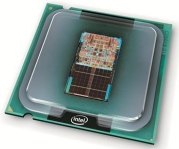


Procesadores IA-32 e Intel® 64

Gestión de Memoria: Paginación

Alejandro Furfaro

11 de noviembre de 2025

Temario

Temario

Paginación de la Memoria

- Segmentación:

Paginación de la Memoria

- Segmentación:
  - Provee un entorno flexible en la programación de aplicaciones.

Paginación de la Memoria

- Segmentación:
  - Provee un entorno flexible en la programación de aplicaciones.
  - Para la administración de la memoria por parte del sistema operativo, la variabilidad del tamaño de los segmentos introduce complejidad en los algoritmos de un sistema de memoria virtual.

Paginación de la Memoria

- Segmentación:
  - Provee un entorno flexible en la programación de aplicaciones.
  - Para la administración de la memoria por parte del sistema operativo, la variabilidad del tamaño de los segmentos introduce complejidad en los algoritmos de un sistema de memoria virtual.
- Paginación

Paginación de la Memoria

- Segmentación:
  - Provee un entorno flexible en la programación de aplicaciones.
  - Para la administración de la memoria por parte del sistema operativo, la variabilidad del tamaño de los segmentos introduce complejidad en los algoritmos de un sistema de memoria virtual.
- Paginación
  - Mas rígido para aplicar en la programación de aplicaciones.

Paginación de la Memoria

- Segmentación:
  - Provee un entorno flexible en la programación de aplicaciones.
  - Para la administración de la memoria por parte del sistema operativo, la variabilidad del tamaño de los segmentos introduce complejidad en los algoritmos de un sistema de memoria virtual.
- Paginación
  - Mas rígido para aplicar en la programación de aplicaciones.
  - Trabajar con bloques del mismo tamaño, simplifica el desarrollo del algoritmo de memoria virtual.

Paginación de la Memoria

- Esto ha hecho que tradicionalmente los sistemas operativos hayan diseñado su sistema de memoria virtual mediante la administración de bloques de memoria de tamaño fijo.

Paginación de la Memoria

- Esto ha hecho que tradicionalmente los sistemas operativos hayan diseñado su sistema de memoria virtual mediante la administración de bloques de memoria de tamaño fijo.
- De hecho, UNIX por citar al decano de los Sistemas Operativos, desde su concepción implementó la administración de la memoria utilizando bloques de tamaño uniforme.

Temario

Diagrama de generación de la dirección física

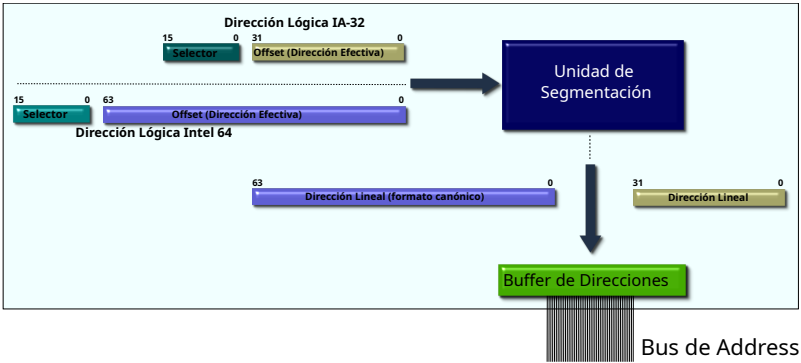
- Un segmento es finalmente un espacio lineal de direcciones, razón por la cual se denomina a este número de 32 o 64 bit obtenido por la Unidad de Segmentación, ***dirección lineal***

Diagrama de generación de la dirección física

- Un segmento es finalmente un espacio lineal de direcciones, razón por la cual se denomina a este número de 32 o 64 bit obtenido por la Unidad de Segmentación, ***dirección lineal***
- De no mediar otra etapa de hardware, este número sale por el bus de Address convertido en ***dirección física***

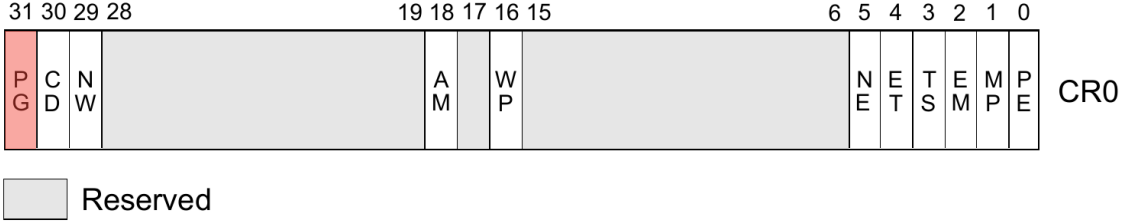
Diagrama de generación de la dirección física

- Un segmento es finalmente un espacio lineal de direcciones, razón por la cual se denomina a este número de 32 o 64 bit obtenido por la Unidad de Segmentación, ***dirección lineal***
- De no mediar otra etapa de hardware, este número sale por el bus de Address convertido en ***dirección física***



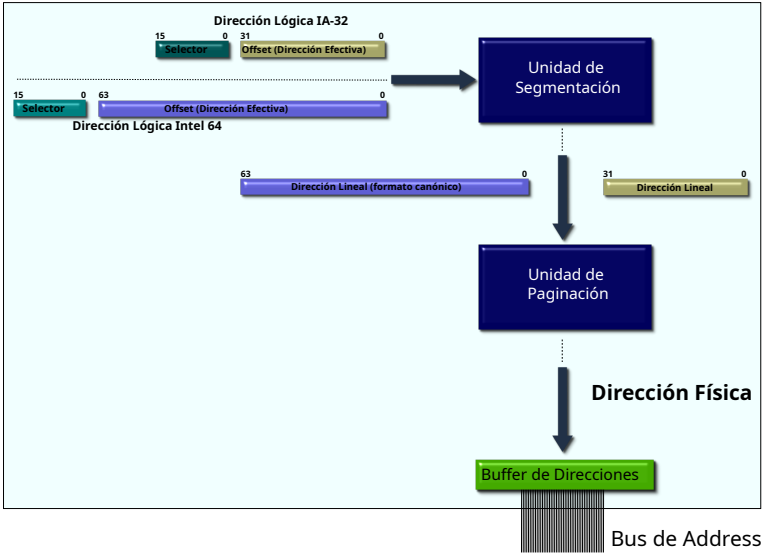
Habilitación de la Paginación

La Unidad de Paginación debe habilitarse seteando el bit **CR0.PG**



Importante: El procesador debe estar previamente en Modo Protegido, de otro modo #GP.

Diagrama de generación de la dirección física



Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- Fraccionar el espacio lineal de un segmento en páginas de menor tamaño.

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- Fraccionar el espacio lineal de un segmento en páginas de menor tamaño.
- Así el Sistema Operativo tiene en RAM únicamente unas pocas páginas de cada segmento

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- Fraccionar el espacio lineal de un segmento en páginas de menor tamaño.
- Así el Sistema Operativo tiene en RAM únicamente unas pocas páginas de cada segmento
- El resto en la Memoria Virtual para ser intercambiadas cuando se las necesite. No requiere el segmento completo el RAM.

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- Fraccionar el espacio lineal de un segmento en páginas de menor tamaño.
- Así el Sistema Operativo tiene en RAM únicamente unas pocas páginas de cada segmento
- El resto en la Memoria Virtual para ser intercambiadas cuando se las necesite. No requiere el segmento completo el RAM.
- Así el Sistema Operativo puede alojar en memoria mayor cantidad de procesos, ya que cada uno requiere menos memoria para iniciar su ejecución.

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- El software por lo general se construye de una serie de estructuras de iteración.

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- El software por lo general se construye de una serie de estructuras de iteración.
- O sea que una página de memoria de código puede ser ejecutada durante muchos ciclos de clock antes de requerirse otra.

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- El software por lo general se construye de una serie de estructuras de iteración.
- O sea que una página de memoria de código puede ser ejecutada durante muchos ciclos de clock antes de requerirse otra.
- Por lo tanto la frecuencia con que cada proceso bien programado requeriría bajar desde la Memoria Virtual otra página a la memoria Física, será bastante moderada.

Unidad de Paginación IA-32

La Unidad de Paginación, permitió a los procesadores IA-32:

- El software por lo general se construye de una serie de estructuras de iteración.
- O sea que una página de memoria de código puede ser ejecutada durante muchos ciclos de clock antes de requerirse otra.
- Por lo tanto la frecuencia con que cada proceso bien programado requeriría bajar desde la Memoria Virtual otra página a la memoria Física, será bastante moderada.
- En la práctica, disponer de paginación permitió ejecutar por primera vez un Sistema Operativo UNIX, en un procesador Intel.

Modos de paginación

Modos de paginación

32 Bits Es el modo original de paginación del 80386 (1984): páginas de 4 Kbytes. Con el lanzamiento del procesador Pentium Pro (1995) un modo de extensión de memoria física y tamaño de página denominado PSE (Page Size Extention).

Modos de paginación

32 Bits Es el modo original de paginación del 80386 (1984): páginas de 4 Kbytes. Con el lanzamiento del procesador Pentium Pro (1995) un modo de extensión de memoria física y tamaño de página denominado PSE (Page Size Extention).

PAE Introducido junto con PSE en el Pentium Pro, es el método que finalmente adoptaron los Sistemas Operativos como Linux para generar kernels de 32 bits capaces de acceder a direcciones de memoria mas allá de los 4 Gbytes. Por este motivo desde entonces hasta el presente se ha “ganado”el derecho a que se lo considere un modo en si mismo.

Modos de paginación

32 Bits Es el modo original de paginación del 80386 (1984): páginas de 4 Kbytes. Con el lanzamiento del procesador Pentium Pro (1995) un modo de extensión de memoria física y tamaño de página denominado PSE (Page Size Extention).

PAE Introducido junto con PSE en el Pentium Pro, es el método que finalmente adoptaron los Sistemas Operativos como Linux para generar kernels de 32 bits capaces de acceder a direcciones de memoria mas allá de los 4 Gbytes. Por este motivo desde entonces hasta el presente se ha “ganado”el derecho a que se lo considere un modo en si mismo.

IA-32e Basado en el PAE, es el modo de paginación que se utiliza cuando el procesador se setea en modo IA-32e, es decir, 64 bits puro.

Temario

Paginación IA-32: Páginas de 4 Kbytes

Paginación IA-32: Páginas de 4 Kbytes

- El procesador 80386, adoptó un tamaño de página de 4 Kbytes.

Paginación IA-32: Páginas de 4 Kbytes

- El procesador 80386, adoptó un tamaño de página de 4 Kbytes.
- Este tamaño es entonces el tamaño estándar que por default usa cualquier procesador subsiguiente que active la paginación (compatibilidad).

Paginación IA-32: Páginas de 4 Kbytes

- El procesador 80386, adoptó un tamaño de página de 4 Kbytes.
- Este tamaño es entonces el tamaño estándar que por default usa cualquier procesador subsiguiente que active la paginación (compatibilidad).
- A partir del Pentium Pro se han incluido otros tamaños de página, pero si se desea utilizar páginas de diferente tamaño que 4 Kbytes, siempre debe ser explícitamente seleccionado el otro tamaño.

Paginación IA-32: Páginas de 4 Kbytes

- El procesador 80386, adoptó un tamaño de página de 4 Kbytes.
- Este tamaño es entonces el tamaño estándar que por default usa cualquier procesador subsiguiente que active la paginación (compatibilidad).
- A partir del Pentium Pro se han incluido otros tamaños de página, pero si se desea utilizar páginas de diferente tamaño que 4 Kbytes, siempre debe ser explícitamente seleccionado el otro tamaño.
- El espacio lineal de 4 Gbytes, se divide en páginas de 4 Kbytes de manera completa (es decir 2^{20} páginas).

Estructuras necesarias

Estructuras necesarias

- Al igual que cuando hubo de tratar con los segmentos, el procesador necesitará similar información para tratar a las páginas.

  Dirección base Necesitamos saber en donde comienza cada página.

Estructuras necesarias

- Al igual que cuando hubo de tratar con los segmentos, el procesador necesitará similar información para tratar a las páginas.

Dirección base Necesitamos saber en donde comienza cada página.

Límite Necesitamos saber donde finaliza cada página.

Estructuras necesarias

- Al igual que cuando hubo de tratar con los segmentos, el procesador necesitará similar información para tratar a las páginas.

Dirección base	Necesitamos saber en donde comienza cada página.
Límite	Necesitamos saber donde finaliza cada página.
Atributos de acceso	Inevitablemente se necesitarán algunos bits para determinar permisos y derechos de acceso.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

  Dirección base :

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

  Dirección base :

  - Cada página comienza en la dirección de memoria siguiente a la del último byte de la página anterior.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

  Dirección base :

  - Cada página comienza en la dirección de memoria siguiente a la del último byte de la página anterior.
  - Es decir: inician a partir de direcciones de memoria alineadas a su tamaño.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

  Dirección base :

  - Cada página comienza en la dirección de memoria siguiente a la del último byte de la página anterior.
  - Es decir: inician a partir de direcciones de memoria alineadas a su tamaño.
  - Tamaño de la página = 2^n. En páginas de 4 Kbytes (2^{12} bytes) los 12 bits menos significativos de su dirección base valen siempre 0.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

Dirección base :

- Cada página comienza en la dirección de memoria siguiente a la del último byte de la página anterior.
- Es decir: inician a partir de direcciones de memoria alineadas a su tamaño.
- Tamaño de la página = 2^n. En páginas de 4 Kbytes (2^{12} bytes) los 12 bits menos significativos de su dirección base valen siempre 0.
- Dirección Base o ***Page Frame*** (marco de página) es el conjunto de bits mas significativos de la dirección de la página que necesitamos para especificar su dirección base.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

  Límite No hace falta especificarlo, ya que las páginas tienen tamaño fijo.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

Límite No hace falta especificarlo, ya que las páginas tienen tamaño fijo.

Atributos de acceso Inevitablemente se necesitarán algunos bits para determinar permisos y derechos de acceso.

Estructuras necesarias

- Aunque hay algunas simplificaciones, a saber:

  Límite No hace falta especificarlo, ya que las páginas tienen tamaño fijo.

  Atributos de acceso Inevitablemente se necesitarán algunos bits para determinar permisos y derechos de acceso.

- ***Conclusión***:
  Con 20 bits para la dirección base de la página mas 12 bits para los atributos podemos conformar un descriptor de páginas de 32 bits.

¿Cuantas páginas cubren el espacio lineal?

¿Cuantas páginas cubren el espacio lineal?

- Si las páginas son de 4 Kbytes (2^{12} bytes) y el espacio lineal máximo a paginar es de 4 Gbytes (2^{32} bytes), páginas que necesitamos para cubrirlo de manera completa es:

$$\frac{2^{32}}{2^{12}} = 2^{20} \tag{1}$$

¿Cuantas páginas cubren el espacio lineal?

- Si las páginas son de 4 Kbytes (2^{12} bytes) y el espacio lineal máximo a paginar es de 4 Gbytes (2^{32} bytes), páginas que necesitamos para cubrirlo de manera completa es:

$$\frac{2^{32}}{2^{12}} = 2^{20} \tag{1}$$

- Esta situación deriva en una tabla de descriptores de página **gigantesca**:
  En este caso, 2^{20} descriptores de 4 bytes, nos llevan a una tabla que ocupa 4 MBytes de memoria física.

¿Cuantas páginas cubren el espacio lineal?

- Si las páginas son de 4 Kbytes (2^{12} bytes) y el espacio lineal máximo a paginar es de 4 Gbytes (2^{32} bytes), páginas que necesitamos para cubrirlo de manera completa es:

$$\frac{2^{32}}{2^{12}} = 2^{20} \tag{1}$$

- Esta situación deriva en una tabla de descriptores de página **gigantesca**:
  En este caso, 2^{20} descriptores de 4 bytes, nos llevan a una tabla que ocupa 4 MBytes de memoria física.

- Por ello conviene pasar a estructuras tablas de descriptores de paginación por niveles.

¿Cuantas páginas cubren el espacio lineal?

