

que hace mucho que MULTICS desapareció, su espíritu sigue vivo porque la memoria virtual de todas las CPU de Intel a partir del 386 ha seguido de cerca su modelo.

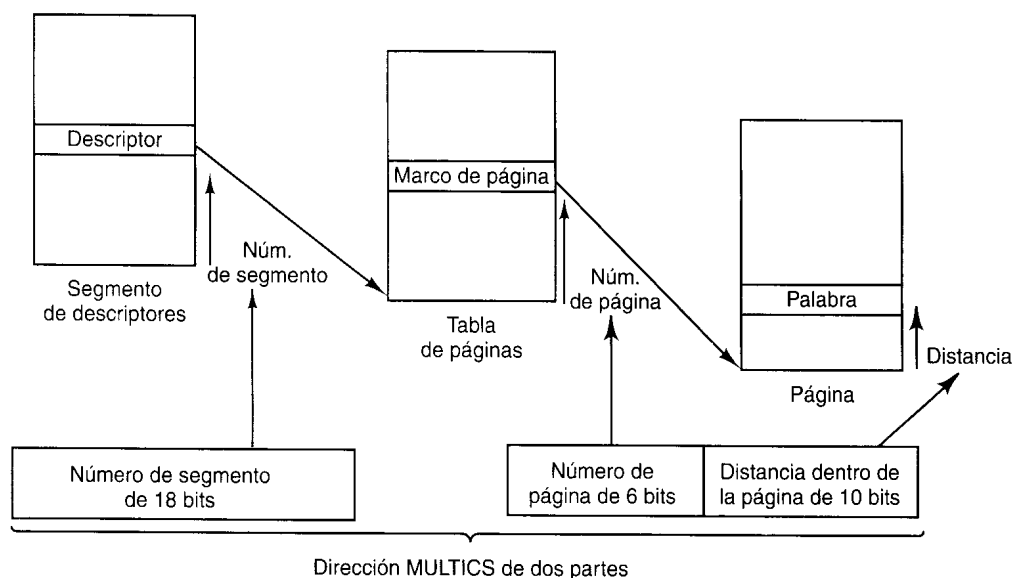


Figura 6-11. Conversión de una dirección MULTICS de dos partes en una dirección de memoria principal.

6.1.8 Memoria virtual en el Pentium II

El Pentium II tiene un sistema de memoria virtual avanzado que maneja paginación por demanda, segmentación pura y segmentación con paginación. El corazón de la memoria virtual del Pentium II consiste de dos tablas: la **tabla de descriptors locales (LDT, Local Descriptor Table)** y la **tabla de descriptors globales (GDT, Global Descriptor Table)**. Cada programa tiene su propia LDT, pero hay una sola GDT compartida por todos los programas de la computadora. La LDT describe los segmentos locales de cada programa, incluidos los de código, datos, pila, etc., mientras que la GDT describe los segmentos del sistema, incluido el sistema operativo mismo.

Como explicamos en el capítulo 5, para que un programa de Pentium II accese a un segmento, primero debe cargar un selector de ese segmento en uno de sus registros de segmento. Durante la ejecución, CS contiene el selector del segmento de código, DS contiene el selector del segmento de datos, etc. Cada selector es un número de 16 bits, como se muestra en la figura 6-12.

Uno de los bits del selector indica si el segmento es local o global (es decir, si está en la LDT o en la GDT). Otros 13 bits especifican el número de entrada de la LDT o GDT, así que estas tablas sólo pueden contener 8K (2^{13}) descriptors de segmentos. Los otros dos bits

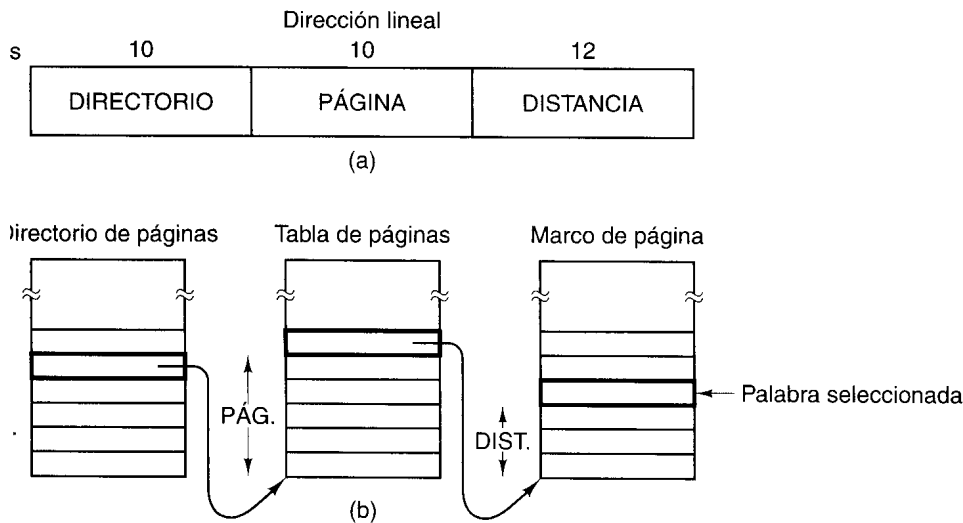


Figura 6-15. Transformación de una dirección lineal en una dirección física.

