MODO PROTEGIDO

Registros de soporte en modo protegido

El procesador apenas inicia me da la dirección base y el límite, aunque sea para acceder a un segmento.

1. ***Registro de la Tabla de Descriptores Globales (GDTR) - 48 bits:*** Contiene la dirección base de 32 bits y el límite de 16 bits de la Tabla Global de Descriptores (GDT).

2. ***Registro de la Tabla de Descriptores de Interrupción (IDTR) - 48 bits:*** Contiene la dirección base lineal de 32 bits y el límite de 16 bits de la tabla de descriptores de interrupción (IDT).

3. ***Registro de la tabla de descriptores locales (LDTR) - 16 bits:*** Contiene el selector de 16 bits para el Descriptor de la Tabla de Descriptores Locales.

4. ***Registro de tareas (TR) - 16 bits:*** Contiene el selector de 16 bits para el descriptor del segmento de estado de la tarea

Traslación de direcciones lógicas a físicas

El procesador Pentium tiene tres espacios de direcciones distintos: Lógico, lineal y físico. Una dirección lógica consiste en un selector y un offset. Un selector es el contenido de un registro de segmento.

Un selector es el contenido de un registro de segmentos. Cada selector de segmento tiene una dirección de base lineal asociada a él, que se almacena en el descriptor de segmentos. Un selector se usa para apuntar al descriptor de segmentos en la tabla de descriptores. La dirección base lineal del descriptor de segmentos es entonces sumada a los 32-bit del offset para generar la dirección lineal de 32-bit.

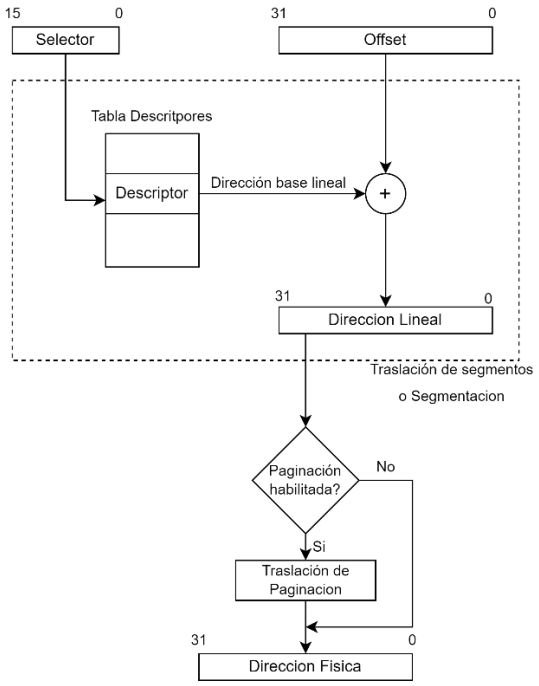


Fig. 1 Descripción general de la traslación de direcciones

Este proceso se conoce como segmentación o traslación de segmentos. Si la unidad de paginación no está habilitada, la dirección lineal de 32 bits corresponde a la dirección física. Pero si la unidad de paginación está habilitada, el mecanismo de paginación traslada el espacio de direcciones lineales al espacio de direcciones físicas mediante la traslación de paginación.

SEGMENTACIÓN

La segmentación es un proceso de conversión de la dirección lógica en una dirección lineal. El selector es usado para acceder al descriptor en una tabla de descriptores. Los 13-bits de índice del selector se multiplica por 8 y es usada como un puntero para el descriptor deseado, esto se hace porque cada descriptor es de 8 bytes.

El descriptor, de la tabla de descriptores contiene principalmente la dirección base, el límite de segmento y el byte derechos de acceso. El procesador suma la dirección base del descriptor a la dirección efectiva u offset para generar la dirección lineal para acceder a la ubicación exacta de mi segmento.