- Si las páginas son de 4 Kbytes (2^{12} bytes) y el espacio lineal máximo a paginar es de 4 Gbytes (2^{32} bytes), páginas que necesitamos para cubrirlo de manera completa es:

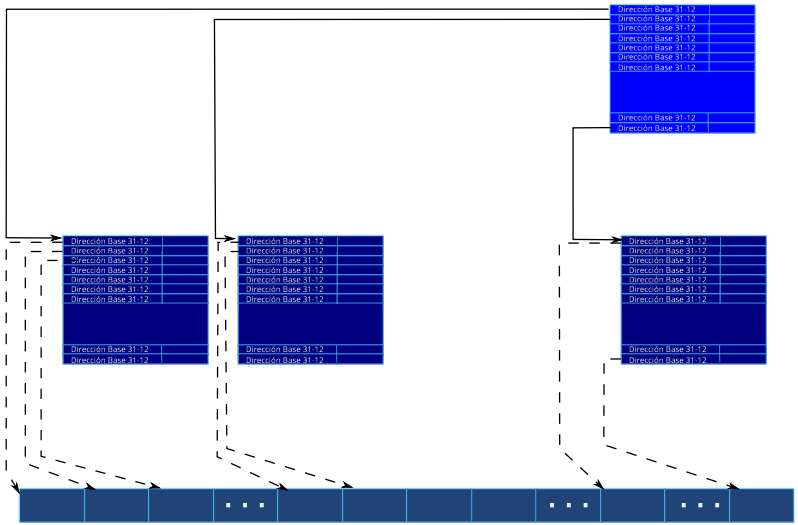
$$\frac{2^{32}}{2^{12}} = 2^{20} \quad (1)$$

- Esta situación deriva en una tabla de descriptores de página **gigantesca**:
  En este caso, 2^{20} descriptores de 4 bytes, nos llevan a una tabla que ocupa 4 MBytes de memoria física.

- Por ello conviene pasar a estructuras tablas de descriptores de paginación por niveles.

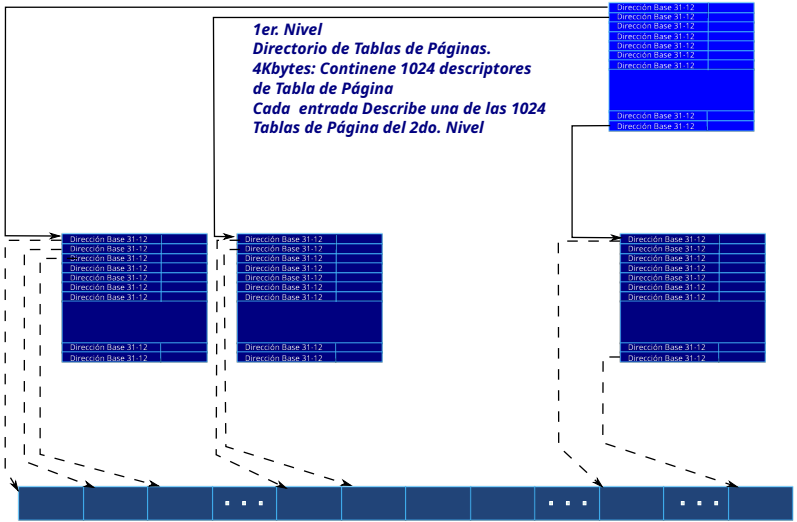
- ¿Como por niveles?... Veamos.

Niveles de Paginación



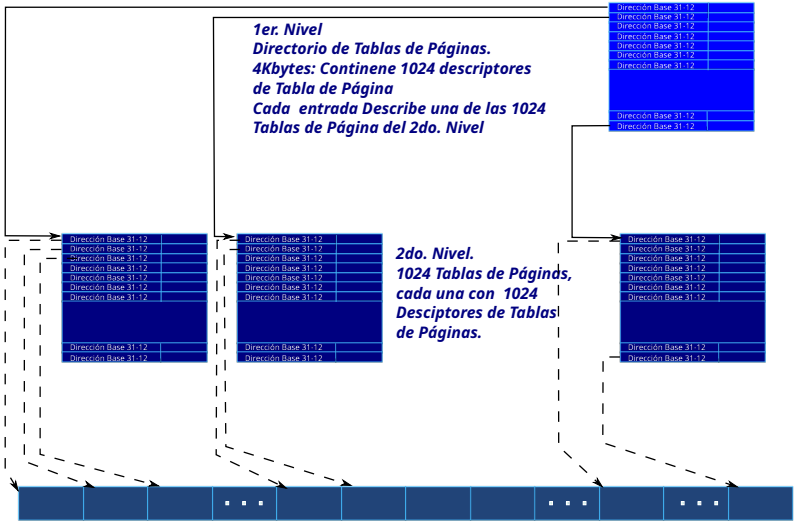
4 Gbytes de Memoria dividido en Páginas de 4 Kbytes.

Niveles de Paginación



4 Gbytes de Memoria dividido en Páginas de 4 Kbytes.

Niveles de Paginación



Paginas y tareas...

Paginas y tareas...

- La implementación en IA-32 de la paginación se ha pensado como un sistema de administración de memoria por tarea, de modo que cada tarea tiene su propia estructura de páginas.

Paginas y tareas...

- La implementación en IA-32 de la paginación se ha pensado como un sistema de administración de memoria por tarea, de modo que cada tarea tiene su propia estructura de páginas.
- Esta característica robustece la seguridad del Sistema Operativo en la administración de memoria.

Paginas y tareas...

- La implementación en IA-32 de la paginación se ha pensado como un sistema de administración de memoria por tarea, de modo que cada tarea tiene su propia estructura de páginas.
- Esta característica robustece la seguridad del Sistema Operativo en la administración de memoria.
- Por esta razón una tabla de 4 Mbytes por cada tarea para gestionar la memoria es un contrasentido en términos de eficiencia.

Paginas y tareas...

- La implementación en IA-32 de la paginación se ha pensado como un sistema de administración de memoria por tarea, de modo que cada tarea tiene su propia estructura de páginas.
- Esta característica robustece la seguridad del Sistema Operativo en la administración de memoria.
- Por esta razón una tabla de 4 Mbytes por cada tarea para gestionar la memoria es un contrasentido en términos de eficiencia.
- Se puede diseñar un sistema con una única tabla de páginas común a todas las tareas, pero tal enfoque da por tierra con la robustez en la de administración de memoria señalada anteriormente.

Páginas y tareas...

- El método permite iniciar la tarea con solo dos tablas de Paginación (dos tablas = dos páginas de 4Kbytes).

Páginas y tareas...

- El método permite iniciar la tarea con solo dos tablas de Paginación (dos tablas = dos páginas de 4Kbytes).

  1. Una página de 4Kbytes para el Directorio de Tablas de Página (de aquí en mas **PD**). Estructura indispensable pues es la raíz de la jerarquía de datos. En principio contendrá tan solo un descriptor válido.

Páginas y tareas...

- El método permite iniciar la tarea con solo dos tablas de Paginación (dos tablas = dos páginas de 4Kbytes).

  1. Una página de 4Kbytes para el Directorio de Tablas de Página (de aquí en mas **PD**). Estructura indispensable pues es la raíz de la jerarquía de datos. En principio contendrá tan solo un descriptor válido.

  2. Una página de 4Kbytes para una Tabla de Páginas (de aquí en mas **PT**) apuntada por la única entrada válida del Directorio, que puede almacenar hasta 1024 descriptores de páginas válidos de 4 Kbytes c/u.

Páginas y tareas...

- Las 1024 entradas contenidas en la **PT** describen sendas páginas de memoria que a su vez contendrán el código y los datos de la tarea.

Páginas y tareas...

- Las 1024 entradas contenidas en la **PT** describen sendas páginas de memoria que a su vez contendrán el código y los datos de la tarea.
- De este modo se puede iniciar un proceso con solo dos tablas de 4 Kbytes para administrar hasta 1024 páginas de 4 Kbytes, es decir 4 Mbytes para acomodar su código y datos.

Páginas y tareas...

- Las 1024 entradas contenidas en la **PT** describen sendas páginas de memoria que a su vez contendrán el código y los datos de la tarea.
- De este modo se puede iniciar un proceso con solo dos tablas de 4 Kbytes para administrar hasta 1024 páginas de 4 Kbytes, es decir 4 Mbytes para acomodar su código y datos.
- Esta cantidad es mas que suficiente para iniciar un proceso. ¿no?

Páginas y tareas...

- Las 1024 entradas contenidas en la **PT** describen sendas páginas de memoria que a su vez contendrán el código y los datos de la tarea.
- De este modo se puede iniciar un proceso con solo dos tablas de 4 Kbytes para administrar hasta 1024 páginas de 4 Kbytes, es decir 4 Mbytes para acomodar su código y datos.
- Esta cantidad es mas que suficiente para iniciar un proceso. ¿no?
- Por supuesto que cada proceso puede utilizar menos de 4 Mbytes de memoria al inicio (de hecho es lo habitual). Lo anterior es solo a efectos de cuantificar la cantidad de memoria que podemos asignarle consumiendo para su gestión solo 8 Kbytes.

Administración de memoria - Guidelines

Memoria dinámicamente alojable por proceso

Administración de memoria - Guidelines

Memoria dinámicamente alojable por proceso

- Lo habitual es que el Sistema Operativo habilite a cada proceso unas pocas páginas para código y datos.

Administración de memoria - Guidelines

Memoria dinámicamente alojable por proceso

- Lo habitual es que el Sistema Operativo habilite a cada proceso unas pocas páginas para código y datos.
- Conforme el proceso requiera memoria, se irán agregando en la tabla de páginas del segundo nivel de la estructura los descriptores necesarios.

Administración de memoria - Guidelines

Memoria dinámicamente alojable por proceso

- Lo habitual es que el Sistema Operativo habilite a cada proceso unas pocas páginas para código y datos.
- Conforme el proceso requiera memoria, se irán agregando en la tabla de páginas del segundo nivel de la estructura los descriptores necesarios.
- Este procedimiento permite asignar al proceso hasta 4 Mbytes de memoria.

Administración de memoria - Guidelines

Memoria dinámicamente alojable por proceso

- Lo habitual es que el Sistema Operativo habilite a cada proceso unas pocas páginas para código y datos.
- Conforme el proceso requiera memoria, se irán agregando en la tabla de páginas del segundo nivel de la estructura los descriptores necesarios.
- Este procedimiento permite asignar al proceso hasta 4 Mbytes de memoria.
- Llegado a ese límite, si solicita mas memoria, se genera un segundo descriptor en el **PD**, que permitirá habilitar otros 4 Kbytes de memoria para colocar allí hasta 1024 nuevos descriptores de Página, es decir otros 4 Mbytes máximo.

Traducción con Páginas de 4 Kbytes en 32 bits

Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.

Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.
- El procesador necesita conocer para cada tarea, la dirección física en donde se inicia la estructura de Descriptores de Páginas.

Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.
- El procesador necesita conocer para cada tarea, la dirección física en donde se inicia la estructura de Descriptores de Páginas.
- En estos procesador el Registro de Control **CR3** contiene esta dirección (***dirección física***).

Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.
- El procesador necesita conocer para cada tarea, la dirección física en donde se inicia la estructura de Descriptores de Páginas.
- En estos procesador el Registro de Control **CR3** contiene esta dirección (***dirección física***).
- En el próximo slide se muestra el proceso que describiremos a continuación.

Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.
- El procesador necesita conocer para cada tarea, la dirección física en donde se inicia la estructura de Descriptores de Páginas.
- En estos procesador el Registro de Control **CR3** contiene esta dirección (***dirección física***).
- En el próximo slide se muestra el proceso que describiremos a continuación.
- El procesador toma la ***dirección lineal*** y la divide en tres campos de bits que serán utilizados respectivamente como:
  - Índice en el **PD**, para determinar la dirección de inicio y derechos de acceso de la página que contiene la **PT** uno de cuyos descriptores corresponde a la página que se está buscando acceder en memoria física.

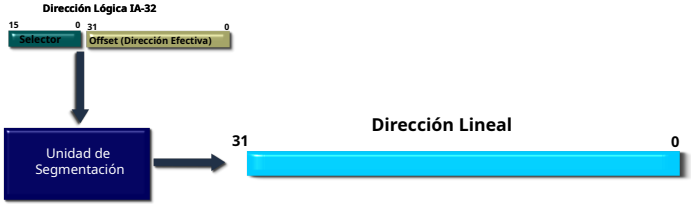
Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.
- El procesador necesita conocer para cada tarea, la dirección física en donde se inicia la estructura de Descriptores de Páginas.
- En estos procesador el Registro de Control **CR3** contiene esta dirección (***dirección física***).
- En el próximo slide se muestra el proceso que describiremos a continuación.
- El procesador toma la ***dirección lineal*** y la divide en tres campos de bits que serán utilizados respectivamente como:
  - Índice en el **PD**, para determinar la dirección de inicio y derechos de acceso de la página que contiene la **PT** uno de cuyos descriptores corresponde a la página que se está buscando acceder en memoria física.
  - Índice en la **PT** que permitiré encontrar el descriptor de la página de memoria física que se está direccionando.

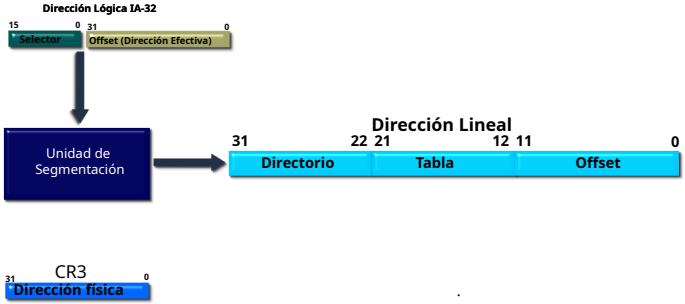
Traducción con Páginas de 4 Kbytes en 32 bits

- Es el modo original de paginación del procesador 80386.
- El procesador necesita conocer para cada tarea, la dirección física en donde se inicia la estructura de Descriptores de Páginas.
- En estos procesador el Registro de Control **CR3** contiene esta dirección (***dirección física***).
- En el próximo slide se muestra el proceso que describiremos a continuación.
- El procesador toma la ***dirección lineal*** y la divide en tres campos de bits que serán utilizados respectivamente como:
  - Índice en el **PD**, para determinar la dirección de inicio y derechos de acceso de la página que contiene la **PT** uno de cuyos descriptores corresponde a la página que se está buscando acceder en memoria física.
  - Índice en la **PT** que permitiré encontrar el descriptor de la página de memoria física que se está direccionando.
  - Desplazamiento relativo al comienzo de la página en la que se encuentra la variable o el código que se está direccionando.

