



Centro de Enseñanza Técnica Industrial

Plantel Colomos

Ingeniería en Desarrollo de Software

Nombre Alumno: José Rafael Ruiz Gudiño

Registro: 20110374

Arquitectura de Sistemas Operativos

Act.4 Memoria Virtual. Resumen y crucigrama.

4°P

T/M

11/10/2021

Memoria principal

La memoria principal puede ser considerada como un arreglo lineal de localidades de almacenamiento de un Byte de tamaño. Cada localidad de almacenamiento tiene asignada una dirección que la identifica.

Jerarquía:

Gracias a un principio llamado cercanía de referencias es factible utilizar una mezcla de los distintos tipos y lograr un rendimiento cercano al de la memoria más rápida. Los niveles que componen la jerarquía de memoria habitualmente son:

- Nivel 0: registros.
- Nivel 1: memoria caché.
- Nivel 2: memoria principal.
- Nivel 3: disco duro (con el mecanismo de memoria virtual).

Algoritmo de reemplazo de páginas.

- Cuando ocurre un fallo de página el sistema operativo debe elegir una página para retirarla de la memoria y hacer un espacio para la página por recuperar.
- Si la página por eliminar fue modificada mientras estaba en memoria, debe escribirla en el disco para mantener actualizada la copia del disco, si por el contrario la página no ha sido modificada la copia del disco ya está actualizada por lo que no es necesario volver a escribir, la página por leer sólo escribe encima de la página por retirar.
- Aunque es posible elegir una página al azar para el reemplazo relacionado con un fallo de página, el rendimiento del sistema es mucho mejor si se elige una página de poco uso.

Memoria Virtual

La memoria virtual, es una técnica de administración de la memoria real que permite al sistema operativo brindarle al software de usuario y a sí mismo un espacio de direcciones mayor que la memoria real o física.

La mayoría de los ordenadores tienen cuatro tipos de memoria: registros en la CPU, la memoria caché (tanto dentro como fuera del CPU), la memoria física (generalmente en forma de RAM, donde la CPU puede escribir y leer directa y razonablemente rápido) y el disco duro que es mucho más lento, pero también más grande y barato.

Muchas aplicaciones requieren acceso a más información (código y datos) que la que se puede mantener en memoria física. Esto es así sobre todo cuando el sistema operativo permite múltiples procesos y aplicaciones ejecutándose simultáneamente. Una solución al problema de necesitar mayor cantidad de memoria de la que se

posee consiste en que las aplicaciones mantengan parte de su información en disco, moviéndola a la memoria principal cuando sea necesario.

Hay varias formas de hacer esto. Una opción es que la aplicación misma sea responsable de decidir qué información será guardada en cada sitio (segmentación), y de traerla y llevarla. La desventaja de esto, además de la dificultad en el diseño e implementación del programa, es que es muy probable que los intereses sobre la memoria de dos o varios programas generen conflictos entre sí: cada programador podría realizar su diseño teniendo en cuenta que es el único programa ejecutándose en el sistema.

La alternativa es usar memoria virtual, donde la combinación entre hardware especial y el sistema operativo hace uso de la memoria principal y la secundaria para hacer parecer que el ordenador tiene mucha más memoria principal (RAM) que la que realmente posee. Este método es invisible a los procesos. La cantidad de memoria máxima que se puede hacer ver que hay tiene que ver con las características del procesador. Por ejemplo, en un sistema de 32 bits, el máximo es 2³², lo que da 4096 Megabytes (4 Gigabytes). Todo esto hace el trabajo del programador de aplicaciones mucho más fácil, al poder ignorar completamente la necesidad de mover datos entre los distintos espacios de memoria.

Aunque la memoria virtual podría estar implementada por el software del sistema operativo, en la práctica casi siempre se usa una combinación de hardware y software, dado el esfuerzo extra que implicaría para el procesador.

La traducción de las direcciones virtuales a reales es implementada por una Unidad de Manejo de Memoria (MMU). El sistema operativo es el responsable de decidir qué partes de la memoria del programa es mantenida en memoria física. Además, mantiene las tablas de traducción de direcciones (si se usa paginación la tabla se denomina tabla de paginación), que proveen las relaciones entre direcciones virtuales y físicas, para uso de la MMU.

Finalmente, cuando una excepción de memoria virtual ocurre, el sistema operativo es responsable de ubicar un área de memoria física para guardar la información faltante, trayendo la información desde el disco, actualizando las tablas de traducción y finalmente continuando la ejecución del programa que dio la excepción de memoria virtual desde la instrucción que causó el fallo.