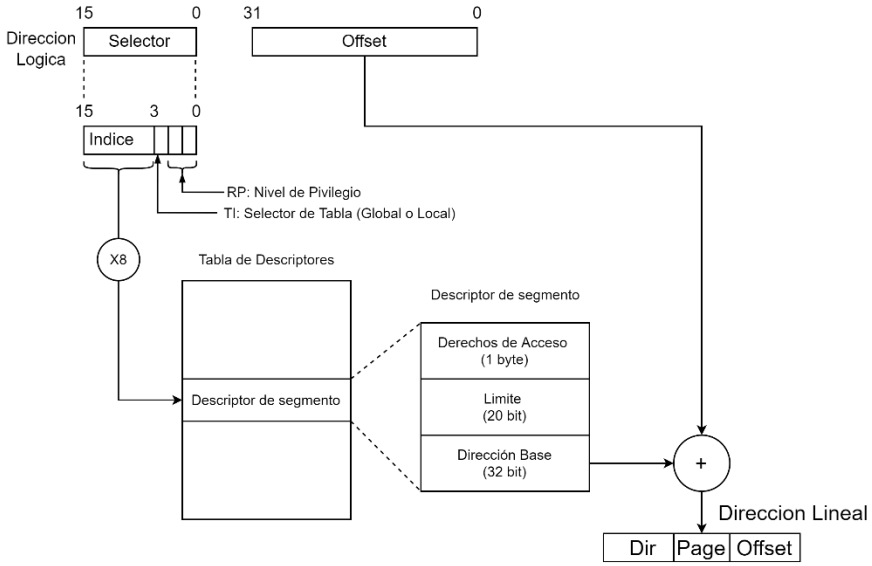


Fig. 2 Mecanismo de traslacion de segmento

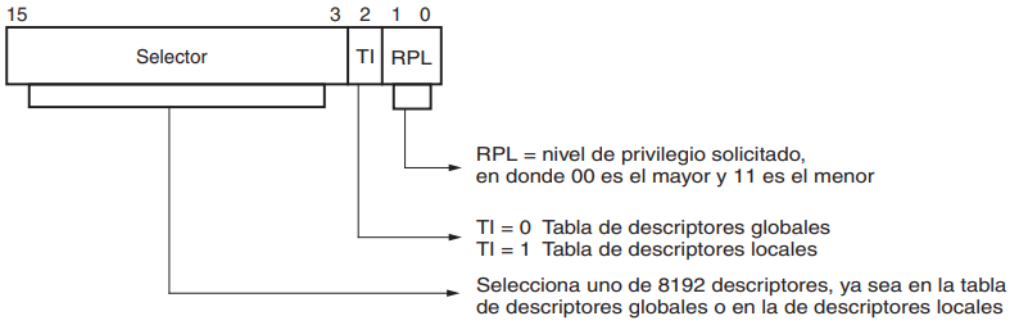
La componente del selector contiene dos bits que son para el nivel de privilegio, es decir cuando se quiere acceder a un segmento la unidad de gestión de memoria (MMU) compara el nivel de privilegio del selector con el descriptor al cual se quiere acceder. Si el nivel de privilegio solicitado concuerda o tiene mayor prioridad que el nivel de privilegio establecido por el byte de derechos de acceso, se otorga el acceso.

Hay dos categorías principales de tablas de descriptores:

- La Tabla de Descriptores Global (GDT) es una tabla de descriptores de propósito general, puede ser utilizada por todos los programas para referenciar segmentos de memoria.

- Mientras que una Tabla de Descriptores Locales (LDT) es usada para para tareas individuales del sistema.

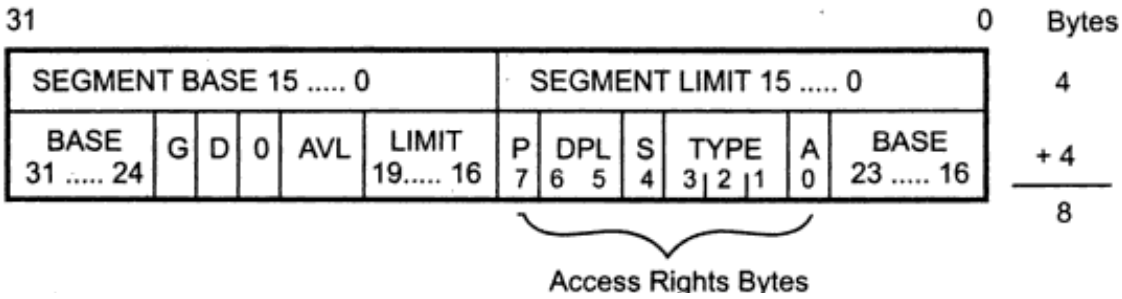
El bit indicador de tabla (TI) en el selector decide a qué tabla de descriptores debe referirse el selector, selecciona ya sea la tabla de descriptores globales (TI = 0) o la tabla de descriptores locales (TI = 1).



Descriptores de segmento y gestión de la memoria mediante segmentación

En modo protegido, la unidad de gestión de memoria (MMU) usa el selector de segmentos para acceder a un descriptor deseado en la tabla de descriptores, el descriptor de segmento es una estructura especial que describe al segmento, exactamente cada descriptor de segmento debe estar referido para cada segmento en memoria.

Los descriptores de segmentos proporcionan algunos atributos como la dirección de base lineal de 32-bits, la longitud del segmento de 20-bits, el nivel de protección de escritura, lectura, y acceso a niveles privilegiados. Para la dirección base se concatena las dos direcciones base de abajo y luego las de arriba formando la dirección base de 32bits.