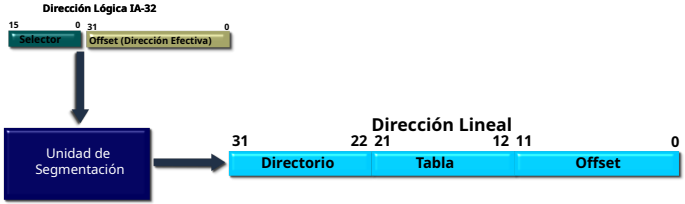
Modo 32 bits: Traducción con Páginas de 4 Kbytes



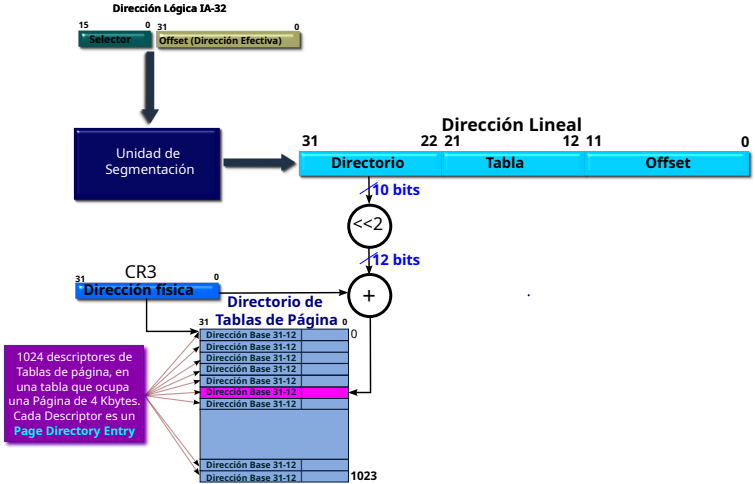
Modo 32 bits: Traducción con Páginas de 4 Kbytes



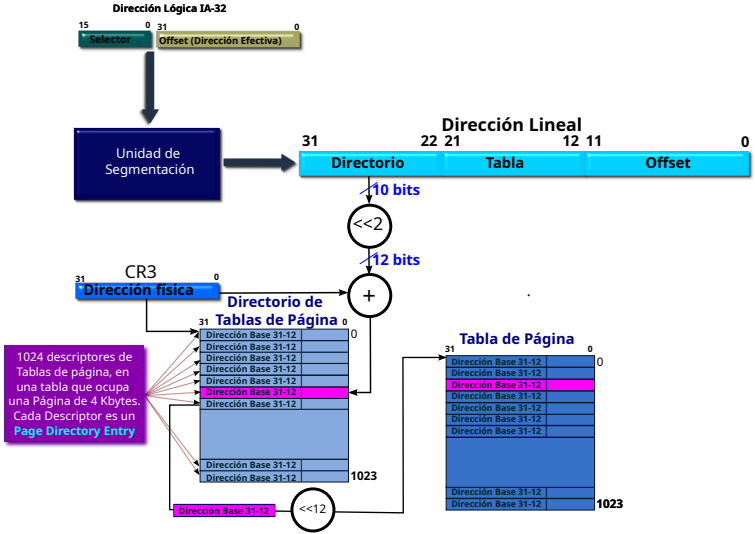
Modo 32 bits: Traducción con Páginas de 4 Kbytes



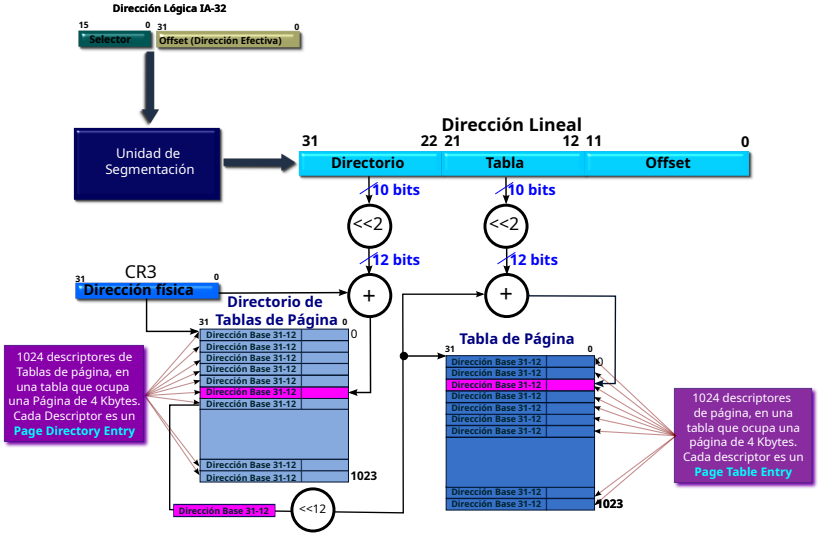
Modo 32 bits: Traducción con Páginas de 4 Kbytes



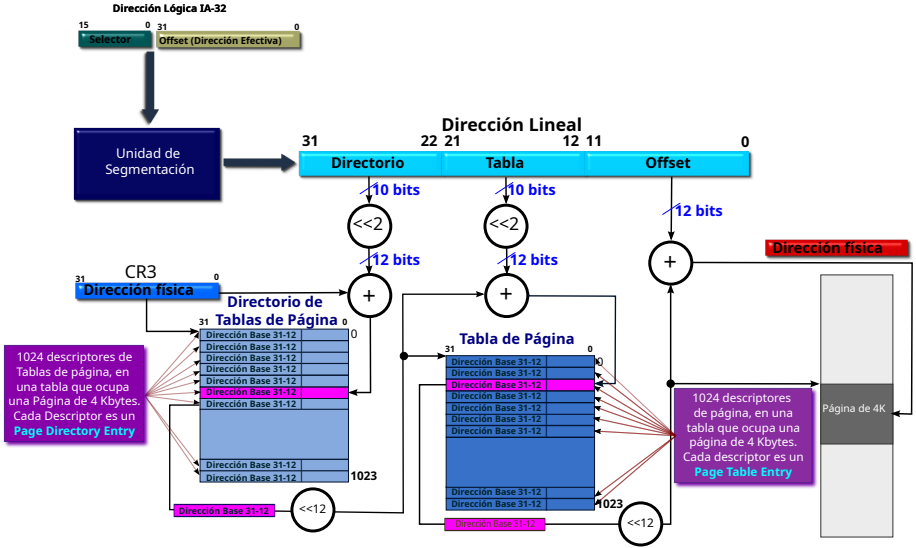
Modo 32 bits: Traducción con Páginas de 4 Kbytes



Modo 32 bits: Traducción con Páginas de 4 Kbytes



Modo 32 bits: Traducción con Páginas de 4 Kbytes



Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Como el procesador trabaja con Páginas de 4 Kbytes, las tablas se acomodarán en bloques del mismo tamaño.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Como el procesador trabaja con Páginas de 4 Kbytes, las tablas se acomodarán en bloques del mismo tamaño.
- Al ser cada descriptor de página de 32 bits cada tabla podrá de este modo almacenar 1024 descriptores.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Como el procesador trabaja con Páginas de 4 Kbytes, las tablas se acomodarán en bloques del mismo tamaño.
- Al ser cada descriptor de página de 32 bits cada tabla podrá de este modo almacenar 1024 descriptores.
- Así tendremos que para ingresar a cada **PT** el procesador necesitará 10 bits ($2^{10} = 1024$).

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Como el procesador trabaja con Páginas de 4 Kbytes, las tablas se acomodarán en bloques del mismo tamaño.
- Al ser cada descriptor de página de 32 bits cada tabla podrá de este modo almacenar 1024 descriptores.
- Así tendremos que para ingresar a cada **PT** el procesador necesitará 10 bits ($2^{10} = 1024$).
- Los 10 bits mas significativos de la ***dirección lineal*** se usan como índice en el **PD**.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Como el procesador trabaja con Páginas de 4 Kbytes, las tablas se acomodarán en bloques del mismo tamaño.
- Al ser cada descriptor de página de 32 bits cada tabla podrá de este modo almacenar 1024 descriptores.
- Así tendremos que para ingresar a cada **PT** el procesador necesitará 10 bits ($2^{10} = 1024$).
- Los 10 bits mas significativos de la ***dirección lineal*** se usan como índice en el **PD**.
- El descriptor seleccionado (denominado **PDE**, por Page Directory Entry), contiene los 20 bits mas significativos de la dirección física de la página que contiene la **PT**.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Esta Tabla es la que contendrá finalmente el descriptor de la página de memoria direccionada en la MMU del procesador.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Esta Tabla es la que contendrá finalmente el descriptor de la página de memoria direccionada en la MMU del procesador.
- Una vez leída la **PDE** y extraída la dirección física donde comienza la **PT**, los siguientes 10 bits de la dirección lineal se utilizan como índice en ésta para acceder al descriptor de la página direccionada.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Esta Tabla es la que contendrá finalmente el descriptor de la página de memoria direccionada en la MMU del procesador.
- Una vez leída la **PDE** y extraída la dirección física donde comienza la **PT**, los siguientes 10 bits de la dirección lineal se utilizan como índice en ésta para acceder al descriptor de la página direccionada.
- Esta entrada se denomina **PTE** por Page Table Entry. En el descriptor estarán los 20 bits mas significativos de la dirección física de memoria en la que comienza la página direccionada.

Modo 32 bits: Traducción con Páginas de 4 Kbytes

- Esta Tabla es la que contendrá finalmente el descriptor de la página de memoria direccionada en la MMU del procesador.
- Una vez leída la **PDE** y extraída la dirección física donde comienza la **PT**, los siguientes 10 bits de la dirección lineal se utilizan como índice en ésta para acceder al descriptor de la página direccionada.
- Esta entrada se denomina **PTE** por Page Table Entry. En el descriptor estarán los 20 bits mas significativos de la dirección física de memoria en la que comienza la página direccionada.
- Finalmente, obtenida la dirección base de la página, queda sumarle los 12 bits menos significativos de la dirección lineal se obtiene la dirección física de memoria direccionada por el procesador.

Primeras conclusiones

Primeras conclusiones

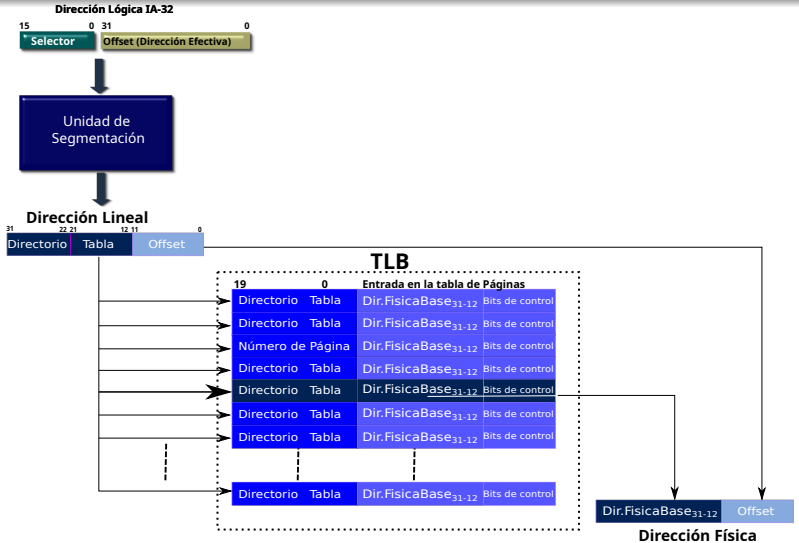
1. Del análisis de funcionamiento del mecanismo de traducción de ***dirección lineal*** a ***dirección física*** surge que no es necesario que las páginas de código o datos estén contiguas. Si una instrucción queda al final de una página, al realizarse el próximo ciclo de fetch, el sistema de Paginación generará un par de entradas **PDE** - **PTE** diferentes de la instrucción anterior de modo que el programa ejecutará de manera transparente a la ubicación del código en memoria.

Primeras conclusiones

1. Del análisis de funcionamiento del mecanismo de traducción de ***dirección lineal*** a ***dirección física*** surge que no es necesario que las páginas de código o datos estén contiguas. Si una instrucción queda al final de una página, al realizarse el próximo ciclo de fetch, el sistema de Paginación generará un par de entradas **PDE** - **PTE** diferentes de la instrucción anterior de modo que el programa ejecutará de manera transparente a la ubicación del código en memoria.

2. Por cada opcode fetch, memory read, o memory write, es necesario acceder a memoria dos veces para leer el **PDE** y el **PTE** y luego hacer sumas y demás. Al igual que en el caso de la segmentación en donde el procesador posee un cache asociado a cada registro de segmento para leer una sola vez el descriptor, en la Unidad de Paginación existe un cache de traducciones: el Traslation Lookaside Buffer (o TLB)

Translation Lookaside Buffer (TLB)



Translation Lookaside Buffer (TLB)

- Se almacenan solo las partes de interés de las direcciones lineal y física.

Translation Lookaside Buffer (TLB)

- Se almacenan solo las partes de interés de las direcciones lineal y física.
- Mas específicamente, los bits de la ***dirección lineal*** que se utilizan para acceder al **PD**, y a la **PT**, y los bits de la ***dirección física*** contenidos en el descriptor de la página direccionada, es decir, 31 a 12.

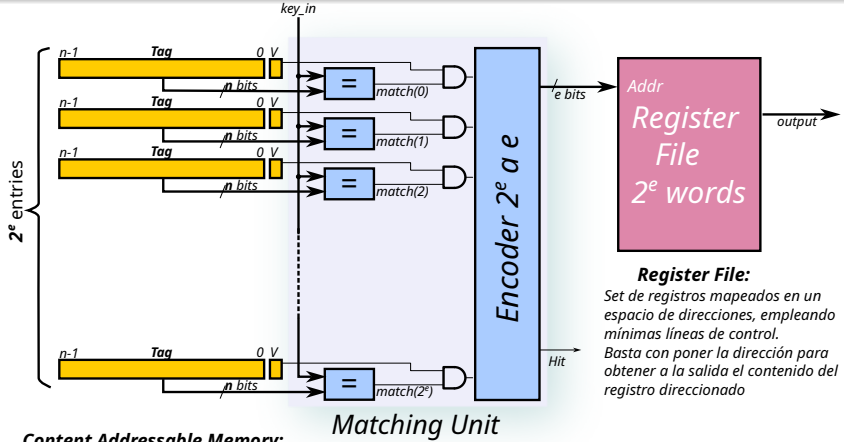
Translation Lookaside Buffer (TLB)

- Se almacenan solo las partes de interés de las direcciones lineal y física.
- Mas específicamente, los bits de la ***dirección lineal*** que se utilizan para acceder al **PD**, y a la **PT**, y los bits de la ***dirección física*** contenidos en el descriptor de la página direccionada, es decir, 31 a 12.
- El resto no son relevantes a los efectos de la traducción, sin embargo se deben contemplar bits para la validación de permisos.

Translation Lookaside Buffer (TLB)

- Se almacenan solo las partes de interés de las direcciones lineal y física.
- Mas específicamente, los bits de la ***dirección lineal*** que se utilizan para acceder al **PD**, y a la **PT**, y los bits de la ***dirección física*** contenidos en el descriptor de la página direccionada, es decir, 31 a 12.
- El resto no son relevantes a los efectos de la traducción, sin embargo se deben contemplar bits para la validación de permisos.
- El Buffer de Traducciones actúa como una pequeña memoria cache, almacenando los pares de valores indicados, para las últimas traducciones, ya que una vez traducida la dirección de una página, si ésta contiene código el procesador seguirá ejecutando dentro de esa página de modo que recurrirá asiduamente a esa entrada del TLB. El mismo comportamiento se verifica para variables, buffers y demás elementos de un trozo de software. Se denomina Vecindad.

Content Addressable Memory



Register File:

Set de registros mapeados en un espacio de direcciones, empleando mínimas líneas de control.
Basta con poner la dirección para obtener a la salida el contenido del registro direccionado

Content Addressable Memory:

En lugar de asociar una dirección a cada registro del Register File, le asigna una etiqueta (Tag). Para seleccionar un registro Solo se requiere ingresar su etiqueta.
La Matching Unit se encarga de comparar cada etiqueta de la CAM con la ingresada. Solo una será (eventualmente) TRUE. Sino se genera un MISS.
Cualquier parecido con el Cache directory de un Cache Controller NO es coincidencia. Es la misma situación. Salvo que en el caso del Cache Directory por lo general ***e=n****.*

Translation Lookaside Buffer (TLB)

- Los bits de control por lo general son los atributos de la página seleccionada, mas otros que no están documentados, pero siendo un cache, imaginamos mínimamente bits LRU para determinar cual es el elemento que debe ser desalojado al necesitarse una entrada para almacenar una nueva traducción estando el TLB lleno.

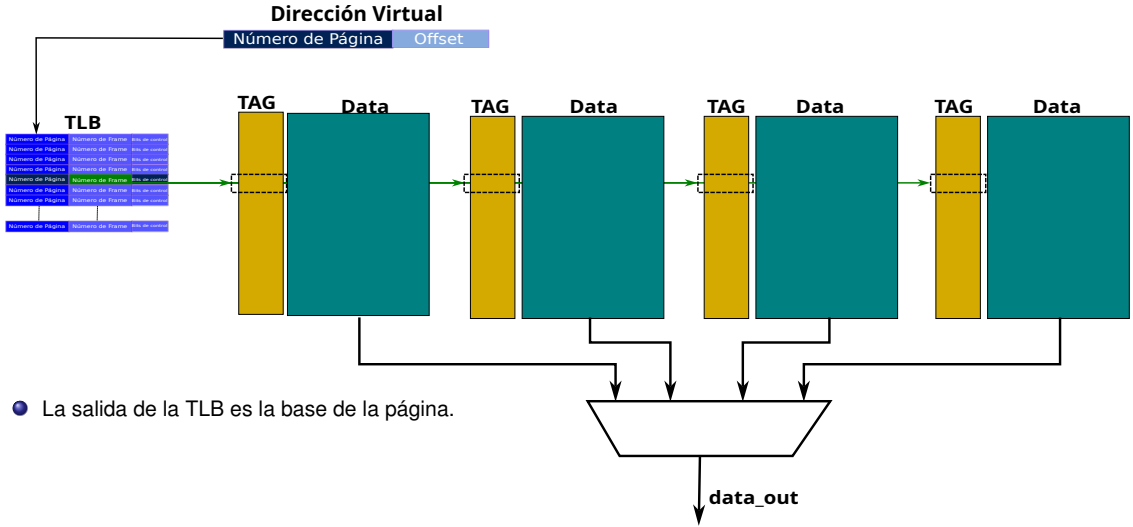
Translation Lookaside Buffer (TLB)

- Los bits de control por lo general son los atributos de la página seleccionada, mas otros que no están documentados, pero siendo un cache, imaginamos mínimamente bits LRU para determinar cual es el elemento que debe ser desalojado al necesitarse una entrada para almacenar una nueva traducción estando el TLB lleno.
- El tamaño del TLB ha ido creciendo conforme se desarrollaron nuevos procesadores. Actualmente se tiene uno para código y otro para datos ubicados en la cache L1 de código y datos del procesador.

