

En los SO, muchas veces se requieren de distintos mecanismos para gestionar la memoria, de tal manera que exista la posibilidad de que se carguen varios procesos simultáneamente en la memoria principal.

Para realizar lo anterior se utiliza la memoria virtual, en las cuales un proceso define direcciones de memoria internamente, las cuales son posteriormente traducidas a memoria física mediante la unidad de administración de memoria (MMU por Memory Management Unit).

De esta manera, pueden existir más localidades de memoria virtual que en la dirección física, en donde también se tiene que un cierto número de localidades de memoria virtual corresponden a una página y, una vez se traen a memoria, se les conoce como marco de página.

El MMU realiza esta traducción de memoria virtual a memoria física, utilizando una estructura de datos llamada tabla de páginas.

Las tablas de página varían en su estructura dependiendo el SO, sin embargo, es común que para cada registro en la tabla se utilicen 32 bits, los cuales permiten ubicar el marco de página, permisos de página, si una página ha sido referenciada, etc.

### - ¿Qué es la paginación en 2 niveles?

Cuando se tienen bastantes páginas en memoria virtual y se necesita que las tablas de página no tengan un tamaño gigantesco, así como acelerar el proceso de traducción entre memoria lógica y física.

Para lograr esto se utiliza la paginación multinivel, la cual consiste en la creación de tablas de páginas que, en vez de apuntar a la dirección física de una página, apunta hacia otra tabla de páginas. Esto debido a que solamente existen ciertas porciones de tablas de página utilizadas en un determinado momento.

En la paginación en 2 niveles, solamente se tiene una tabla de orden superior o primer nivel, la cual apunta a tablas de orden inferior o de segundo nivel. En la tabla de páginas de segundo nivel es donde se encuentran las direcciones físicas de las páginas.

### - ¿Qué es una tabla de página invertida?

Una tabla de página contiene toda la información necesaria para relacionar las direcciones lógicas (páginas) con sus direcciones físicas (marcos), sin embargo, como los marcos con una menor cantidad a todas las páginas de un proceso, existen muchas páginas sin marco asignado, desperdiciando así memoria y tiempo de consulta.



Para resolver esta problemática se crean las tablas de página invertido, las cuales únicamente almacenan las páginas cargadas en direcciones físicas (marcos). Con la finalidad de buscar un marco, ahora se utiliza el ID del proceso y su página como llave.

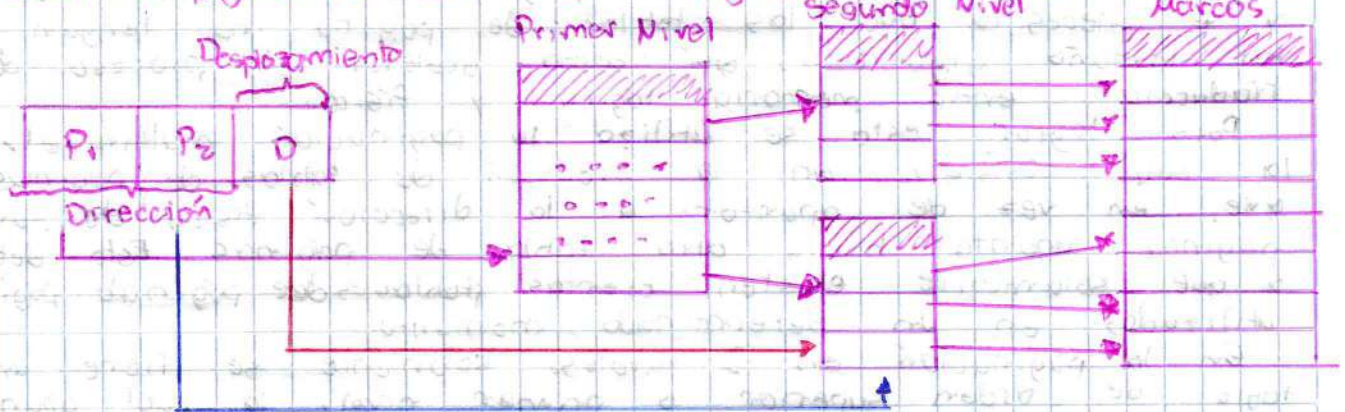
Esto presenta la problemática de que se debe buscar en toda la tabla invertida el conjunto PID (Process ID) y página para encontrar su respectivo marco.

Para solucionarlo se hace uso de un TLB y una función hash.

■ TLB (Translation Lookaside Buffer): También nombrada como memoria asociativa. Es un pequeño dispositivo de hardware dentro del MMU que contiene la información de los páginas más frecuentadas. Su tamaño varía entre 8 y 256 usualmente.