Tablas de Descriptores

El límite máximo para la longitud de la tabla de descriptores es de 64KBytes y sabemos que cada descriptor toma 8 bytes para almacenar la información de un segmento particular. Así que la tabla de descriptores puede tener hasta 8192 descriptores.

Tipos de tablas descriptoras:

- ***Tabla global de descriptores (GDT):*** Es una tabla de propósito general, que puede ser utilizada por todos los programas para referenciar segmentos de memoria.
- ***Tabla de descriptores locales (LDT):*** Se configuran en el sistema para una tarea individual.
- ***Tabla de descriptores de interrupción (IDT):*** Contiene los descriptores de segmento que definen las rutinas de manejo de excepciones de las interrupciones.

Registro de tabla de descriptores global (GDTR): El contenido de GDTR es usado para definir la GDT en el espacio de direcciones de memoria física del procesador Pentium, ya que especifica el límite y la base, es decir la cantidad de descriptores.

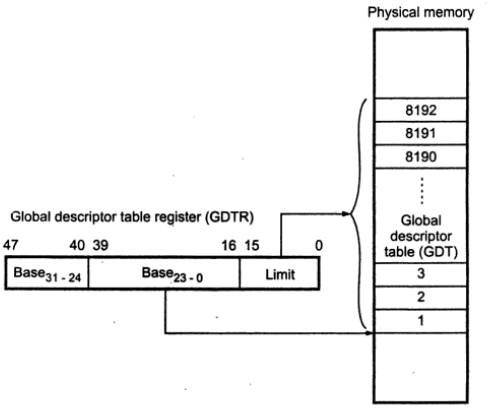


Fig. 4.13 GDTR and GDT

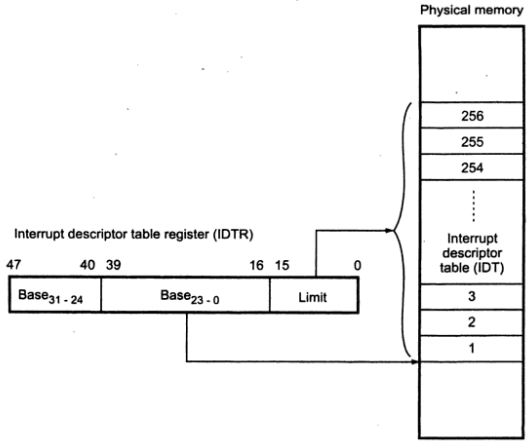


Fig. 4.14 IDTR and IDT

Registro de la tabla de descriptores de interrupción (IDTR): Es igual que GDTR, pero para IDT.

Registro de la tabla de descriptores local (LDTR): Es un registro de 16-bits no de 48-bits como las anteriores, no especifica ni el límite ni la base, pero especifica la dirección del descriptor para la LDT almacenada en la GDT.

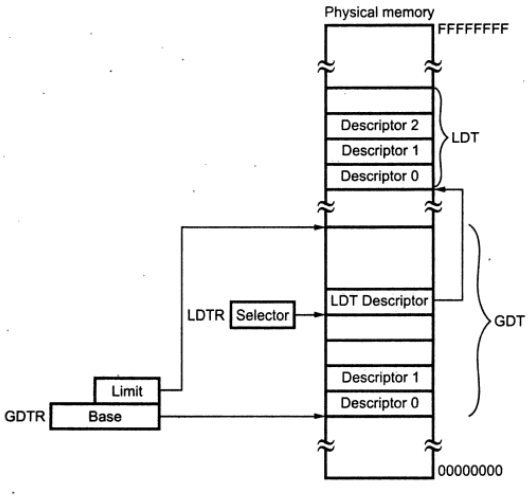


Fig. 4.15 Global and local descriptor tables

Primero tengo que tener el bit TI puesto en 1 y luego de usar el Selector de LDTR para que apunte al descriptor de la LDT de GDT que me da mi limite y base, vuelvo al selector normal para que apunte a la LDT ya que este tiene la dirección de base lineal asociada al descriptor de segmento de la LDT, y de ahí sumar la dirección de base lineal con el offset para obtener mi dato.

LDTR almacena un selector que apunta al descriptor de LDT en la GDT (Direccionamiento indirecto). Cuando un selector es cargado en el LDTR el correspondiente descriptor se localiza en la GDT (Digamos que el selector que se cargó apunta al descriptor de LDT en GDT), el contenido de este descriptor define la LDT.