Translation Lookaside Buffer (TLB)

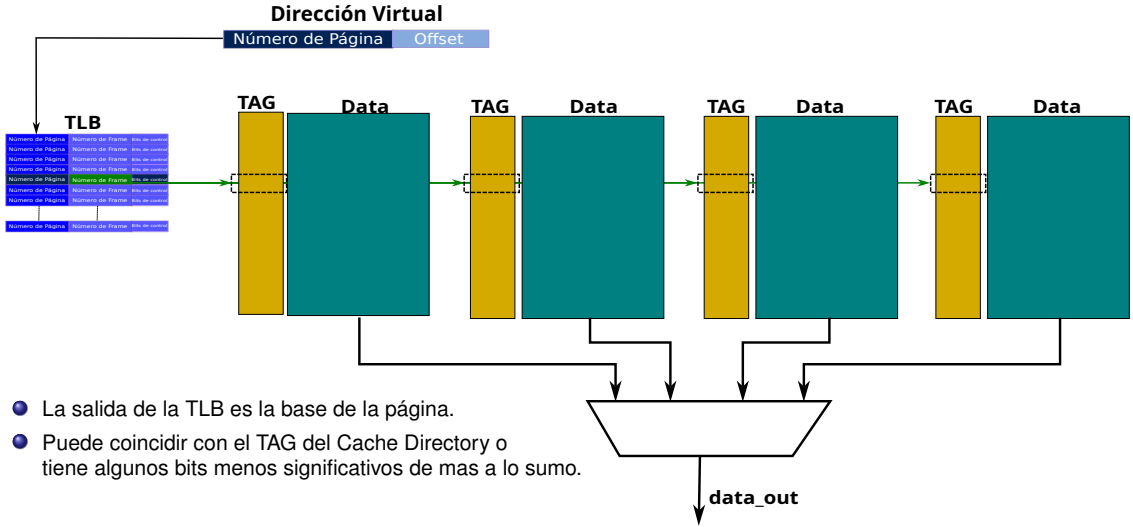
- Los bits de control por lo general son los atributos de la página seleccionada, mas otros que no están documentados, pero siendo un cache, imaginamos mínimamente bits LRU para determinar cual es el elemento que debe ser desalojado al necesitarse una entrada para almacenar una nueva traducción estando el TLB lleno.
- El tamaño del TLB ha ido creciendo conforme se desarrollaron nuevos procesadores. Actualmente se tiene uno para código y otro para datos ubicados en la cache L1 de código y datos del procesador.
- Por otra parte como la paginación se organiza en base a tareas, cada tarea tendrá su propia estructura con lo cual el registro **CR3** cambiará de una tarea a otra. La escritura de un valor en el registro **CR3** flushea el contenido de la TLB (Excepto aquellas entradas que, como veremos, se setean como Globales)

TLB: Relación con la Cache Memory



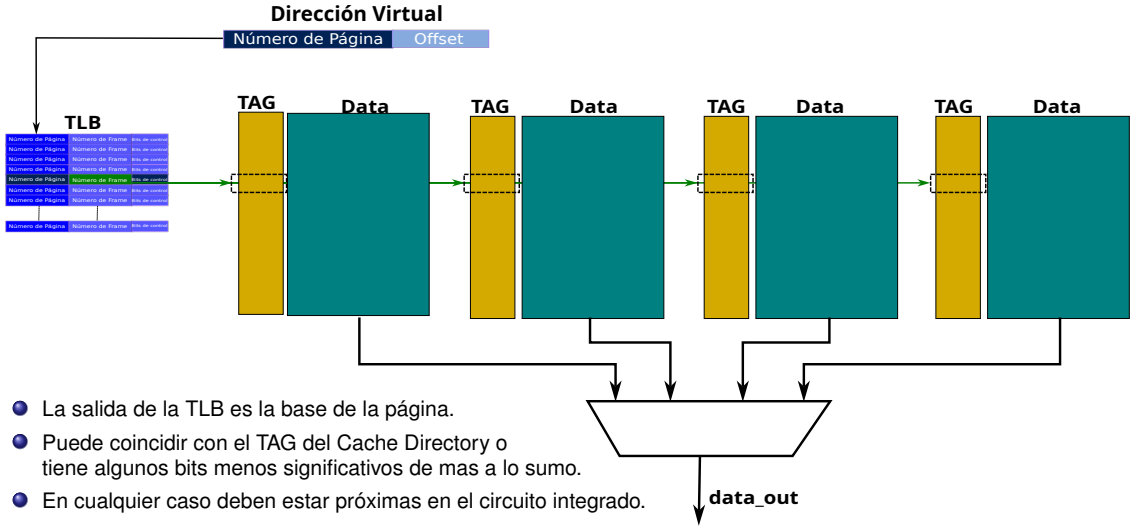
- La salida de la TLB es la base de la página.

TLB: Relación con la Cache Memory



- La salida de la TLB es la base de la página.
- Puede coincidir con el TAG del Cache Directory o tiene algunos bits menos significativos de mas a lo sumo.

TLB: Relación con la Cache Memory



- La salida de la TLB es la base de la página.
- Puede coincidir con el TAG del Cache Directory o tiene algunos bits menos significativos de mas a lo sumo.
- En cualquier caso deben estar próximas en el circuito integrado.

Temario

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

- Los bits 31 a 12 del registro contienen el page frame del (**PD**).

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

- Los bits 31 a 12 del registro contienen el page frame del (**PD**).
- Esta dirección es física. O sea que el registro **CR3** contiene la dirección de RAM en donde comienza el **PD**.

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

- Los bits 31 a 12 del registro contienen el page frame del (**PD**).
- Esta dirección es física. O sea que el registro **CR3** contiene la dirección de RAM en donde comienza el **PD**.
- En modo 64 bits, este registro tiene 64 bits. En el modo 32 bits, los bits 32 a 63 se ignoran.

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

- Los bits 31 a 12 del registro contienen el page frame del (**PD**).
- Esta dirección es física. O sea que el registro **CR3** contiene la dirección de RAM en donde comienza el **PD**.
- En modo 64 bits, este registro tiene 64 bits. En el modo 32 bits, los bits 32 a 63 se ignoran.
- Bits de control de cache en tiempo de traducción.

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

- Los bits 31 a 12 del registro contienen el page frame del (**PD**).
- Esta dirección es física. O sea que el registro **CR3** contiene la dirección de RAM en donde comienza el **PD**.
- En modo 64 bits, este registro tiene 64 bits. En el modo 32 bits, los bits 32 a 63 se ignoran.
- Bits de control de cache en tiempo de traducción.

  PWT **P**age-Level **W**rite **T**hrough. Establece el modo de escritura que tendrá la página en el Cache.

CR3: Descriptor de la Página que contiene el PD

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del Directorio de Páginas																				Ignorados							PCD	PWT	Ignorados		

- Los bits 31 a 12 del registro contienen el page frame del (**PD**).
- Esta dirección es física. O sea que el registro **CR3** contiene la dirección de RAM en donde comienza el **PD**.
- En modo 64 bits, este registro tiene 64 bits. En el modo 32 bits, los bits 32 a 63 se ignoran.
- Bits de control de cache en tiempo de traducción.
  - PWT **P**age-Level **W**rite **T**hrough. Establece el modo de escritura que tendrá la página en el Cache.
  - PCD **P**age-Level **C**ache **D**isable. Establece que una página integre el tipo de memoria no cacheable.

Descriptores en el Directorio de Tablas de Páginas

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de Página																				Ignorados				PS	IGN	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Tabla de Páginas de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bits 31 a 22 de la Dirección física del frame de la Página de 4M										Reservados deben ser 0					Bits 36 a 32 de la Dirección				PAT	Ignorados			G	PS	D	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Página de 4 Mbytes (PSE-36 activa)

Descriptores en el Directorio de Tablas de Páginas

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de Página																				Ignorados				PS	IGN	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Tabla de Páginas de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bits 31 a 22 de la Dirección física del frame de la Página de 4M										Reservados deben ser 0					Bits 36 a 32 de la Dirección				PAT	Ignorados			G	PS	D	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Página de 4 Mbytes (PSE-36 activa)

- Ambos pueden coexistir en la misma tabla.

Descriptores en el Directorio de Tablas de Páginas

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de Página																				Ignorados				PS	IGN	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Tabla de Páginas de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bits 31 a 22 de la Dirección física del frame de la Página de 4M										Reservados deben ser 0					Bits 36 a 32 de la Dirección				PAT	Ignorados			G	PS	D	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Página de 4 Mbytes (PSE-36 activa)

- Ambos pueden coexistir en la misma tabla.
- En el primer descriptor, los bits 31 a 12 son los bits de la dirección física (el Page Frame) de la página que contiene la (**PT**).

Descriptores en el Directorio de Tablas de Páginas

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de Página																				Ignorados				PS	IGN	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Tabla de Páginas de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bits 31 a 22 de la Dirección física del frame de la Página de 4M										Reservados deben ser 0					Bits 36 a 32 de la Dirección				PAT	Ignorados			G	PS	D	A	PCD	PWT	U/S	R/W	P

Figura: Descriptor de la Página de 4 Mbytes (PSE-36 activa)

- Ambos pueden coexistir en la misma tabla.
- En el primer descriptor, los bits 31 a 12 son los bits de la dirección física (el Page Frame) de la página que contiene la (**PT**).
- Una de las 1024 **PTE** de esa tabla describe la página que contiene la dirección de memoria solicitada por el procesador.

Formato del descriptor de una Página de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del frame de la Página de 4 Kbytes																				Ignorados			G	PAT	D	A	PCD	PWT	U/S	R/W	P

Formato del descriptor de una Página de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del frame de la Página de 4 Kbytes																				Ignorados			G	PAT	D	A	PCD	PWT	U/S	R/W	P

- Hay entre 1 y 1024 de estos descriptores en cada una de las **PT** definidas en el **PD**.

Formato del descriptor de una Página de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del frame de la Página de 4 Kbytes																				Ignorados			G	PAT	D	A	PCD	PWT	U/S	R/W	P

- Hay entre 1 y 1024 de estos descriptores en cada una de las **PT** definidas en el **PD**.
- Es el último nivel de traducción.

Formato del descriptor de una Página de 4 Kbytes

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del frame de la Página de 4 Kbytes																				Ignorados			G	PAT	D	A	PCD	PWT	U/S	R/W	P

- Hay entre 1 y 1024 de estos descriptores en cada una de las **PT** definidas en el **PD**.
- Es el último nivel de traducción.
- Los bits 31 a 12, son el Page Frame de la página de memoria, es decir los 20 bits mas significativos de la dirección física donde ésta comienza (los 12 menos significativos son ’0’).

Atributos de los descriptores

Son los mismos para todos los descriptores vistos y tienen el mismo significado.

Atributos de los descriptores

Son los mismos para todos los descriptores vistos y tienen el mismo significado.

PS **P**age **S**ize: Existe solo en el **PD**. Si es '0' la **PDE** corresponde a una **PT** de 4 Kbytes, y si es '1' a una página de 4 Mbytes.

Atributos de los descriptores

Son los mismos para todos los descriptores vistos y tienen el mismo significado.

PS **P**age **S**ize: Existe solo en el **PD**. Si es '0' la **PDE** corresponde a una **PT** de 4 Kbytes, y si es '1' a una página de 4 Mbytes.

PAT **P**age **A**ttribute **T**able: Los MSRs **MTRR** definen y controlan el cache de diferentes zonas de memoria. A partir del Procesador Pentium III se dispone de este atributo que los complementa en base a páginas.

Atributos de los descriptores

Son los mismos para todos los descriptores vistos y tienen el mismo significado.

PS **P**age **S**ize: Existe solo en el **PD**. Si es '0' la **PDE** corresponde a una **PT** de 4 Kbytes, y si es '1' a una página de 4 Mbytes.

PAT **P**age **A**ttribute **T**able: Los MSRs **MTRR** definen y controlan el cache de diferentes zonas de memoria. A partir del Procesador Pentium III se dispone de este atributo que los complementa en base a páginas.

G **G**lobal: Si **CR4.PGE = 1** Tiene efecto la activación de la funcionalidad Global, que otorga ese caracter a la traducción de esa página almacenada en la TLB. La entrada no se flushea cuando se recarga el registro **CR3**.

Atributos de los descriptores

Atributos de los descriptores

D **D**irty. Indica que la página ha sido modificada (está “sucia”). El Sistema Operativo lo inicializa en ’0’, y se setea en forma automática en cada escritura en la página. El algoritmo de swap en el momento de desalojar una página de la memoria RAM, analiza este bit y no la copia al disco si no ha sido modificada, mejorando de este modo su eficiencia. Solo tiene sentido en descriptores de Página (de 4K o 4M). En las entradas del **PD** que describen **TP** de 4 Kbytes se ignora ya que el Bit D existirá en el siguiente descriptor de la estructura.

Atributos de los descriptores

D **D**irty. Indica que la página ha sido modificada (está “sucia”). El Sistema Operativo lo inicializa en ’0’, y se setea en forma automática en cada escritura en la página. El algoritmo de swap en el momento de desalojar una página de la memoria RAM, analiza este bit y no la copia al disco si no ha sido modificada, mejorando de este modo su eficiencia. Solo tiene sentido en descriptores de Página (de 4K o 4M). En las entradas del **PD** que describen **TP** de 4 Kbytes se ignora ya que el Bit D existirá en el siguiente descriptor de la estructura.

A **A**ccedido. Se setea cada vez que la página es accedida. El Sistema Operativo puede contabilizar los accesos de modo de elaborar estadísticas de uso que permitan identificar cual es la página candidata a ser desalojada llegado el momento.

Atributos de los descriptores

Atributos de los descriptores

U/S **U**ser / **S**upervisor: Privilegio de la Página: ’0’ Supervisor (Kernel), y ’1’ Usuario. En general corresponden U/S = 0 a ***DPL*** = 00, y U/S = 1 al resto de los valores de ***DPL***. El procesador chequea el ***CPL*** del segmento de código para autorizar o no el acceso a las páginas de acuerdo a la combinación de los valores de U/S de los diferentes descriptores de su estructura.

Atributos de los descriptores

U/S **U**ser / **S**upervisor: Privilegio de la Página: ’0’ Supervisor (Kernel), y ’1’ Usuario. En general corresponden U/S = 0 a ***DPL*** = 00, y U/S = 1 al resto de los valores de ***DPL***. El procesador chequea el ***CPL*** del segmento de código para autorizar o no el acceso a las páginas de acuerdo a la combinación de los valores de U/S de los diferentes descriptores de su estructura.

R/W **R**eadable / **W**ritable: Establece si la página es Read Only (0) o si puede ser escrita (1).

Atributos de los descriptores

U/S **U**ser / **S**upervisor: Privilegio de la Página: ’0’ Supervisor (Kernel), y ’1’ Usuario. En general corresponden U/S = 0 a ***DPL*** = 00, y U/S = 1 al resto de los valores de ***DPL***. El procesador chequea el ***CPL*** del segmento de código para autorizar o no el acceso a las páginas de acuerdo a la combinación de los valores de U/S de los diferentes descriptores de su estructura.

R/W **R**eadable / **W**ritable: Establece si la página es Read Only (0) o si puede ser escrita (1).