■ Función Hash: Cuando una página no se encuentra en el TLB y se debe buscar en la tabla invertida, se crea una tabla hash, la cual es, idealmente, del mismo tamaño que la tabla invertida, para que las cadenas asociadas a cada registro de la tabla hash sean de longitud 1 y la búsqueda sea más rápida. Cada nodo de las cadenas de la tabla hash almacena la página y su respectivo marco.

### Tablas de página de 2do nivel / paginación de 2 niveles



### Tablas de página invertidas





## • ¿Cómo se implementa la paginación en Linux?

Linux utiliza un tamaño de página de 4KB. En arquitecturas de 64 bits usa una tabla de página de tres niveles: el directorio global de la página, el directorio de en medio y la tabla de página. En consecuencia, una dirección lógica consta de cuatro partes, tres de las cuales son para los tres niveles y la cuarta es el número de byte dentro de una página.

Define los siguientes estados para marcos de página

- Libre: No ha sido asignado a un proceso
- Activo: Está en uso por un proceso
- Sucia inactiva: Fue modificada por el proceso aunque ya no se encuentra en uso.
- Inactiva brada: Era sucia inactiva y se está escribiendo en disco.

El reemplazo de página en Linux se basa en el algoritmo de reloj. El Kernel intenta mantener un número suficiente de marcos de página libres todo el tiempo de modo que los fallos de página puedan ser atendidos rápidamente utilizando uno de los tres marcos de página.

Linux utiliza un asignador acompañante para asignar marcos de página a los procesos.

El espacio de direcciones lógicas de un proceso puede constar de varias regiones de memoria virtual. Una página en una región de memoria cero llena se llena de ceros en su primera utilización. Una región respaldada con archivos facilita el mapeo en la memoria de archivos. Las entradas en la tabla de página de sus páginas apuntan a los memoriales intermedios de disco utilizados por el sistema de archivos. De esta manera, cualquier actualización en una página de dicho región se refleja de inmediato en el archivo y es visible para usuarios simultáneos del archivo. Una región de memoria privada se maneja en forma diferente. Cuando un nuevo proceso se bifurca, al proceso hijo se le proporciona una copia de la tabla de página del padre. En ese instante, a las páginas de una memoria privada se les proporciona un estado de copia durante la escritura. Cuando un proceso modifica esta página, se le hace una copia de la página.

## • ¿Cómo se implementa la paginación en Windows?

Windows opera en diferentes arquitecturas. Admite direcciones lógicas de 32 y 64 bits. El espacio de direcciones de un proceso es 2 o 3 GB. El resto del espacio de direcciones lógicas se reserva para uso del SO. El Kernel es mapeado en esta parte de todo espacio de direcciones del proceso. El tamaño de página es 4KB.



En una arquitectura X-86, Windows utiliza una organización de tabla de página de dos niveles, semejante a la vista para paginación en 2 niveles. La tabla de página de nivel superior se denomina directorio de página (PD). PD contiene 1024 entradas de 4 bytes cada una. Cada entrada en PD apunta a una tabla de página (PT). Cada tabla de página contiene 1024 entradas de tabla de página de 4 bytes cada una. A continuación se muestran los componentes de cada dirección lógica.

10 bits	10 bits	12 bits
Índice PD	Índice PT	Índice de bytes

El índice de bytes se encadena con la dirección del marco de página para obtener la dirección física efectiva. Cada entrada de una tabla de página contiene una dirección de 30 bits del marco de página que la contiene. Los 12 bits restantes se usan para distintos fines como estado, protección, etc.

En otras estructuras, Windows usa tablas de página de tres o cuatro niveles y diferentes formatos para las entradas de la tabla de página.

Un marco de página puede estar en ocho estados:

- Valid: La página está en uso activo.
- Free: La página no está en uso activo.
- Zeroed: La página está limpia y disponible para su uso inmediato.
- Standby: La página ha sido eliminada del conjunto de trabajo del proceso al que fue asignada, aunque es posible reconocerla al proceso si vuelve a ser referenciada de nuevo.
- Modified: La página es sucia y todavía no se ha escrito afuera.
- Bad: No es posible acceder a la página por un problema de hardware.

## Bibliografía

- Dhamdhere, D. M. (2008). Sistemas Operativos: Un enfoque basado en conceptos (2a. ed.). ProQuest Ebook Central.
- Sol, L. D. (2015). Sistemas operativos: Panorama para la ingeniería en computación e informática. ProQuest Ebook Central.
- Tanenbaum, A. (2015). Modern operating systems. Pearson.