Como sabemos ya, un selector es el contenido de un registro de segmento el cual apunta a un descriptor en la tabla de descriptores. Esta parte del registro de segmentos es visible para el programador, pero también tiene una parte oculta la cual está referida como registro cache del descriptor de segmento. Utilizando estos registros el procesador Pentium almacena la información del descriptor, evitando así la necesidad de consultar una tabla de descriptores cada vez que se accede a la memoria. El contenido del registro de segmento (parte visible) es manipulado por los programas, mientras que el contenido del registro de caché del descriptor de segmento (parte oculta) es manipulado por el procesador. Una vez que los descriptores se almacenan en la caché, las subsiguientes referencias a ellos se realizan sin ninguna sobrecarga para la carga del descriptor.

PAGINACION

La paginación es la segunda fase de la traslación de direcciones, transforma una dirección lineal en una dirección física (este paso es opcional). Tiene efecto únicamente cuando el bit PG del registro CR0 está habilitado.

El objetivo de este esquema de dos niveles es reducir la cantidad de RAM necesaria para las tablas de páginas por proceso. Cuando la paginación está habilitada la unidad de paginación organiza el espacio de la dirección física, que se organiza en 2^20 páginas de 4Kbytes.

Registro y tablas de soporte

Hay 3 componentes para el mecanismo de paginación:

- ***El directorio de páginas:*** Tiene forma de tabla y esta formada por descriptores de 32 bits y deben contener 1024 descriptores debido a su tamaño de 4Kbytes.
- ***Tablas de páginas:*** Tiene forma de tabla y esta formada por descriptores de 32 bits y deben contener 1024 descriptores debido a su tamaño de 4Kbytes.
- ***Marco de páginas:*** Es una entidad de 4Kbytes de direcciones contiguas de memoria física.

El procesador utiliza dos niveles de tablas para trasladar la dirección lineal en una dirección física (La primera tabla de traslación se llama Directorio de páginas y la segunda se llama Tabla de páginas). El procesador divide internamente una dirección lineal en tres campos: Dos campos de 10 bits cada uno y un campo de 12 bits.

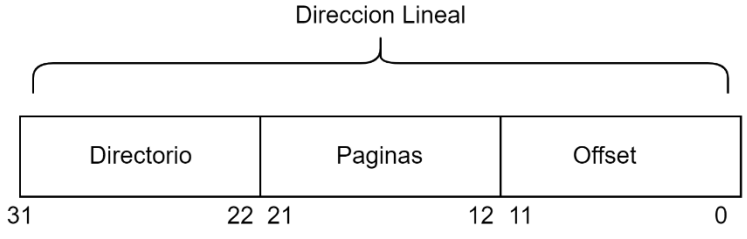


Fig. 4 Formato de Direccion Lineal

Los 10 bits más significativos (campo DIR) de la dirección lineal se utilizan como índice en un directorio de páginas. Los siguientes 10 bits más significativos (campo PAGE) de la dirección lineal se utilizan como índice en la tabla de páginas determinada por el directorio de páginas.

Los 12 bits menos significativos (OFFSET) seleccionan uno de los 4 Kbyte de memoria del marco de páginas determinado por la tabla de páginas. La dirección física del directorio de páginas actual se almacena en el registro de control (CR3) que también se denomina registro base del directorio de páginas (PDBR). Cuando arranca el procesador genera una dirección del directorio de página y la almacena en PDBR.