P **P**resente: Indica si la página está en la memoria (P=1), generando una excepción #PF cuando se intenta acceder a una dirección de memoria que tiene al menos un descriptor con P=0 a lo largo de la estructura de tablas. Cuando P = 0, el resto del contenido del descriptor se ignora.

Atributos de los descriptores

U/S **U**ser / **S**upervisor: Privilegio de la Página: ’0’ Supervisor (Kernel), y ’1’ Usuario. En general corresponden U/S = 0 a ***DPL*** = 00, y U/S = 1 al resto de los valores de ***DPL***. El procesador chequea el ***CPL*** del segmento de código para autorizar o no el acceso a las páginas de acuerdo a la combinación de los valores de U/S de los diferentes descriptores de su estructura.

R/W **R**eadable / **W**ritable: Establece si la página es Read Only (0) o si puede ser escrita (1).

P **P**resente: Indica si la página está en la memoria (P=1), generando una excepción #PF cuando se intenta acceder a una dirección de memoria que tiene al menos un descriptor con P=0 a lo largo de la estructura de tablas. Cuando P = 0, el resto del contenido del descriptor se ignora.

PCD y PWT Idem registro **CR3**.

Temario

Extensiones de Memoria Física

Extensiones de Memoria Física

- Con el procesador **Pentium Pro**, Intel inaugura una línea de de procesadores IA-32 para servers, posteriormente continuada por el procesador **Xeon**. El **Pentium** por su parte queda destinado al segmento desktop .

Extensiones de Memoria Física

- Con el procesador **Pentium Pro**, Intel inaugura una línea de de procesadores IA-32 para servers, posteriormente continuada por el procesador **Xeon**. El **Pentium** por su parte queda destinado al segmento desktop .

- **Pentium Pro** introduce dos métodos para extender la capacidad de direccionamiento físico: Physical Address Extensions (PAE) y Page Size Extensions (PSE-36).

Extensiones de Memoria Física

- Con el procesador **Pentium Pro**, Intel inaugura una línea de de procesadores IA-32 para servers, posteriormente continuada por el procesador **Xeon**. El **Pentium** por su parte queda destinado al segmento desktop .

- **Pentium Pro** introduce dos métodos para extender la capacidad de direccionamiento físico: Physical Address Extensions (PAE) y Page Size Extensions (PSE-36).

- Ambos son mutuamente excluyentes y se seleccionan mediante flags del registro **CR4**.

Extensiones de Memoria Física

- Con el procesador **Pentium Pro**, Intel inaugura una línea de de procesadores IA-32 para servers, posteriormente continuada por el procesador **Xeon**. El **Pentium** por su parte queda destinado al segmento desktop .
- **Pentium Pro** introduce dos métodos para extender la capacidad de direccionamiento físico: Physical Address Extensions (PAE) y Page Size Extensions (PSE-36).
- Ambos son mutuamente excluyentes y se seleccionan mediante flags del registro **CR4**.
- PAE es en sí un modo de paginación (es el que se ha impuesto entre los desarrolladores de Sistemas Operativos). No obstante vale la pena un vistazo a la traducción basada en PSE-36.

Page Size Extensions (PSE-36)

Page Size Extensions (PSE-36)

- Se activa seteando el bit **CR4.PSE**. Es el bit 4.

Page Size Extensions (PSE-36)

- Se activa seteando el bit **CR4.PSE**. Es el bit 4.
- Debe asegurarse de tener **CR4.PAE = 0**, ya que ambos modos de extensión son mutuamente excluyentes.

Page Size Extensions (PSE-36)

- Se activa seteando el bit **CR4.PSE**. Es el bit 4.
- Debe asegurarse de tener **CR4.PAE = 0**, ya que ambos modos de extensión son mutuamente excluyentes.
- Habilita en el procesador 4 terminales de address adicionales: A_{35-32}. Esto extiende el bus de Address a 36 líneas, permitiendo manejar 64 Gbytes de memoria física.

Page Size Extensions (PSE-36)

- La limitación de este modo es que el remanente de 4 Gbytes solo puede ser manejado en páginas de 4 Mbytes de tamaño.

Page Size Extensions (PSE-36)

- La limitación de este modo es que el remanente de 4 Gbytes solo puede ser manejado en páginas de 4 Mbytes de tamaño.

- Con la evolución de esta arquitectura se activaron mas terminales de Address, y debido a la amplia aceptación de PAE, hay modelos de procesadores que no soportan PSE-36.

Page Size Extensions (PSE-36)

- La limitación de este modo es que el remanente de 4 Gbytes solo puede ser manejado en páginas de 4 Mbytes de tamaño.
- Con la evolución de esta arquitectura se activaron mas terminales de Address, y debido a la amplia aceptación de PAE, hay modelos de procesadores que no soportan PSE-36.
- Por lo tanto antes de activar PSE-36, lo recomendable es chequear previamente que el procesador soporte este modo. Esta comprobación se realiza mediante la instrucción **CPUID.01H:EDX.PSE[bit 3]**=1.

Comprobación y activación de (PSE-36)

Este código comprueba el soporte del procesador al modo PSE-36 y su activación tomando los recaudos antes mencionados.

```
1  mov     eax,1                   ;cpuid function
2  cpuid
3  test    edx, 8                  ;bit 3 en 1?
4  jz      PSE_not_supported
5  mov     eax, cr4
6  test    eax, 0x20               ;bit PAE en '1'?
7  jnz     PAE_activo
8  or      eax, 0x10               ;bit 4 a '1'
9  mov     cr4, eax                ;PSE-36 activo
```

Memoria física mas allá de 64 Gbytes

Comprobando la cantidad de memoria física

Memoria física mas allá de 64 Gbytes

Comprobando la cantidad de memoria física

- La cantidad de memoria física cambia de modelo a modelo.

Memoria física mas allá de 64 Gbytes

Comprobando la cantidad de memoria física

- La cantidad de memoria física cambia de modelo a modelo.
- La instrucción CPUID.80000008H:EAX[7:0] es la manera indicada de determinar la cantidad de terminales de address que posee el procesador.

Memoria física mas allá de 64 Gbytes

Comprobando la cantidad de memoria física

- La cantidad de memoria física cambia de modelo a modelo.
- La instrucción CPUID.80000008H:EAX[7:0] es la manera indicada de determinar la cantidad de terminales de address que posee el procesador.
- Esta cantidad es el número binario contenido en el registro **AL** (**EAX**[7:0], coincide con este registro de 8 bits) a la salida de la CPUID invocada con **EAX** = 0x80000008.

Memoria física mas allá de 64 Gbytes

Comprobando la cantidad de memoria física

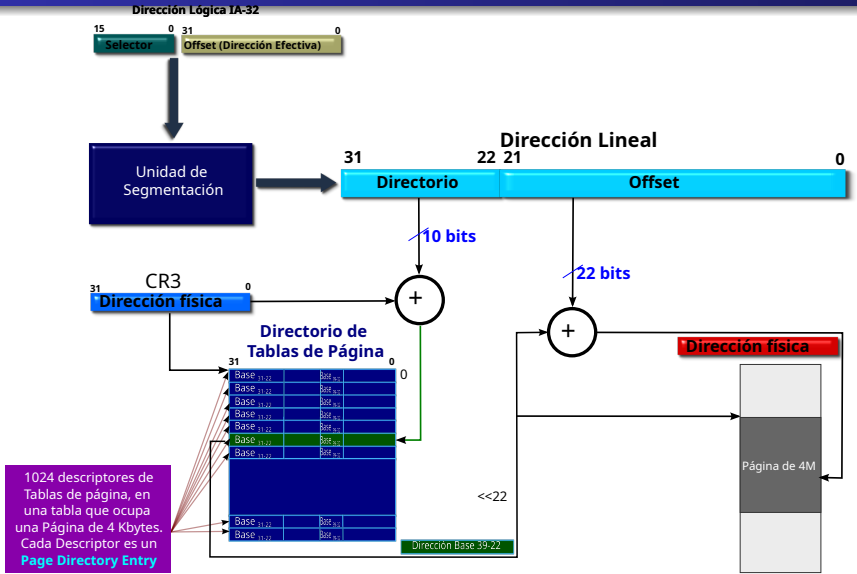
- La cantidad de memoria física cambia de modelo a modelo.
- La instrucción CPUID.80000008H:EAX[7:0] es la manera indicada de determinar la cantidad de terminales de address que posee el procesador.
- Esta cantidad es el número binario contenido en el registro **AL** (**EAX**[7:0], coincide con este registro de 8 bits) a la salida de la CPUID invocada con **EAX** = 0x80000008.
- Intel denomina a este parámetro **`MAXPHYADDR`**.

Memoria física mas allá de 64 Gbytes

Comprobando la cantidad de memoria física

- La cantidad de memoria física cambia de modelo a modelo.
- La instrucción CPUID.80000008H:EAX[7:0] es la manera indicada de determinar la cantidad de terminales de address que posee el procesador.
- Esta cantidad es el número binario contenido en el registro **AL** (**EAX**[7:0], coincide con este registro de 8 bits) a la salida de la CPUID invocada con **EAX** = 0x80000008.
- Intel denomina a este parámetro **MAXPHYADDR**.
- Si el procesador no soporta CPUID.8000008H, tiene a lo sumo 36 bits de direcciones (siempre que soporte PSE-36).

Traducción con Páginas de 4 Mbytes



Traducción con Páginas de 4 Mbytes

¿Que cambia?

Traducción con Páginas de 4 Mbytes

¿Que cambia?

- La figura anterior muestra el esquema de traducción solo con el **PD**, ya que en una página de 4 Mbytes los 22 bits menos significativos de la dirección base son ’0’.

Traducción con Páginas de 4 Mbytes

¿Que cambia?

- La figura anterior muestra el esquema de traducción solo con el **PD**, ya que en una página de 4 Mbytes los 22 bits menos significativos de la dirección base son '0'.
- Por lo tanto los 10 bits que en el sistema de 4 Kbytes se utilizaban como índice para la **PT** ahora no se necesitan. Se aprovecha para alojar allí los bits A_{39-32} de la dirección física.

Traducción con Páginas de 4 Mbytes

¿Que cambia?

- La figura anterior muestra el esquema de traducción solo con el **PD**, ya que en una página de 4 Mbytes los 22 bits menos significativos de la dirección base son '0'.
- Por lo tanto los 10 bits que en el sistema de 4 Kbytes se utilizaban como índice para la **PT** ahora no se necesitan. Se aprovecha para alojar allí los bits A_{39-32} de la dirección física.
- La cantidad significativa surge de la instrucción CPUID.80000008H.

Traducción con Páginas de 4 Mbytes

Precauciones

Traducción con Páginas de 4 Mbytes

Precauciones

- Se sigue teniendo descriptores de 32 bits, ya que los bits adicionales de address se alojan en el espacio dejado por los bits A_{21-12}, que en una página de 4 Mbytes son siempre '0'.

Traducción con Páginas de 4 Mbytes

Precauciones

- Se sigue teniendo descriptores de 32 bits, ya que los bits adicionales de address se alojan en el espacio dejado por los bits A_{21-12}, que en una página de 4 Mbytes son siempre '0'.
- Cabe destacar que es posible diseñar un sistema de paginación mixto con páginas de 4 Kbytes y de 4 Mbytes. A continuación veremos como son los descriptores de página en detalle para cada caso y cuales son las limitaciones.

Temario

Características del Modo PAE

Características del Modo PAE

- Introducido como hemos dicho por el procesador Pentium Pro.

Características del Modo PAE

- Introducido como hemos dicho por el procesador Pentium Pro.
- Permite trabajar con páginas de 4 Kbytes o de 2 Mbytes, pudiendo particionar con cualquiera de los tamaños de páginas o incluso combinando ambos tamaños, el espacio físico completo.

Características del Modo PAE

- Introducido como hemos dicho por el procesador Pentium Pro.
- Permite trabajar con páginas de 4 Kbytes o de 2 Mbytes, pudiendo particionar con cualquiera de los tamaños de páginas o incluso combinando ambos tamaños, el espacio físico completo.
- PAE evolucionó con la arquitectura y en la actualidad permite trabajar hasta con 52 bits de memoria física (hasta 4 Pbyte de RAM).

Características del Modo PAE

- Introducido como hemos dicho por el procesador Pentium Pro.
- Permite trabajar con páginas de 4 Kbytes o de 2 Mbytes, pudiendo particionar con cualquiera de los tamaños de páginas o incluso combinando ambos tamaños, el espacio físico completo.
- PAE evolucionó con la arquitectura y en la actualidad permite trabajar hasta con 52 bits de memoria física (hasta 4 Pbyte de RAM).
- Recordemos que la cantidad de bits de memoria Física se obtiene como resultado de CPUID.0x80000008, en el registro **AL**, y se denomina **`MAXPHYADDR`**.

Características del Modo PAE

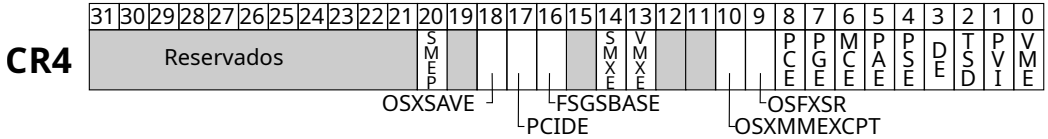
- Introducido como hemos dicho por el procesador Pentium Pro.
- Permite trabajar con páginas de 4 Kbytes o de 2 Mbytes, pudiendo particionar con cualquiera de los tamaños de páginas o incluso combinando ambos tamaños, el espacio físico completo.
- PAE evolucionó con la arquitectura y en la actualidad permite trabajar hasta con 52 bits de memoria física (hasta 4 Pbyte de RAM).
- Recordemos que la cantidad de bits de memoria Física se obtiene como resultado de CPUID.0x80000008, en el registro **AL**, y se denomina **MAXPHYADDR**.
- Si **MAXPHYADDR** $<$52, los bits 51 y consecutivos en orden descendente, hasta el que corresponda al peso **MAXPHYADDR**, deben estar en '0'.

Características del Modo PAE

- Introducido como hemos dicho por el procesador Pentium Pro.
- Permite trabajar con páginas de 4 Kbytes o de 2 Mbytes, pudiendo particionar con cualquiera de los tamaños de páginas o incluso combinando ambos tamaños, el espacio físico completo.
- PAE evolucionó con la arquitectura y en la actualidad permite trabajar hasta con 52 bits de memoria física (hasta 4 Pbyte de RAM).
- Recordemos que la cantidad de bits de memoria Física se obtiene como resultado de CPUID.0x80000008, en el registro **AL**, y se denomina **MAXPHYADDR**.
- Si **MAXPHYADDR** <52, los bits 51 y consecutivos en orden descendente, hasta el que corresponda al peso **MAXPHYADDR**, deben estar en '0'.
- Es la base del modo de paginación IA-32e.

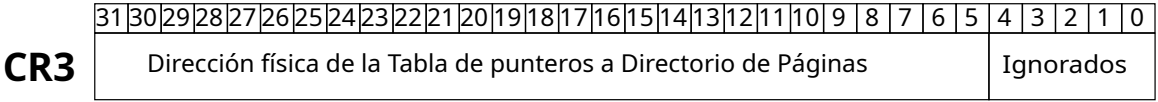
Activación del Modo PAE

PAE se activa haciendo **CR4.PAE = 1**. Hemos nombrado varios bits del registro **CR4**. A continuación el layout de este registro.



Una vez activado, PAE permite, traducir una ***dirección lineal*** de 32 bits en una ***dirección física*** de 52 bits.
Observar que de todos modos accedemos a un espacio lineal de 4 Gbytes tope en cada momento.

Estructura de traducción del Modo PAE



Estructura de traducción del Modo PAE

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de punteros a Directorio de Páginas																											Ignorados					

- En PAE el procesador arma una estructura jerárquica de tres niveles.

Estructura de traducción del Modo PAE

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de punteros a Directorio de Páginas																											Ignorados					

- En PAE el procesador arma una estructura jerárquica de tres niveles.
- La base es siempre **CR3** (que como vemos difiere del layout para el modo 32 bits)

Estructura de traducción del Modo PAE

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de punteros a Directorio de Páginas																											Ignorados					

- En PAE el procesador arma una estructura jerárquica de tres niveles.
- La base es siempre **CR3** (que como vemos difiere del layout para el modo 32 bits)
- El primer nivel es un array de cuatro punteros a Directorios de Tablas de Páginas.