el campo **PÁGINA** como índice de la tabla de páginas para encontrar la dirección física marco de página. Por último, **DISTANCIA** se suma a la dirección del marco de página para obtener la dirección física del byte o palabra direccionado.

Las entradas de la tabla de páginas tienen 32 bits cada una, de los cuales 20 contienen un número de marco de página. Los demás bits controlan el acceso e indican si la página está válida o no. El hardware ajusta estos bits que son utilizados por el sistema operativo, los bits de rotación y otros bits utilitarios.

Cada tabla de páginas tiene entradas para 1024 marcos de página de 4K, por lo que una tabla de páginas maneja 4 MB de memoria. Un segmento menor que 4M tendrá un directorio de páginas con una sola entrada, un apuntador a su única tabla de páginas. Así, el overhead extra en el caso de un segmento corto es de sólo dos páginas, en lugar del millón de páginas que se necesitaría en una tabla de páginas de un solo nivel.

A fin de evitar referencias repetidas a la memoria, la MMU del Pentium II cuenta con hardware especial que busca las combinaciones **DIRECTORIO-PÁGINA** utilizadas más recientemente y las transforma en la dirección física del marco de página correspondiente. Sólo se evan a cabo los pasos de la figura 6-15 si la combinación actual no se usó recientemente.

Si nos ponemos a pensar un poco, veremos que si se usa paginación en realidad no tiene que el campo **BASE** del descriptor sea distinto de cero. Para lo único que sirve **BASE** es desplazarse un poco dentro del directorio de páginas a fin de usar una entrada en ese lugar en lugar de al principio. La verdadera razón para incluir **BASE** es hacer posible la segmentación pura (no paginada), y mantener la compatibilidad con el viejo 80286, que no usa paginación.

También vale la pena señalar que si una aplicación dada no necesita segmentación y se forma con un solo espacio de direcciones paginado de 32 bits, es fácil satisfacerla. Se asigna

a todos los registros de segmento el mismo selector, cuyo descriptor tiene **BASE = 0** y **LÍMITE** igual al máximo. Entonces, la distancia de la instrucción será la dirección lineal, empleando un solo espacio de direcciones; esto es, de hecho, paginación tradicional.

Con esto terminamos nuestro tratamiento de la memoria virtual en el Pentium II. Sin embargo, vale la pena hablar un poco de protección, ya que este tema está íntimamente relacionado con la memoria virtual. El Pentium II maneja cuatro niveles de protección, siendo el nivel 0 el que goza de más privilegios, y el nivel 3, el que menos. Dichos niveles se muestran en la figura 6-16. En un instante dado, un programa en ejecución está en cierto nivel, indicado por un campo de dos bits en su **palabra de estado del programa (PSW)**, un registro en hardware que contiene los códigos de condición y varios otros bits de estado. Además, cada segmento del sistema pertenece a un nivel dado.

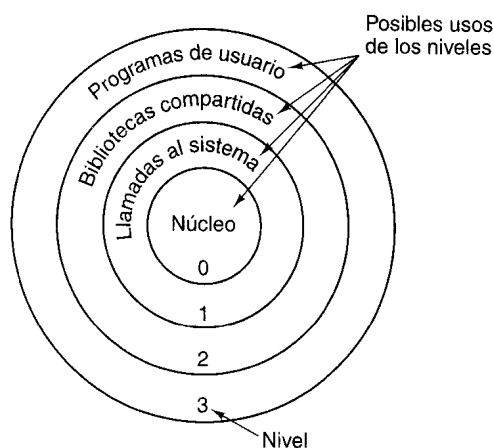


Figura 6-16. Protección en el Pentium II.

