como la TLB está estructurada en cuatro grupos de ocho entradas cada uno, la comparación con el campo etiqueta se hace de la siguiente forma: Con los bits 12, 13 y 14 de la dirección lineal a traducir, se selecciona una de las ocho entradas en los cuatro grupos, simultáneamente. Dichas entradas o etiquetas constan de 20 bits: los 16 bits de más peso de la dirección lineal, un bit de VALIDEZ y tres más de atributos (D: Sucio, U: Usuario y W: Escritura). Después, mediante cuatro comparadores que trabajan en paralelo, se comparan las cuatro entradas o etiquetas seleccionadas con los bits 15 al 31 de la dirección lineal. En caso de que algún comparador detecte PRESENCIA (HIT), significa que la dirección lineal está traducida en la TLB. Entonces la CPU procede a extraer la información ligada con la etiqueta del acierto y saca los 20 bits de más peso de la dirección física que carga en la líneas 12-31 del bus de direcciones. Añadiendo los 12 bits de menos peso de la dirección lineal a las líneas 0-11 del bus de direcciones, se obtiene la dirección física completa.

Si los circuitos comparadores señalan AUSENCIA (MISS), se pone en marcha el mecanismo de paginación y, tras acceder al Directorio y a una Tabla de Páginas, se obtiene los 20 bits de más peso de la dirección física, que se cargan en una de las entradas da la TLB. Después, la CPU reintenta el acceso por la TLB, que dará ACIERTO, procediendo a su acceso (figura 9.6).

Cada vez que se modifican las Tablas de Páginas al cambiar de Directorio como consecuencia de modificarse el contenido del registro CR3. Hay que borrar la TLB, puesto que el 386 no hace esta operación automáticamente. El procesador asume la traducción de dirección lineal a física, sin considerar las conmutaciones de tarea. El sistema de explotación debe encargarse de inicializar las tablas de paginación, soportar las rutinas ante fallos de páginas y reinicializar la TLB después de toda modificación de las tablas de páginas.

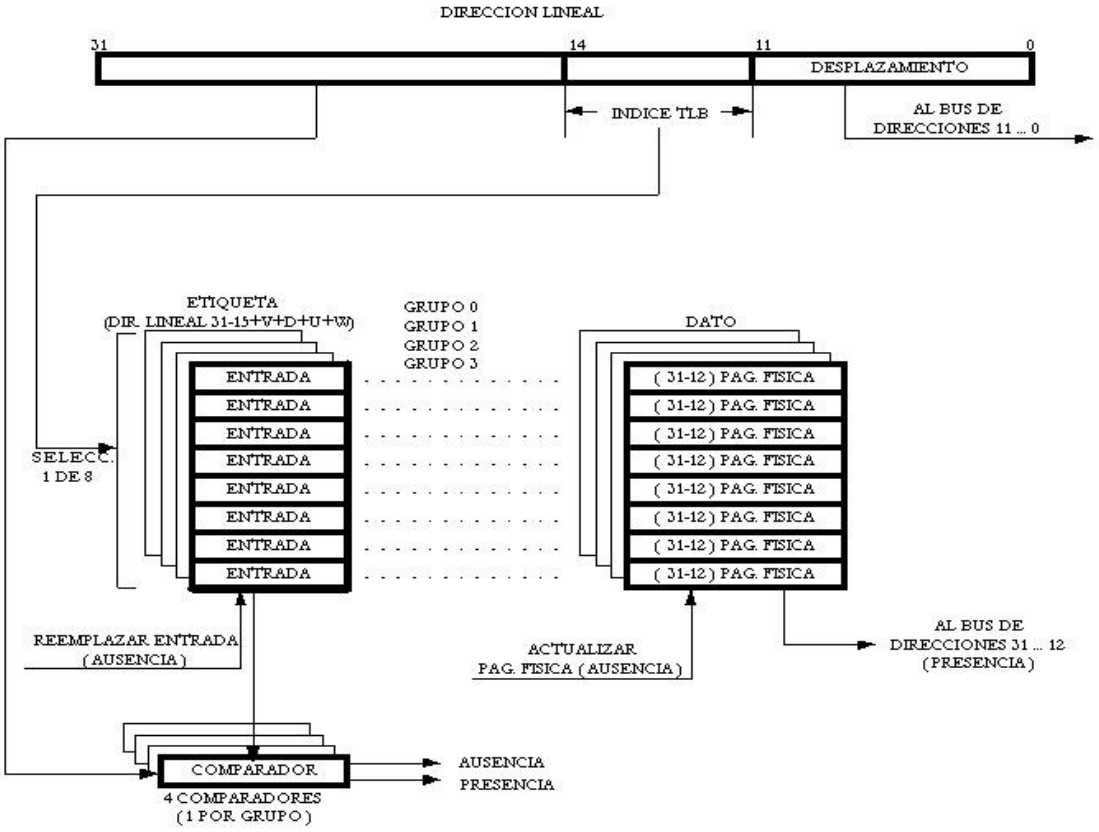

Figura 9.7 – Organización y esquema simplificado sobre el comportamiento de la TLB.

El Pentium tiene dos TLB independientes, una para la caché de instrucciones y otra para la de datos. La caché de datos tiene 8 entradas exclusivas para páginas de 4MB y 64 para las de 4 KB.

Las TLB son invisibles para todos los programas con excepción de los del Sistema Operativo con nivel de prioridad PL = 0.

El programador del Sistema Operativo debe invalidar las tablas de TLB en cuanto se produzcan cambios en las entradas de las mismas. Para realizar dicha invalidación puede usar una instrucción MOV para cargar CR3 o provocar una conmutación de tareas. Esto se debe a que el programador del sistema operativo posee dos registros de prueba de la TLB (TR6 y TR7, que se explicarán en los siguientes capítulos), con los que puede leer y escribir el contenido de una entrada de la TLB.

Existe una instrucción en el Pentium, INVLPG, que permite invalidar una entrada concreta de la TLB, generando una dirección virtual a partir del operando dado e invalidando la correspondiente entrada de la caché de la tabla de páginas, la TLB.