Estructura de traducción del Modo PAE

CR3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física de la Tabla de punteros a Directorio de Páginas																											Ignorados					

- En PAE el procesador arma una estructura jerárquica de tres niveles.
- La base es siempre **CR3** (que como vemos difiere del layout para el modo 32 bits)
- El primer nivel es un array de cuatro punteros a Directorios de Tablas de Páginas.
- El registro **CR3** contendrá la dirección base de esta tabla de 4 punteros, **la cual debe estar alineada a 32 bytes**. Esta condición debe ser respetada **o el procesador generará una excepción**.

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

- La Tabla de Punteros de Directorio de Página (de ahora en mas **PDPT**), apuntada por **CR3**, consta de 4 entradas denominadas **PDPTE** (por Page Directory Pointer Table Entry)

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

- La Tabla de Punteros de Directorio de Página (de ahora en mas **PDPT**), apuntada por **CR3**, consta de 4 entradas denominadas **PDPTE** (por Page Directory Pointer Table Entry)
- Cada **PDPTE** maneja el acceso a 1 Gbyte de memoria lineal.

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

- La Tabla de Punteros de Directorio de Página (de ahora en mas **PDPT**), apuntada por **CR3**, consta de 4 entradas denominadas **PDPTE** (por Page Directory Pointer Table Entry)
- Cada **PDPTE** maneja el acceso a 1 Gbyte de memoria lineal.
- El procesador mantiene un set de 4 registros internos (no arquitecturales): **PDPTE0**, **PDPTE1**, **PDPTE2**, y **PDPTE3**.

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

- La Tabla de Punteros de Directorio de Página (de ahora en mas **PDPT**), apuntada por **CR3**, consta de 4 entradas denominadas **PDPTE** (por Page Directory Pointer Table Entry)
- Cada **PDPTE** maneja el acceso a 1 Gbyte de memoria lineal.
- El procesador mantiene un set de 4 registros internos (no arquitecturales): **PDPTE0**, **PDPTE1**, **PDPTE2**, y **PDPTE3**.
- Estos registros mantienen el puntero a su respectivo **PD** asociado, para agilizar la traducción de direcciones, tal como un cache.

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

- La Tabla de Punteros de Directorio de Página (de ahora en mas **PDPT**), apuntada por **CR3**, consta de 4 entradas denominadas **PDPTE** (por Page Directory Pointer Table Entry)
- Cada **PDPTE** maneja el acceso a 1 Gbyte de memoria lineal.
- El procesador mantiene un set de 4 registros internos (no arquitecturales): **PDPTE0**, **PDPTE1**, **PDPTE2**, y **PDPTE3**.
- Estos registros mantienen el puntero a su respectivo **PD** asociado, para agilizar la traducción de direcciones, tal como un cache.
- Se recargan desde sus respectivas copias en la memoria RAM cada vez que ésta altera su valor.

Estructura de traducción del Modo PAE

Tabla de Punteros de Directorio de Página

- La Tabla de Punteros de Directorio de Página (de ahora en mas **PDPT**), apuntada por **CR3**, consta de 4 entradas denominadas **PDPTE** (por Page Directory Pointer Table Entry)
- Cada **PDPTE** maneja el acceso a 1 Gbyte de memoria lineal.
- El procesador mantiene un set de 4 registros internos (no arquitecturales): **PDPTE0**, **PDPTE1**, **PDPTE2**, y **PDPTE3**.
- Estos registros mantienen el puntero a su respectivo **PD** asociado, para agilizar la traducción de direcciones, tal como un cache.
- Se recargan desde sus respectivas copias en la memoria RAM cada vez que ésta altera su valor.
- El sistema trabaja con la versión RAM de la **PDTP**, el procesador debe releerlas para mantener actualizada esta cache interna.

Cuando se regargan los 4 registros

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

  CR0.CD **C**ache **D**isable. ’1’ deshabilita el cache del procesador.

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

  CR0.CD **C**ache **D**isable. '1' deshabilita el cache del procesador.

  CR0.NW **N**on_**W**rite Through. Establece globalmente esta política de escritura. Solo tiene sentido si **CR0.CD** = 0.

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

  CR0.CD **C**ache **D**isable. '1' deshabilita el cache del procesador.

  CR0.NW **N**on_**W**rite Through. Establece globalmente esta política de escritura. Solo tiene sentido si **CR0.CD** = 0.

  CR0.PG Activa Paginación.

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

  CR0.CD **C**ache **D**isable. '1' deshabilita el cache del procesador.

  CR0.NW **N**on_**W**rite Through. Establece globalmente esta política de escritura. Solo tiene sentido si **CR0.CD** = 0.

  CR0.PG Activa Paginación.

  CR4.PAE Activa el modo PAE.

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

  CR0.CD **C**ache **D**isable. '1' deshabilita el cache del procesador.

  CR0.NW **N**on_**W**rite Through. Establece globalmente esta política de escritura. Solo tiene sentido si **CR0.CD** = 0.

  CR0.PG Activa Paginación.

  CR4.PAE Activa el modo PAE.

  CR4.PGE Permite definir el uso de Páginas globales que se habilitan una a una con el bit **PG** de DPTE.

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

  CR0.CD **C**ache **D**isable. '1' deshabilita el cache del procesador.

  CR0.NW **N**on_**W**rite Through. Establece globalmente esta política de escritura. Solo tiene sentido si **CR0.CD** = 0.

  CR0.PG Activa Paginación.

  CR4.PAE Activa el modo PAE.

  CR4.PGE Permite definir el uso de Páginas globales que se habilitan una a una con el bit **PG** de DPTE.

  CR4.PSE Habilita o deshabilita el modo PSE-36

Cuando se regargan los 4 registros

- Si se ejecuta una instrucción cuyo operando destino es el registro **CR0**, o el registro **CR4** y como resultado de la cual se modifica uno o mas de los bits:

CR0.CD **C**ache **D**isable. '1' deshabilita el cache del procesador.

CR0.NW **N**on_**W**rite Through. Establece globalmente esta política de escritura. Solo tiene sentido si **CR0.CD** = 0.

CR0.PG Activa Paginación.

CR4.PAE Activa el modo PAE.

CR4.PGE Permite definir el uso de Páginas globales que se habilitan una a una con el bit **PG** de DPTE.

CR4.PSE Habilita o deshabilita el modo PSE-36

CR4.SMEP Habilita un modo de protección denominado **S**upervisor **M**ode **E**xcecution **P**revention.

Cuando se regargan los 4 registros

- Cuando se modifica el registro **CR3**, ya que debe buscarse de memoria el nuevo set de cuatro punteros que gestionará el espacio lineal de 4 Gbytes.

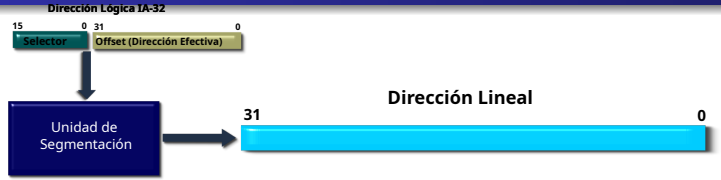
Cuando se regargan los 4 registros

- Cuando se modifica el registro **CR3**, ya que debe buscarse de memoria el nuevo set de cuatro punteros que gestionará el espacio lineal de 4 Gbytes.
- Tal como veremos al abordar el manejo de Tareas, el registro **CR3** forma parte del contexto de ejecución que el procesador salvará y recuperará cuando llegue el momento.

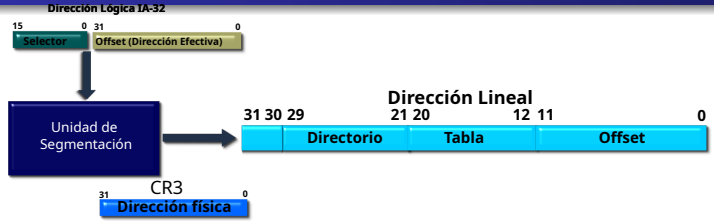
Cuando se regargan los 4 registros

- Cuando se modifica el registro **CR3**, ya que debe buscarse de memoria el nuevo set de cuatro punteros que gestionará el espacio lineal de 4 Gbytes.
- Tal como veremos al abordar el manejo de Tareas, el registro **CR3** forma parte del contexto de ejecución que el procesador salvará y recuperará cuando llegue el momento.
- Algunas instrucciones de las extenciones de virtualización (VMX) motivarán también la carga de estos registros por parte del procesador.

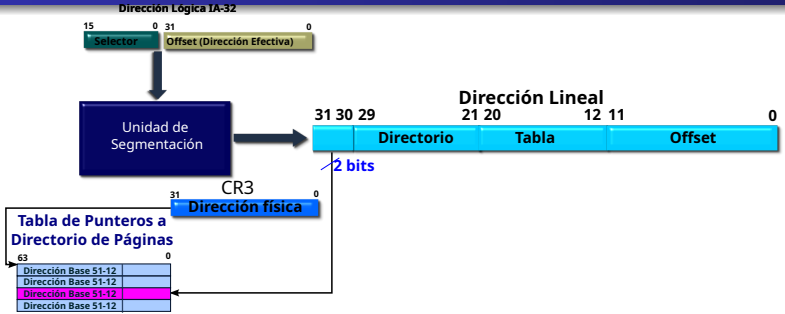
PAE: Traducción



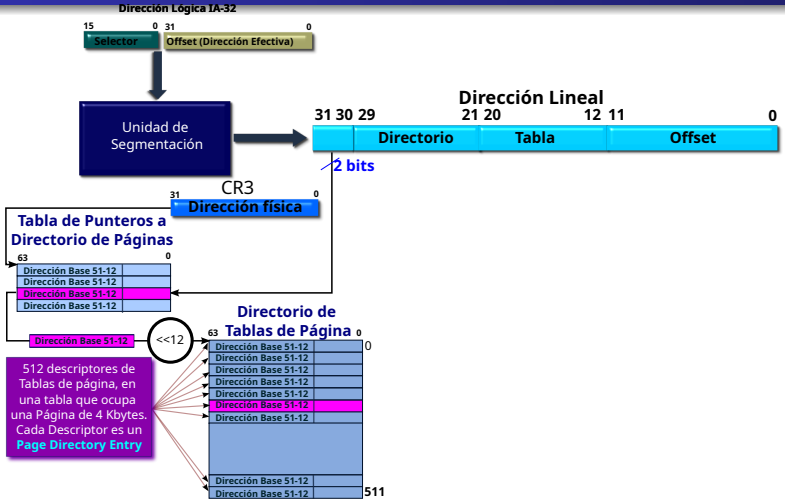
PAE: Traducción



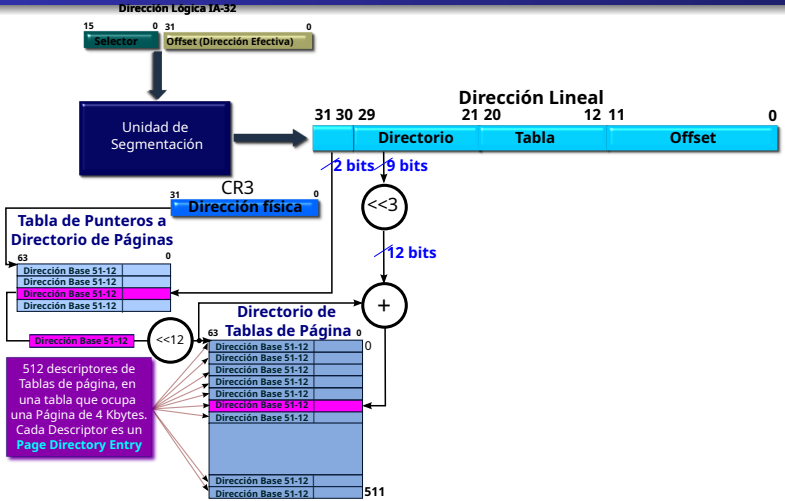
PAE: Traducción



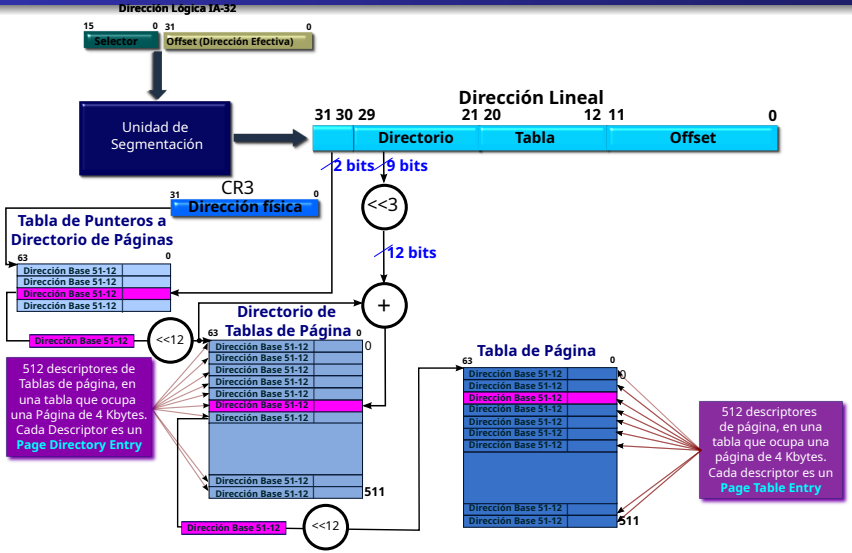
PAE: Traducción



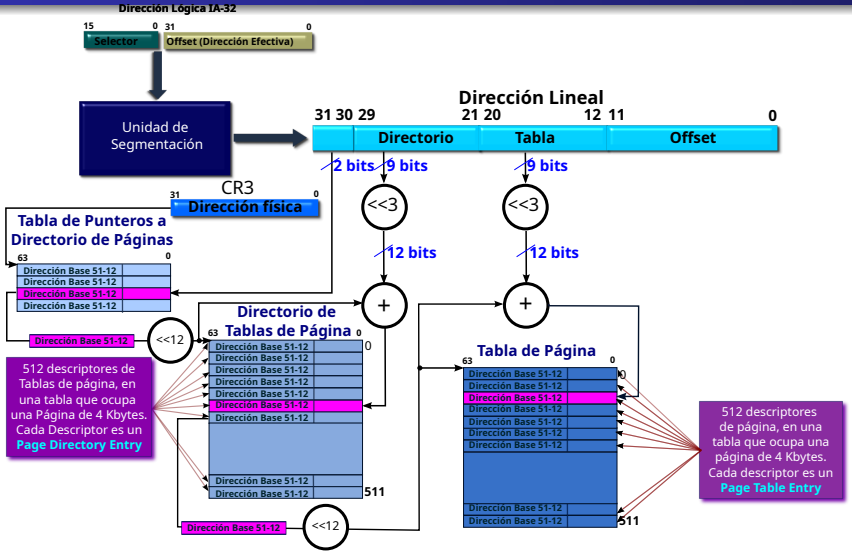
PAE: Traducción



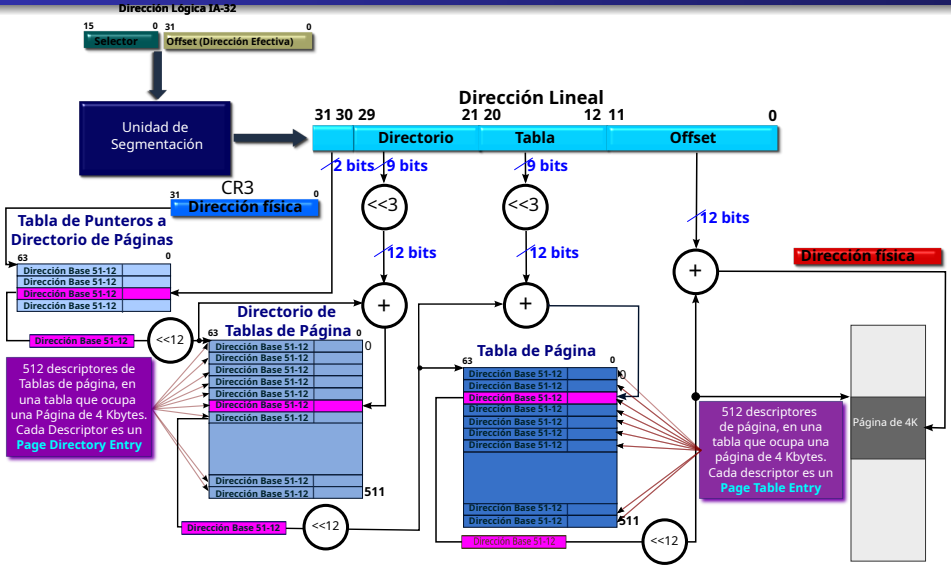
PAE: Traducción



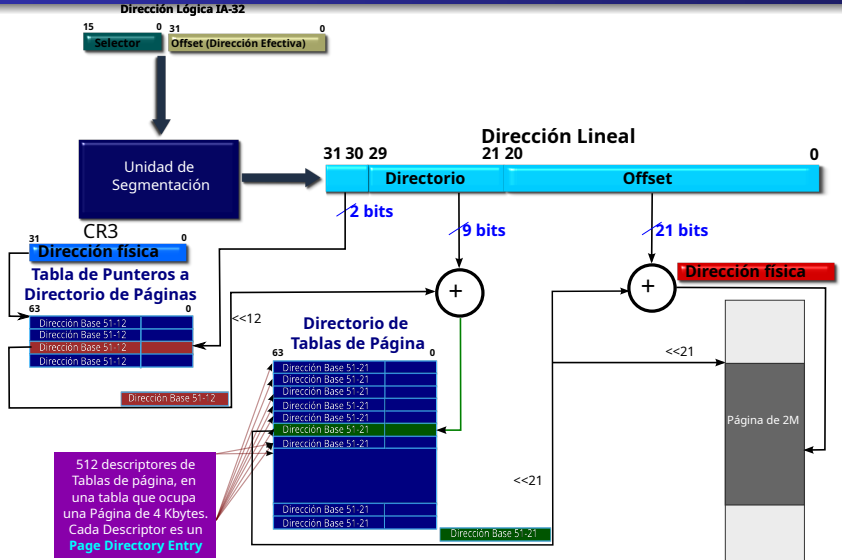
PAE: Traducción



PAE: Traducción



PAE: Traducción



PAE highlights

PAE highlights

- Permite mapear páginas de 4 Kbytes o 2 Mbytes en cualquier parte del espacio físico.