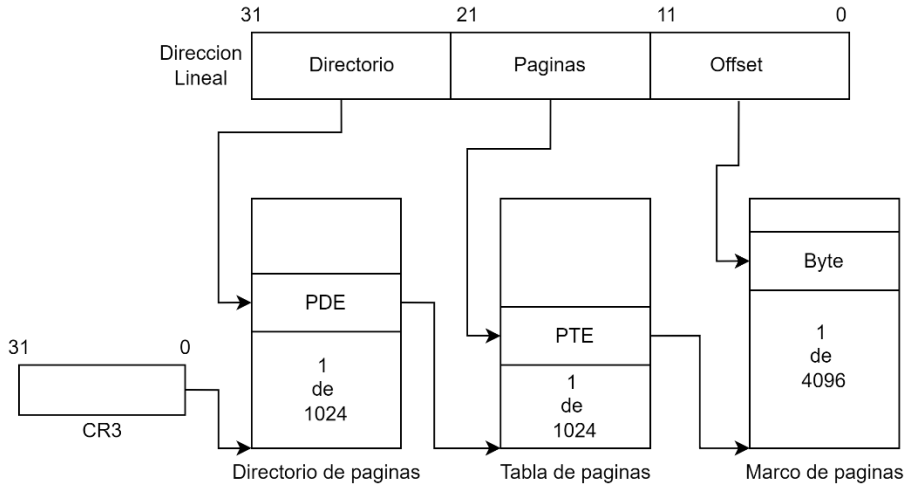


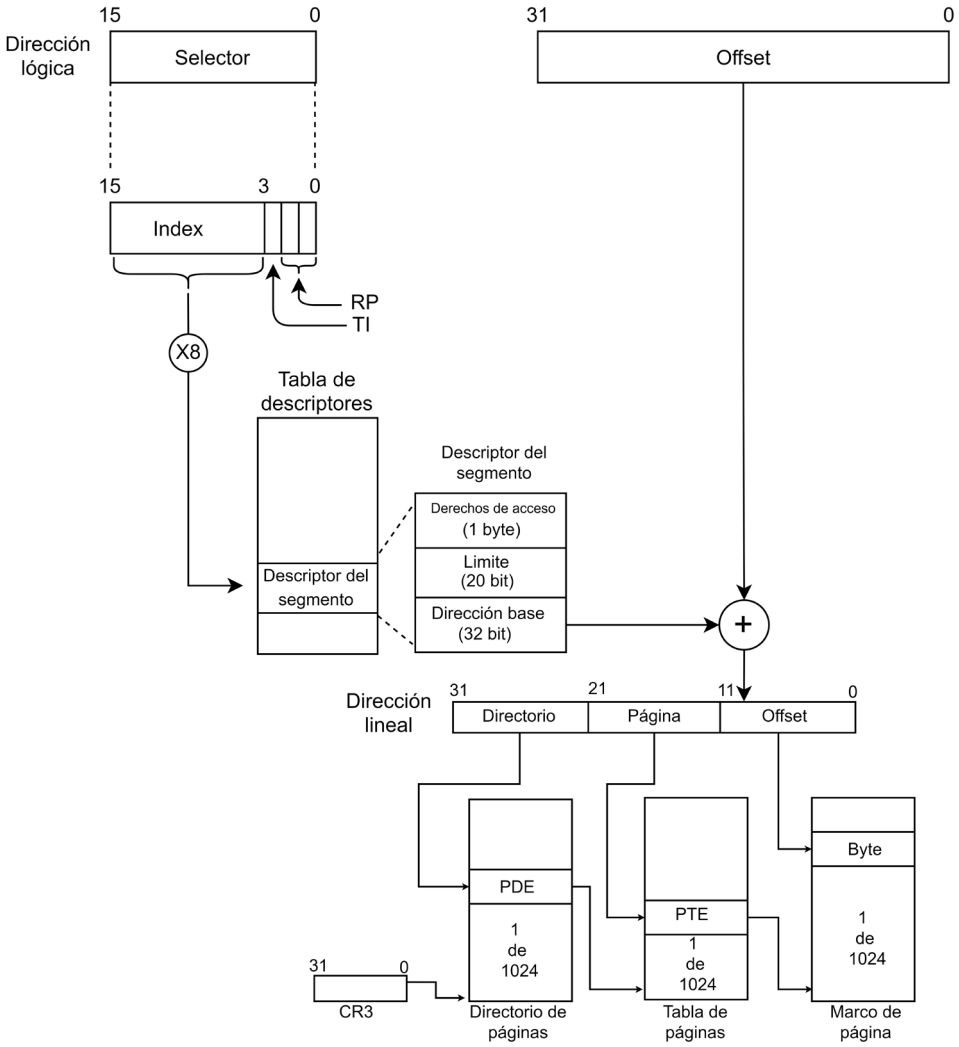
Fig. 5 Traslacion de dirección lineal a física

El descriptor de un directorio de páginas se denomina Entrada de directorio de páginas (PDE) y el descriptor de la tabla de páginas se denomina Entrada de tabla de páginas (PTE).

En la siguiente imagen se muestra el proceso completo de traducción de una dirección lógica a una dirección física con el método de paginación (SIN TLB):

1. El programa en assembler quiere acceder a una dirección y coloca el selector y el offset. El offset es de 32 bits porque puede ser cualquier cosa (PC, reg, etc.). El offset y los descriptores están en RAM.
2. El Index del selector se multiplica por 8 porque el descriptor requiere de 8 Bytes.
3. El CPU comprueba los permisos con el RPL y el DPL. Si se cumple que RPL <= que DPL se puede acceder al descriptor y usando la base + offset se llega a la dirección lineal.
4. Si la paginación esta OFF esta dirección lineal es igual a la dirección física. Si no, se hace la traslación.
5. Partiendo del CR3 se usan los 10 bits de DIRECTORY como offset dentro de una tabla que se llama directorio de página, y acá con estos 10 bits puedo tener una entrada de unas 1024 posibles entradas de tabla de directorio. Supongamos que direcciona a PDE, donde esta tiene la dirección base de la TABLA DE PAGINA.
6. Con los bits PAGE hace un offset y obtiene una entrada en la TABLA DE PAGINA que le llama PTE, y esto tiene la dirección base de un MARCO DE PAGINA que esto si sería la memoria física.
7. Con los bits OFFSET más la dirección base del marco de página yo me muevo algún Byte de la memoria física que yo quería.