En tanto un programa se limite a usar segmentos que estén en su mismo nivel, todo funcionará de maravilla. Se permiten los intentos de acceder a datos que están en un nivel más alto, pero los intentos de acceder a datos en un nivel inferior están prohibidos y causan trampas. Los intentos por invocar procedimientos de otro nivel (más alto o más bajo) están permitidos, pero de forma muy controlada. Para efectuar una llamada internivel, la instrucción **CALL** debe contener un selector en lugar de una dirección. Este selector designa un descriptor llamado **puerta de llamada**, que da la dirección del procedimiento que se desea invocar. Así, no es posible saltar a la mitad de un segmento de código arbitrario en un nivel distinto; sólo pueden usarse los puntos de ingreso oficiales.

En la figura 6-16 se sugiere un posible uso de este mecanismo. En el nivel 0 está el núcleo del sistema operativo, que se encarga de la E/S, la gestión de memoria y otras tareas cruciales. En el nivel 1 está el manejador de llamadas al sistema. Los programas de usuario pueden invocar procedimientos de este nivel para solicitar que se efectúen llamadas al sistema, pero sólo es posible invocar una lista específica y protegida de procedimientos. El nivel 2 contiene procedimientos de biblioteca, posiblemente compartidos entre muchos programas

en ejecución. Los programas de usuario pueden invocar estos procedimientos, pero no modificarlos. Por último, los programas de usuario se ejecutan en el nivel 3, que es el que menos protección tiene. Al igual que el esquema de gestión de memoria del Pentium II, el sistema de protección se basa en el de MULTICS.

Las trampas e interrupciones usan un mecanismo similar a las puertas de llamada; también hacen referencia a descriptores, no a direcciones absolutas, y dichos descriptores apuntan a procedimientos específicos que deben ejecutarse. El campo TIPO de la figura 6-13 distingue entre segmentos de código, segmentos de datos y los diversos tipos de puertas.

6.1.9 Memoria virtual en el UltraSPARC II

El UltraSPARC II es una máquina de 64 bits y maneja una memoria virtual paginada basada en una dirección virtual de 64 bits. Sin embargo, por razones de ingeniería y costos, los programas no pueden usar todo el espacio de direcciones virtual de 64 bits. Sólo se reconocen 44 bits, por lo que los programas no pueden exceder 1.8×10^{13} bytes. La memoria virtual permitida se divide en dos zonas de 2^{43} bytes cada una, una en la parte superior del espacio de direcciones virtual y una en la parte inferior. En medio hay un hueco que contiene direcciones que no pueden usarse; un intento por usarlas causa un fallo de página.

La memoria física máxima en un UltraSPARC II es de 2^{41} bytes, que equivale a unos 2200 GB, lo bastante grande para casi todas las aplicaciones ordinarias. Se reconocen cuatro tamaños de página: 8 KB, 64 KB, 512 KB y 4 MB. En la figura 6-17 se ilustran las correspondencias implícitas en estos cuatro tamaños de página.

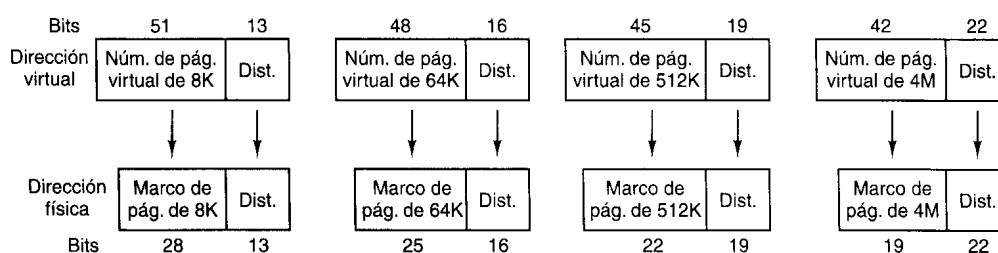


Figura 6-17. Transformaciones virtual-física en el UltraSPARC II.

Por lo extremadamente grande de espacio de direcciones virtual, una tabla de páginas sencilla como la del Pentium II no sería práctica. En vez de ello, la MMU del UltraSPARC II adopta una estrategia muy diferente: contiene una tabla en hardware llamada **buffer de consulta para traducción (TLB, Translation Lookaside Buffer)** que transforma números de página virtual en números de marco de página físico. Para el tamaño de página de 8K hay 2^{31} números de página virtual, o sea, más de dos millones. Es evidente que no todos pueden estar en el mapa.

En vez de ello, el TLB sólo contiene los números de página virtual más recientemente usados. Se sigue la pista por separado a las páginas de instrucciones y a las de datos, y el TLB guarda los 64 números de página virtual más recientemente usados de cada categoría. Cada entrada del TLB contiene un número de página virtual y el número de marco de página físico correspondiente. Cuando se presentan a la MMU un número de proceso, llamado **contexto**, y