PAE highlights

- Permite mapear páginas de 4 Kbytes o 2 Mbytes en cualquier parte del espacio físico.
- Soporta la cantidad de memoria física que fuese necesaria sin restricciones.

PAE highlights

- Permite mapear páginas de 4 Kbytes o 2 Mbytes en cualquier parte del espacio físico.
- Soporta la cantidad de memoria física que fuese necesaria sin restricciones.
- Los descriptores de página ocupan en esta jerarquía el doble de tamaño (8 bytes).

PAE highlights

- Permite mapear páginas de 4 Kbytes o 2 Mbytes en cualquier parte del espacio físico.
- Soporta la cantidad de memoria física que fuese necesaria sin restricciones.
- Los descriptores de página ocupan en esta jerarquía el doble de tamaño (8 bytes).
- Por lo tanto en cada página física de 4 Kbytes caben 512 descriptores (la mitad respecto del Modo 32 bits).

PAE highlights

- Permite mapear páginas de 4 Kbytes o 2 Mbytes en cualquier parte del espacio físico.
- Soporta la cantidad de memoria física que fuese necesaria sin restricciones.
- Los descriptores de página ocupan en esta jerarquía el doble de tamaño (8 bytes).
- Por lo tanto en cada página física de 4 Kbytes caben 512 descriptores (la mitad respecto del Modo 32 bits).
- Se requieren 9 bits para indexar al descriptor buscado entre los 512 contenidos en la página.

Formato de los Descriptores

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Reservados												Dirección física Page Directory de 4 Kbytes [1]																			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física Page Directory de 4 Kbytes [1]																				Ignorados			Reservados (deben ser 0)				PCD	PWT	Resrv deben ser 0		P

[1] Los Bits 51 a M (con M=MAXPHYADDR) deben ser '0' si M < 52. MAXPHYADDR se obtiene, en el registro AL como resultado de la instrucción CPUID.0x80000008.

Figura: Formato de una entrada de la Tabla de Punteros a Directorio de Páginas (PDPTE)

Formato de los Descriptores

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
XD	Reservados											Dirección física del frame de la Tabla de Páginas de 4 Kbytes[1]																			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del frame de la Tabla de Páginas de 4 Kbytes[1]																				Ignorados				PS	IGN[2]	A	PCD	PWT	U/S	R/W	P

[1] Los Bits 51 a M (con M=MAXPHYADDR) deben ser '0' si M < 52. MAXPHYADDR se obtiene, en el registro AL como resultado de la instrucción CPUID.0x80000008.

[2] En el Bit 6 ***IGN*** significa ignorado.

Figura: Formato de una entrada del Directorio de Páginas (DPTE), que describe una Tabla de Páginas de 4 Kbytes

Formato de los Descriptores

63	62–52	51–32
XD	Reservados	Dirección física del frame de la Página de 2 Mbytes[1]

31–22	21–20	19–13	12	11–9	8	7	6	5	4	3	2	1	0
Dirección física del frame de Página de 2 Mbytes[1]	Reservados (deben ser 0)		PAT	Ignorados	G	PS	D	A	PCD	PWT	U/S	R/W	P

[1] Los Bits 51 a M (con M=MAXPHYADDR) deben ser '0' si M < 51. MAXPHYADDR se obtiene, en el registro AL como resultado de la instrucción CPUID.0x80000008.

Figura: Formato de una entrada del Directorio de Páginas (DPTE), que describe una Página de 2 Mbytes

Formato de los Descriptores

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
XD	Reservados											Dirección física del frame de la Página de 4 Kbytes [1]																			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física del frame de la Página de 4 Kbytes																				Ignorados			G	PAT	D	A	PCD	PWT	U/S	R/W	P

[1] Los Bits 51 a M (con M=MAXPHYADDR) deben ser '0' si M < 51. MAXPHYADDR se obtiene, en el registro AL como resultado de la instrucción CPUID.0x80000008.

Figura: Formato de una entrada del Tabla de Páginas (PTE), que describe una Página de 4 Kbytes

Temario

En 64 bits, la paginación lo es casi todo...

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.
- Los tres modos de Paginación dependen del estado de tres bits:

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.
- Los tres modos de Paginación dependen del estado de tres bits:

  CR0.PG Habilita Paginación

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.
- Los tres modos de Paginación dependen del estado de tres bits:

CR0.PG Habilita Paginación

CR4.PAE Habilita PAE

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.
- Los tres modos de Paginación dependen del estado de tres bits:

CR0.PG Habilita Paginación

CR4.PAE Habilita PAE

IA32_EFER.LME Hablita la Operación en modo IA-32e sub-modo 64 bits

En 64 bits, la paginación lo es casi todo...

- Este modo se activa únicamente al ingresar el procesador al modo de trabajo IA-32e sub modo 64 bits.
- Los tres modos de Paginación dependen del estado de tres bits:

CR0.PG	Habilita Paginación
CR4.PAE	Habilita PAE
IA32_EFER.LME	Hablita la Operación en modo IA-32e sub-modo 64 bits

- Este último al establecimier el modo IA-32e, automáticamente habilita la paginación de 64 bits.

En 64 bits, la paginación lo es casi todo...

Para cada uno de los modos los tres bits deben tener estas combinaciones:

En 64 bits, la paginación lo es casi todo...

Para cada uno de los modos los tres bits deben tener estas combinaciones:

1. ***Paginación de 32 bits***.
   Para ingresar a este modo se deben establecer los siguientes valores: ***CR0.PG*** = 1, ***CR4.PAE*** = 0, y ***IA32_EFER.LME*** = 0.

En 64 bits, la paginación lo es casi todo...

Para cada uno de los modos los tres bits deben tener estas combinaciones:

1. ***Paginación de 32 bits***.
   Para ingresar a este modo se deben establecer los siguientes valores:
   CR0.PG = 1, ***CR4.PAE*** = 0, y ***IA32_EFER.LME*** = 0.

2. ***Paginación PAE***.
   Para ingresar a este modo se deben establecer los siguientes valores:
   CR0.PG = 1, ***CR4.PAE*** = 1, y ***IA32_EFER.LME*** = 0.

En 64 bits, la paginación lo es casi todo...

Para cada uno de los modos los tres bits deben tener estas combinaciones:

1. ***Paginación de 32 bits***.
   Para ingresar a este modo se deben establecer los siguientes valores:
   CR0.PG = 1, ***CR4.PAE*** = 0, y ***IA32_EFER.LME*** = 0.

2. ***Paginación PAE***.
   Para ingresar a este modo se deben establecer los siguientes valores:
   CR0.PG = 1, ***CR4.PAE*** = 1, y ***IA32_EFER.LME*** = 0.

3. ***Paginación de 64 bits***.
   Para ingresar a este modo se deben establecer los siguientes valores:
   CR0.PG = 1, ***CR4.PAE*** = 1, y ***IA32_EFER.LME*** = 1.

Traducción en 64 bits

Traducción en 64 bits

- Cuando se ingresa en el modo IA-32e el procesador genera direcciones lineales de 64 bits, si bien las microarquitecturas utilizarán direcciones de menor tamaño.

Traducción en 64 bits

- Cuando se ingresa en el modo IA-32e el procesador genera direcciones lineales de 64 bits, si bien las microarquitecturas utilizarán direcciones de menor tamaño.
- Hasta el momento los procesadores que soportan el modo IA-32e utilizan direcciones lineales de 48 bits.

Traducción en 64 bits

- Cuando se ingresa en el modo IA-32e el procesador genera direcciones lineales de 64 bits, si bien las microarquitecturas utilizarán direcciones de menor tamaño.
- Hasta el momento los procesadores que soportan el modo IA-32e utilizan direcciones lineales de 48 bits.
- Con esto se tienen 256 Tbytes (TeraBytes) de espacio lineal de direccionamiento.

Traducción en 64 bits

- Cuando se ingresa en el modo IA-32e el procesador genera direcciones lineales de 64 bits, si bien las microarquitecturas utilizarán direcciones de menor tamaño.
- Hasta el momento los procesadores que soportan el modo IA-32e utilizan direcciones lineales de 48 bits.
- Con esto se tienen 256 Tbytes (TeraBytes) de espacio lineal de direccionamiento.
- Para prevenir futuras ampliaciones Intel introdujo entre las funciones largas de la instrucción CPUID, una que informa la cantidad de bits que componen una ***dirección lineal*** , que en la documentación de CPUID aparece como ***dirección virtual*** , y la cantidad de bits que componen una ***dirección física*** , que no es mas que el parámetro **`MAXPHYADDR`** ya mencionado.

Cantidades de bits en cada dirección

```
1 ;////////////////////////////////////////////////////////////////////////////
2 ;Llamar a CPUID con EAX = 0x80000008
3 ;////////////////////////////////////////////////////////////////////////////
4   mov eax,0x80000008
5   cpuid
6 ; En este punto se tiene
7 ; EAX 15-8 = Cantidad de bits soportados que conforman la dirección virtual
8 ; EAX 7-0 = Cantidad de bits soportados que conforman la dirección física
```

Estructuras de Paginación en 64 bits

Raiz de la estructura

Estructuras de Paginación en 64 bits

Raiz de la estructura

- Partimos del registro CR3. En el modo 64 bits su contenido depende de la habilitación o no del bit **CR4.PCIDE**.

Estructuras de Paginación en 64 bits

Raiz de la estructura

- Partimos del registro CR3. En el modo 64 bits su contenido depende de la habilitación o no del bit **CR4.PCIDE**.
- Este bit permite habilitar en el sistema cache la función **P**rocess **C**ontext **ID**entifier (PCID), que permite a un procesador lógico cachear información desde múltiples espacios de direcciones lineales.

Estructuras de Paginación en 64 bits

Raiz de la estructura

- Partimos del registro CR3. En el modo 64 bits su contenido depende de la habilitación o no del bit **CR4.PCIDE**.
- Este bit permite habilitar en el sistema cache la función **P**rocess **C**ontext **ID**entifier (PCID), que permite a un procesador lógico cachear información desde múltiples espacios de direcciones lineales.
- Así el procesador lógico puede retener información en el cache aun cuando el software conmute a otro espacio lineal de direccionamiento, por ejemplo cambiando el valor de CR3.

Estructuras de Paginación en 64 bits

Raiz de la estructura

- Partimos del registro CR3. En el modo 64 bits su contenido depende de la habilitación o no del bit **CR4.PCIDE**.
- Este bit permite habilitar en el sistema cache la función **P**rocess **C**ontext **ID**entifier (PCID), que permite a un procesador lógico cachear información desde múltiples espacios de direcciones lineales.
- Así el procesador lógico puede retener información en el cache aun cuando el software conmute a otro espacio lineal de direccionamiento, por ejemplo cambiando el valor de CR3.
- Se usa solo en el modo 64 bits con cuatro niveles de paginación habilitados.

Estructuras de Paginación en 64 bits

Soporte a PCID

- Si un procesador soporta PCID luego de invocar CPUID.01h, el bit 17 del registro ECX es '1'.

Estructuras de Paginación en 64 bits

Soporte a PCID

- Si un procesador soporta PCID luego de invocar CPUID.01h, el bit 17 del registro ECX es '1'.
- Una vez detectada su disponibilidad se habilita seteando el bit **CR4.PCIDE**

Estructuras de Paginación en 64 bits

Soporte a PCID

- Si un procesador soporta PCID luego de invocar CPUID.01h, el bit 17 del registro ECX es '1'.
- Una vez detectada su disponibilidad se habilita seteando el bit **CR4.PCIDE**
- Cada vez que el procesador genere una entrada en la estructura de la memoria cache y en la TLB, las asocia con el valor de PCID que figura en el campo homónimo del registro CR3 (layout en el siguiente slide).

Formato de CR3 en modo IA-32e

CR3

63											M_1+	M										39	38	37	36	35	34	33	32
Reservados (deben ser 0)												Dirección física alineada a 4 Kbytes de la PML4																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física alineada a 4 Kbytes de la PML4																				Ignorados							PCD	PWT	Ignor		

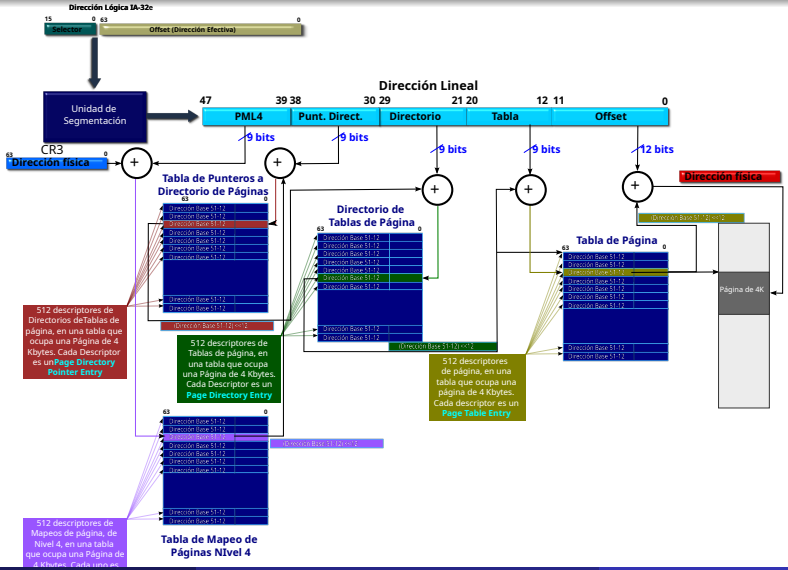
Figura: Layout del registro CR3 si CR4.PCIDE = 0

CR3

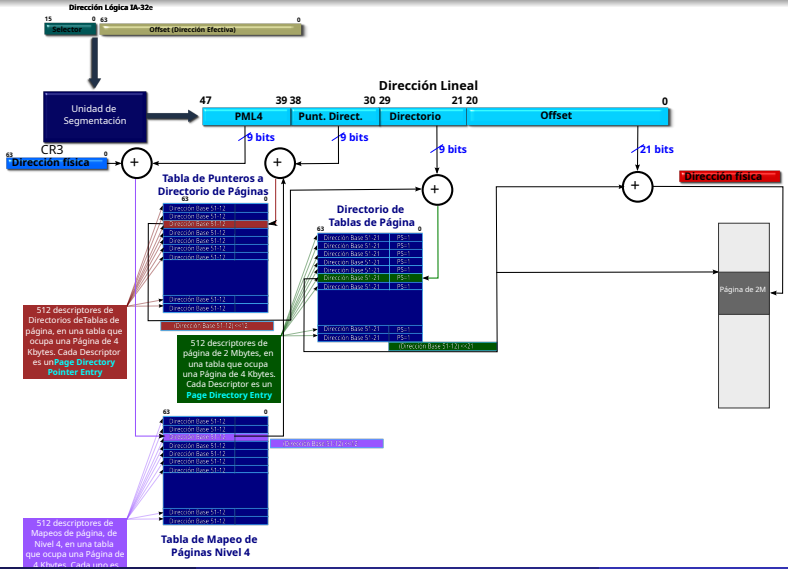
63											M_1+	M										39	38	37	36	35	34	33	32
Reservados (deben ser 0)												Dirección física alineada a 4 Kbytes de la PML4																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dirección física alineada a 4 Kbytes de la PML4																				PCID											

Figura: Layout del registro CR3 si CR4.PCIDE = 1

Traducción de 4 niveles y páginas de 4 Kbytes



Traducción de 3 niveles y páginas de 2 Mbytes



Páginas de 1 Gbyte de tamaño

Páginas de 1 Gbyte de tamaño

- El modo IA-32e tiene la posibilidad de trabajar con páginas de tamaño 1 Gbyte.