DATO: Se necesitan 4 accesos a memoria para acceder al dato.

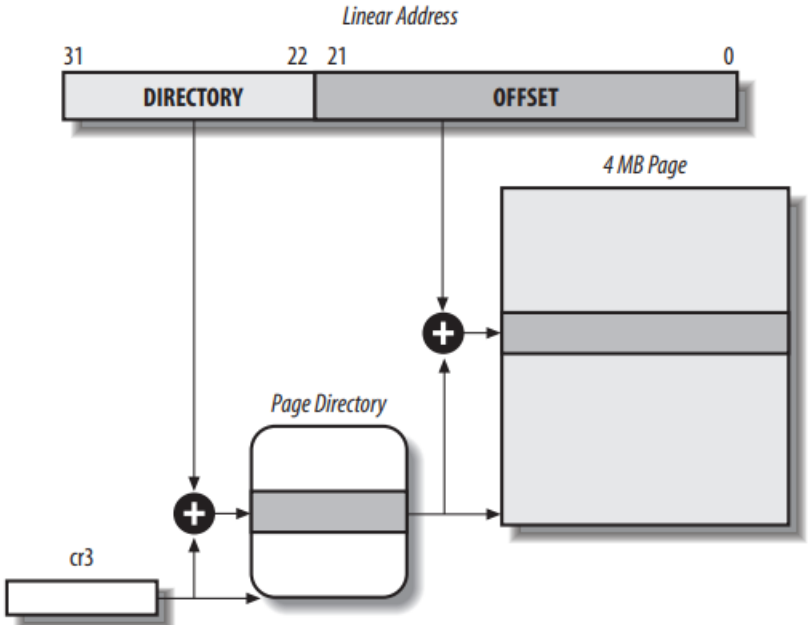


Cache de Traslación de Pagina (TLB)

Para que el procesador en cada acceso de memoria no tenga que pasar por los dos niveles de tabla, el procesador almacena las entradas de tablas de páginas usadas recientemente (PDE y PTE), en una cache llamada TLB, esta almacena hasta 32 entradas de tabla páginas. Cuando el procesador genera una dirección lineal que mapea a la entrada de tabla de páginas (PTE) ya almacenadas en la cache, el procesador puede usar esta información para llegar al segmento del marco de página, pero solo compara los primeros 20 bits en el TLB.

Operación de paginación

El mecanismo de paginación obtiene una dirección lineal de 32-bits desde la unidad de segmentación. Los 20 bits superiores de la dirección lineal (Campo de Directorio y Campo de página) son comparados con las 32 entradas del TLB de 4Kbyte de tamaño cada una para determinar si hay alguna coincidencia, si hay una coincidencia entonces los 32 bits de la dirección física son calculados y es almacenada en el bus de direcciones. Es necesario cambiar las entradas de la cache cuando las tablas de página son cambiadas, esto se puede hacer realizando un cambio de tareas para que tengan un diferente CR3, o moviendo el contenido de CR3 a EAX.



PROTECCIÓN

El procesador utiliza mecanismos de protección a nivel de segmento y a nivel de privilegio para proteger las secciones críticas.

Protección Por Segmentación

Cuando se intenta acceder a un segmento en primer lugar, el procesador comprueba si la tabla de descriptores indexada por el selector contiene un descriptor válido para ese selector. Si el selector intenta acceder a una ubicación fuera del límite de la tabla de descriptores o la ubicación indexada por el selector en la tabla de descriptores no contiene un descriptor válido, se produce una excepción. El procesador también comprueba si el descriptor de segmento es del tipo adecuado para ser cargado en la caché.

Si se cumplen todas las condiciones de protección anteriores, los bytes de límite, base y derechos de acceso del descriptor del segmento se copian en la parte oculta del registro del segmento. El procesador Pentium comprueba entonces el bit P (presente) del byte de acceso para ver si el segmento para ese descriptor está presente.

Después de que un selector de segmento y un descriptor son cargados en el registro de segmento, chequeos adicionales se realizan cada vez que una ubicación en el segmento actual es accedida. Estos chequeos son de tipo y de límite.

Chequeo de tipo: Es usado para detectar cuando algún programa está intentando usar segmentos de modos no especificados para el programador.

Chequeo de límite: El campo de límite del descriptor de segmentos es utilizado para evitar que los programas se dirijan fuera de los segmentos, es decir que el offset más la base no exceda el tamaño.