Páginas de 1 Gbyte de tamaño

- El modo IA-32e tiene la posibilidad de trabajar con páginas de tamaño 1 Gbyte.
- Para ello emplea una estructura jerárquica de dos niveles, eliminándose el nivel de Directorio de Tablas de Página.

Páginas de 1 Gbyte de tamaño

- El modo IA-32e tiene la posibilidad de trabajar con páginas de tamaño 1 Gbyte.
- Para ello emplea una estructura jerárquica de dos niveles, eliminándose el nivel de Directorio de Tablas de Página.
- Los 30 bits menos significativos de la dirección lineal son entonces el offset dentro de la página.

Páginas de 1 Gbyte de tamaño

- El modo IA-32e tiene la posibilidad de trabajar con páginas de tamaño 1 Gbyte.
- Para ello emplea una estructura jerárquica de dos niveles, eliminándose el nivel de Directorio de Tablas de Página.
- Los 30 bits menos significativos de la dirección lineal son entonces el offset dentro de la página.
- La tabla de descriptores (PML4) permite direccionar hasta 512 tablas de Punteros a Directorios de Página, cada una de cuyas entradas contiene un descriptor de página cuyo atributo PS=1, y describe directamente una página alineada a 1 Gbyte y de ese tamaño.

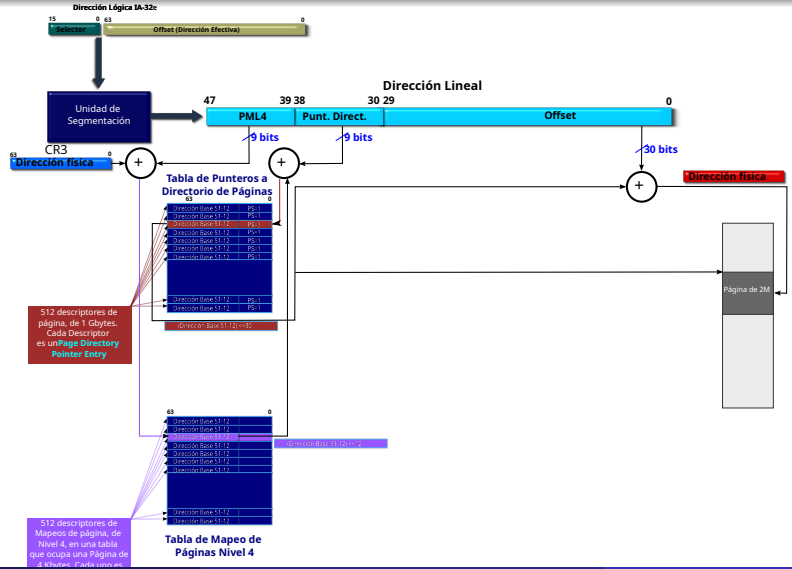
Páginas de 1 Gbyte de tamaño

- El modo IA-32e tiene la posibilidad de trabajar con páginas de tamaño 1 Gbyte.
- Para ello emplea una estructura jerárquica de dos niveles, eliminándose el nivel de Directorio de Tablas de Página.
- Los 30 bits menos significativos de la dirección lineal son entonces el offset dentro de la página.
- La tabla de descriptores (PML4) permite direccionar hasta 512 tablas de Punteros a Directorios de Página, cada una de cuyas entradas contiene un descriptor de página cuyo atributo PS=1, y describe directamente una página alineada a 1 Gbyte y de ese tamaño.
- 250 Mega Páginas de 1 Gbyte, nos dá un espacio físico de 256 Peta Bytes.

Páginas de 1 Gbyte de tamaño

- El modo IA-32e tiene la posibilidad de trabajar con páginas de tamaño 1 Gbyte.
- Para ello emplea una estructura jerárquica de dos niveles, eliminándose el nivel de Directorio de Tablas de Página.
- Los 30 bits menos significativos de la dirección lineal son entonces el offset dentro de la página.
- La tabla de descriptores (PML4) permite direccionar hasta 512 tablas de Punteros a Directorios de Página, cada una de cuyas entradas contiene un descriptor de página cuyo atributo PS=1, y describe directamente una página alineada a 1 Gbyte y de ese tamaño.
- 250 Mega Páginas de 1 Gbyte, nos dá un espacio físico de 256 Peta Bytes.
- Para activar páginas de este tamaño es necesario antes chequear si el procesador lo soporta. La función 0x80000001 de CPUID, deja el bit 26 de EDX en '1'.

Traducción de 2 niveles y páginas de 1 Gbytes



Formato de CPUID Función 0x80000001

Bit	Name	Description when Flag = 1	Comments
31:30		**Reserved**	Do not count on the value.
29	Intel® 64	Intel® 64 Instruction Set Architecture	The processor supports Intel® 64 Architecture extensions to the IA-32 Architecture. For additional information refer to http://developer.intel.com/technology/architecture-silicon/intel64/index.htm
28		**Reserved**	Do not count on the value.
27	RDTSCP	RDTSCP and IA32_TSC_AUX	The processor supports RDTSCP and IA32_TSC_AUX.
26	1 GB Pages	1 GB Pages	The processor supports 1-GB pages.
25:21		**Reserved**	Do not count on the value.
20	XD Bit	Execution Disable Bit	The processor supports the XD Bit when PAE mode paging is enabled.
19:12		**Reserved**	Do not count on the value.
11	SYSCALL	SYSCALL/SYSRET	The processor supports the SYSCALL and SYSRET instructions.
10:0		**Reserved**	Do not count on the value.

Temario

Comparación entre los modos de paginación

La estructura General de Paginación varía de acuerdo al Modo de Paginación empleado.

Tabla de características de los diferentes modos de paginación, organizada por cada nivel de paginación de la estructura.

Paging Structure	Entry Name	Paging Mode	Physical Address of Structure	Bits Selecting Entry	Page Mapping
PML4 table	PML4E	32-bit, PAE	N/A		
		IA-32e	CR3	47:39	N/A (PS must be 0)
Page-directory-pointer table	PDPTE	32-bit	N/A		
		PAE	CR3	31:30	N/A (PS must be 0)
		IA-32e	PML4E	38:30	1-GByte page if PS=1[1]
Page directory	PDE	32-bit	CR3	31:22	4-MByte page if PS=1[2]
		PAE, IA-32e	PDPTE	29:21	2-MByte page if PS=1
Page table	PTE	32-bit	PDE	21:12	4-KByte page
		PAE, IA-32e		20:12	4-KByte page

Resumen de cada Modo de Paginación

Paging Mode	PG in CR0	PAE in CR4	LME in IA32_EFER	Lin.-Addr. Width	Phys.-Addr. Width[1]	Page Sizes	Supports Execute-Disable?	Supports PCIDs?
None	0	N/A	N/A	32	32	N/A	No	No
32-bit	1	0	0[2]	32	Up to 40[3]	4 KB 4 MB[4]	No	No
PAE	1	1	0	32	Up to 52	4 KB 2 MB	Yes[5]	No
IA-32e	1	1	2	48	Up to 52	4 KB 2 MB 1 GB[6]	Yes[5]	Yes[7]

NOTES:

1. The physical-address width is always bounded by MAXPHYADDR; see Section 4.1.4.
2. The processor ensures that IA32_EFER.LME must be 0 if CR0.PG = 1 and CR4.PAE = 0.
3. 32-bit paging supports physical-address widths of more than 32 bits only for 4-MByte pages and only if the PSE-36 mechanism is supported; see Section 4.1.4 and Section 4.3.
4. 4-MByte pages are used with 32-bit paging only if CR4.PSE = 1; see Section 4.3.
5. Execute-disable access rights are applied only if IA32_EFER.NXE = 1; see Section 4.6.
6. Not all processors that support IA-32e paging support 1-GByte pages; see Section 4.1.4.
7. PCIDs are used only if CR4.PCIDE = 1; see Section 4.10.1.

Temario

Generalidades

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone que compila para diferentes microprocesadores.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone que compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone que compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.
- No basta para ello simplemente escribir código en C, sino hacerlo para que este resulte suficiente para cumplir estos objetivos.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone que compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.
- No basta para ello simplemente escribir código en C, sino hacerlo para que este resulte suficiente para cumplir estos objetivos.
- Implementar un código suficientemente flexible para un procesador x86 en lo que a paginación se refiere por si solo se hace un interesante problema, que implica detenerse a pensar de que forma definir una única estructura que sirva para implementar Paginación de 32, bits, PAE, o IA-32e.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone que compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.
- No basta para ello simplemente escribir código en C, sino hacerlo para que este resulte suficiente para cumplir estos objetivos.
- Implementar un código suficientemente flexible para un procesador x86 en lo que a paginación se refiere por si solo se hace un interesante problema, que implica detenerse a pensar de que forma definir una única estructura que sirva para implementar Paginación de 32, bits, PAE, o IA-32e.
- En general todos los procesadores tienen una estructura de paginación que básicamente se encarga de traducir una ***dirección virtual*** (en os procesadores x86 llamada ***dirección lineal***), en una ***dirección física*** , apta para enviar al exterior por el bus de direcciones.

Generalidades

- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.

Generalidades

- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.
- Hemos analizado la implementación de la paginación en los procesadores x86, en sus diferentes modos.

Generalidades

- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.
- Hemos analizado la implementación de la paginación en los procesadores x86, en sus diferentes modos.
- Esto es lo que hace atractivo el estudio de esta familia de procesadores: No hay ninguna mas compleja, y prácticamente en si misma presenta variantes para las que se requiere reunir otras varias arquitecturas diferentes.

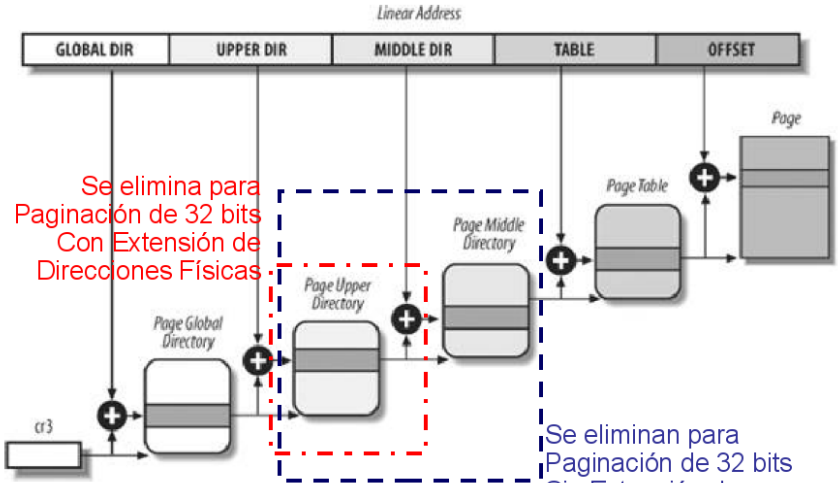
Generalidades

- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.
- Hemos analizado la implementación de la paginación en los procesadores x86, en sus diferentes modos.
- Esto es lo que hace atractivo el estudio de esta familia de procesadores: No hay ninguna mas compleja, y prácticamente en si misma presenta variantes para las que se requiere reunir otras varias arquitecturas diferentes.
- Habrá diferencias en la cantidad de bits de la ***dirección virtual*** , el tamaño de página, y otros detalles de implementación, en el mecanismo de traducción de cada arquitectura.

Generalidades

- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.
- Hemos analizado la implementación de la paginación en los procesadores x86, en sus diferentes modos.
- Esto es lo que hace atractivo el estudio de esta familia de procesadores: No hay ninguna mas compleja, y prácticamente en si misma presenta variantes para las que se requiere reunir otras varias arquitecturas diferentes.
- Habrá diferencias en la cantidad de bits de la ***dirección virtual*** , el tamaño de página, y otros detalles de implementación, en el mecanismo de traducción de cada arquitectura.
- Para implementar una estructura suficientemente flexible de programación, Linux se basa en un modelo genérico.

Estructura de Paginación General de Linux



Paginación para diferentes procesadores en Linux 32-bit

Arquitectura	PageSize	BitAddr	Niveles	Bits de cada campo
alpha	8 Kbytes	43	3	10 + 10 + 10 + 13
IA-64 (Itanium)	4 Kbytes	39	3	9 + 9 + 9 + 12
ppc64	4 Kbytes	41	3	10 + 10 + 9 + 12
sh64	4 Kbytes	41	3	10 + 10 + 9 + 12
Intel IA32	4 Kbyes	32	2	10 + 10 + 12
Intel IA32 PAE	4 Kbytes	32	3	2 + 9 + 9 + 12
Intel® 64	4 Kbytes	48	4	9 + 9 + 9 + 9 + 12
ARM CORTEX A	4 Kbyes	32	2	12 + 8 + 12

Así es que para cada arquitectura un conjunto de macros permiten definir la cantidad de bits de cada acampo de la ***dirección virtual*** particionada en 4 niveles en la Figura anterior, pudiendo colapsar uno o hasta dos campos intermedios simplemente poniendo 0 como su valor a la marco que define la cantidad de bits de ancho de ese campo.

Lineamientos para Administración de la memoria

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

1. El kernel se asigna de modo permanente un área de memoria para su código y sus estructuras de datos estáticas

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

1. El kernel se asigna de modo permanente un área de memoria para su código y sus estructuras de datos estáticas
2. El resto se denomina *memoria dinámica*, y consiste en un recurso fundamental, tanto para los procesos como para el mismo kernel.

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

1. El kernel se asigna de modo permanente un área de memoria para su código y sus estructuras de datos estáticas
2. El resto se denomina *memoria dinámica*, y consiste en un recurso fundamental, tanto para los procesos como para el mismo kernel.
3. Como consecuencia de los dos ítems anteriores, la correcta administración de la *memoria dinámica* impacta directamente en la performance del sistema.

Lineamientos para Administración de la memoria

Lineamientos para Administración de la memoria

Las distribuciones de Linux de 32 bits trabajan con páginas de 4 Kbytes solamente (aún las que implementan PAE). Pesaron en la decisión los siguientes factores que se señalan como cruciales.

Lineamientos para Administración de la memoria

Las distribuciones de Linux de 32 bits trabajan con páginas de 4 Kbytes solamente (aún las que implementan PAE). Pesaron en la decisión los siguientes factores que se señalan como cruciales.

- El algoritmo de swaping es mas simple.

Lineamientos para Administración de la memoria

Las distribuciones de Linux de 32 bits trabajan con páginas de 4 Kbytes solamente (aún las que implementan PAE). Pesaron en la decisión los siguientes factores que se señalan como cruciales.

- El algoritmo de swaping es mas simple.
- Si bien las transferencias de disco se hacen en múltiplos enteros de los tamaños de páginas de 4 Kbytes, lo cual permitiría emplear tamaños de páginas grandes, las transferencias de disco a memoria de menor tamaño son mas eficientes.

Detalle del Page Fault handler

Eso es todo

¿Preguntas?