Protección Por Nivel de Privilegios

Tiene 4 niveles de privilegio, el nivel cero representa el nivel de privilegio más alto.

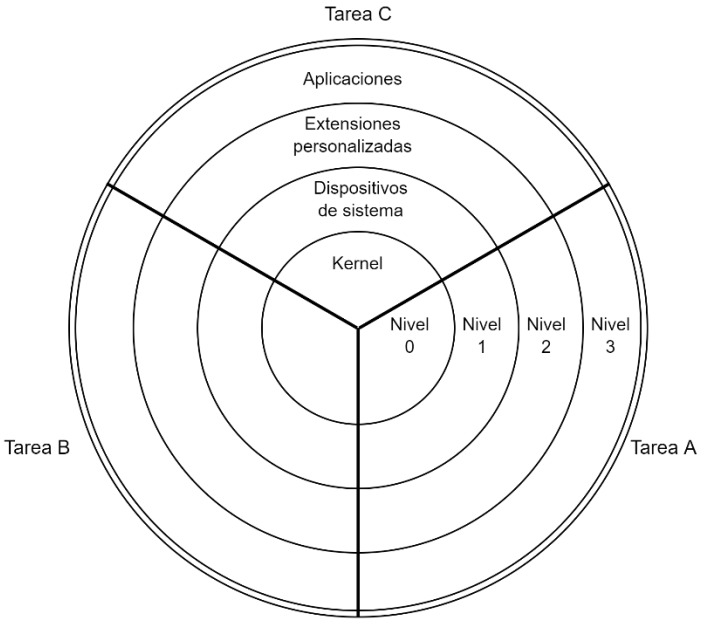


Fig.5 Asignación de niveles de privilegio

Nivel de privilegio del descriptor al cual queremos acceder **(DPL)**
Nivel de privilegio solicitado donde quiero ir **(RPL)**
Nivel de privilegio actual que es el descriptor almacenado en la parte oculta de la cache **(CPL)**
Para poder acceder a mi descriptor de segmentos la CPU compara estos 3 niveles de privilegios los cuales tienen que cumplir la siguiente condición $(CPL, RPL) \leq DPL$ $(Numericamente)$ para poder acceder al segmento.

Call Gate

La necesidad de usar call gate es que yo necesito saltar de un nivel de privilegio menor a uno mayor. Cuando hago un CALL mi selector ya no apunta a un descriptor de segmentos si no que apunta a un descriptor de puerta el cual contiene un selector y un offset, para acceder a dicho descriptor de puerta se tiene que cumplir que $(CPL, RPL) \leq DPL\ de\ Gate$ $(Numericamente)$, si se cumple esto entonces el descriptor se carga en la porción oculta del registro de segmentos y el selector en la parte visible por lo cual ahora mi **CPL = DPL de Gate** y mi **RPL = DPL Objetivo**, por ejemplo si quiero ir de un **PL = 3** a **PL = 0** mi **DPL de Gate = 3 o 4** para poder acceder al descriptor de puerta, entonces una vez que accedí se carga mi selector del descriptor de puerta en la parte visible con un **RPL =0** debido a que apunto a un descriptor de segmento que está en **PL = 0** y en la parte oculta se copia mi descriptor de puerta por lo tanto mi **CPL = 3 o 4** y el **DPL Objetivo = 0** del descriptor de segmento que quiero acceder si la comparación de la CPU cumple que $DPL\ Objetivo \leq (CPL, RPL) \leq DPL\ de\ Gate$ $(Numericamente)$. Entonces accedo a mi descriptor en el cual obtengo la dirección base lineal y se la sumo al offset que contiene el descriptor de puerta, pudiendo así acceder a mi segmento de código que una vez que se ejecutó todo me retorna un valor a mi programa en el cual se hizo la llamada.

MULTITAREAS

Un sistema operativo multitarea es aquel que puede intercalar la ejecución simultánea de más de un proceso.

Segmento de estado de tareas (TSS): Es un tipo especial de segmento, utilizado para gestionar la tarea, el procesador lo usa como un block de notas, ósea que aquí se almacenan el contenido o contexto de una tarea.

TSS Descriptor: Al igual que otros segmentos, el segmento de estado de la tarea se define mediante un descriptor llamado descriptor TSS.

Registro de tareas (TR): Especifica la tarea que se está ejecutando actualmente apuntando al TSS. El Registro de Tarea es un selector para el TSS.

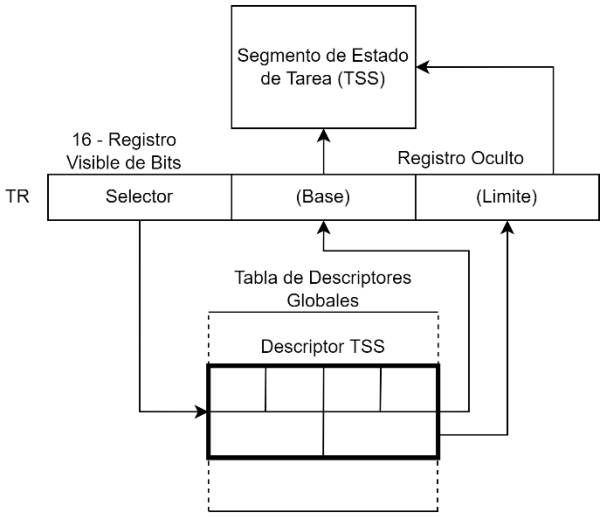


Fig. 6 Registro de tareas

El selector en la parte visible se utiliza para especificar un descriptor TSS en el GDT y la parte invisible se utiliza para almacenar en caché los valores base y límite del descriptor TSS. El procesador ofrece dos instrucciones para leer y modificar la porción visible de la tarea: **LTR** (Cargar registro de tareas) y **STR** (Almacenar el registro de tareas).

Las puertas de tarea (Task Gate), al igual que las puertas de llamada (Call Gate), son puertas especiales del sistema. Tienen su propio descriptor. Un descriptor de puerta de tarea no define un segmento de memoria, sino que actúa como un punto de interfaz entre el código de usuario y un segmento de estado de la tarea.

El campo Back Link del TSS de la tarea actual, contiene el selector de segmento de TSS de la tarea anterior. Este campo permite que se inicie un cambio de tarea a la tarea anterior cuando se ejecuta una instrucción IRET.

Cambio de Tareas sin Task Gate

Los pasos son los siguientes:

1. ***Comprobación de Privilegios:*** Esto se hace verificando DPL del TSS designado con RPL y CPL de la tarea actual.
2. ***Comprobación de los bits de límite y de presencia.***

3. ***Guardar el estado de la tarea actual:*** Encuentra la dirección base de la TSS actual almacenada en caché en el registro de tareas y copia los registros en el TSS actual. El selector de la tarea actual se guarda como selector de enlace posterior (back link) en la nueva tarea.
4. ***Carga del registro de tareas:*** La parte visible del registro de tareas se carga con el selector del descriptor TSS de la tarea designada. Esto establece el bit TS (Cambio de tarea) en la Palabra de Estado de la Máquina (MSW).
5. ***Reanudación de la ejecución.***

Cambio de Tareas con Task Gate

En este caso, se utiliza el método indirecto para la conmutación de tareas. La conmutación de tareas se realiza saltando o llamando a una puerta de tarea.

Cuando se utiliza la puerta de tarea, el DPL del nuevo descriptor TSS no se utiliza para la comprobación de privilegios. El DPL de la puerta de tarea se compara con el CPL y el RPL del selector de puerta. Si el DPL de la puerta de la tarea es numéricamente mayor o igual que el máximo del CPL y el RPL del selector de puerta, se permite que la tarea actual cambie a la tarea designada/nueva. Los pasos restantes son similares, salvo que para cargar el selector del descriptor TSS en la puerta de la tarea TR se remite a la instrucción CALL o JMP.

Manejo de interrupciones en modo real y protegido

Interrupciones: Son usadas para el manejo de eventos externos asíncronos para el procesador.
Excepciones: Manejan las condiciones detectadas por el procesador mismo en el curso de las ejecuciones de las instrucciones.

Modo real

Hay una tabla la cual contiene punteros que definen el comienzo de ISR. Cada puntero representa dos palabras. La palabra que tiene la dirección de memoria más alta contiene la dirección base del segmento y la que contiene la dirección en memoria más baja contiene el offset. Cada vez que una interrupción ocurre el procesador multiplica el numero de la interrupción por 4 para generar un índice en la IDT.

Modo Protegido

Ahora la IDT es un arreglo de descriptores de 8-byte y cuando una interrupción ocurre, su identificador que es un número se multiplica por 8 y se suma a la dirección base de IDT almacenada en IDTR. El resultado me apunta a un descriptor de puerta en la IDT que puede ser de 3 tipos (interrupt gate, trap gate, task gate). Este descriptor contiene un selector y un offset, de modo que el selector apunta al descriptor de segmento que queremos acceder obteniendo la dirección de base lineal que le sumamos el offset de la puerta para así obtener la dirección lineal y acceder al segmento donde se encuentra la ISR (rutina de servicio de interrupciones) que se ejecuta y luego se retorna al programa