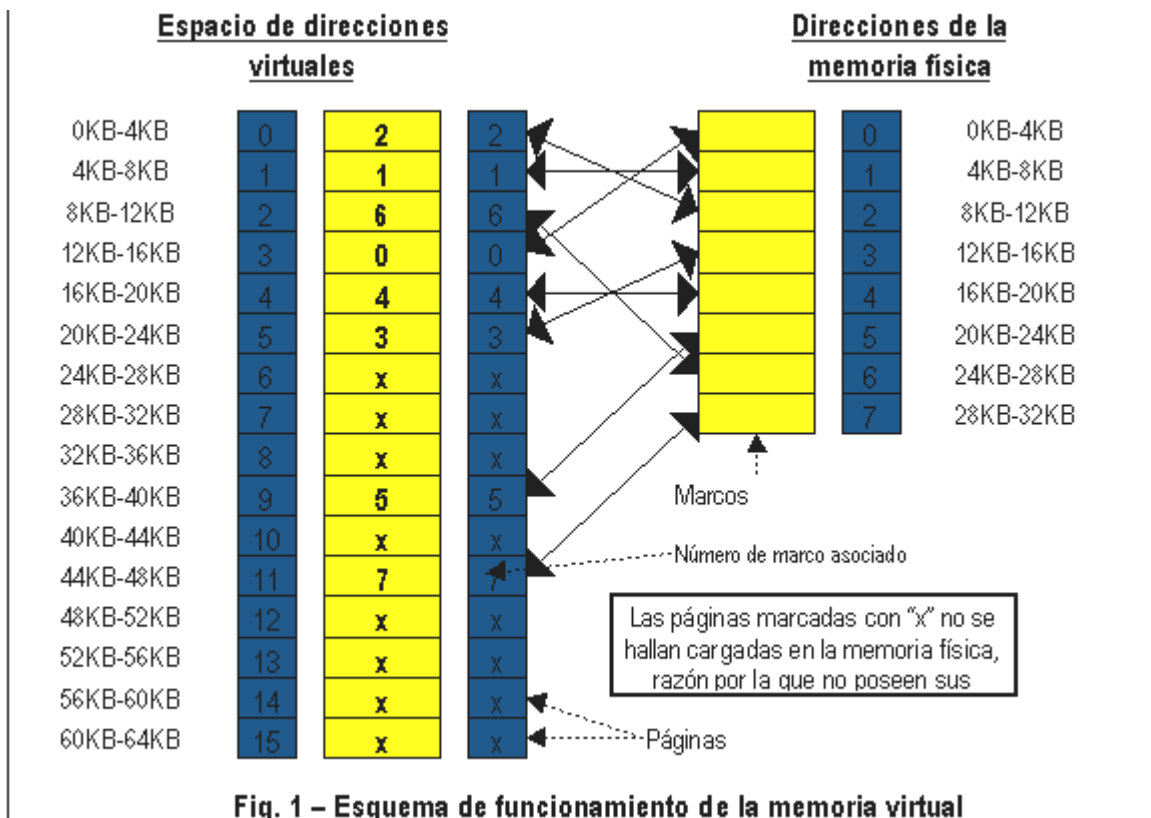


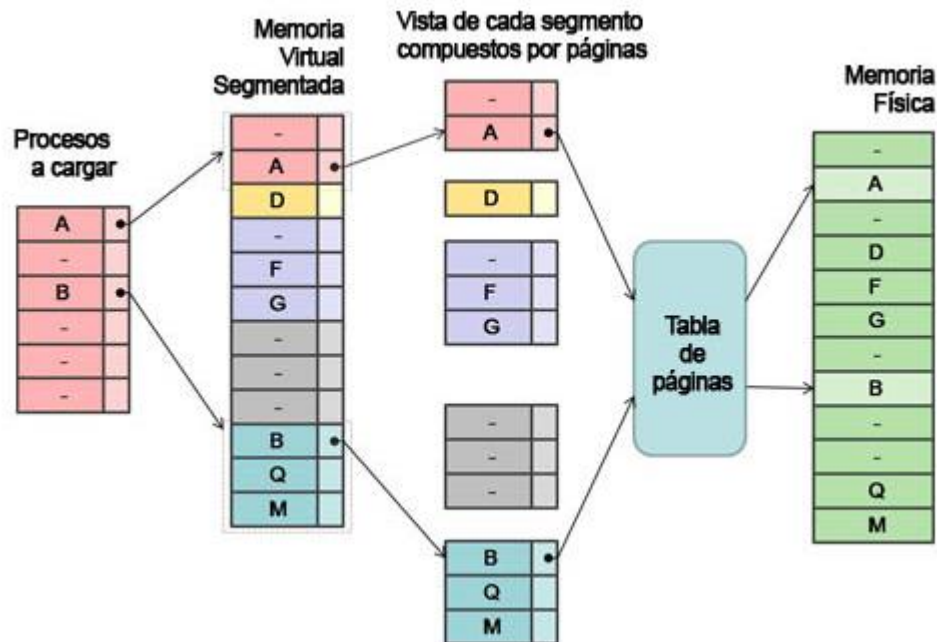
Fig. 1 – Esquema de funcionamiento de la memoria virtual

En la mayoría de las computadoras, las tablas de traducción de direcciones de memoria se encuentran en memoria física. Esto implica que una referencia a una dirección virtual de memoria necesitará una o dos referencias para encontrar la entrada en la tabla de traducción, y una más para completar el acceso a esa dirección.

Para acelerar el desempeño de este sistema, la mayoría de las Unidades Centrales de Proceso (CPU) incluyen una MMU en el mismo chip, y mantienen una tabla de las traducciones de direcciones virtuales a reales usadas recientemente, llamada Translation Lookaside Buffer (TLB).

El uso de este buffer hace que no se requieran referencias de memoria adicionales, por lo que se ahorra tiempo al traducir.

En algunos procesadores, esto es realizado enteramente por el hardware. En otros, se necesita de la asistencia del sistema operativo: se levanta una excepción, y en ella el sistema operativo reemplaza una de las entradas del TLB con una entrada de la tabla de traducción, y la instrucción que hizo la referencia original a memoria es reejecutada.



El hardware que tiene soporte para memoria virtual, la mayoría de las veces también permite protección de memoria. La MMU puede tener la habilidad de variar su forma de operación de acuerdo al tipo de referencia a memoria (para leer, escribir, o ejecutar), así como el modo en que se encontraba el CPU en el momento de hacer la referencia a memoria. Esto permite al sistema operativo proteger su propio código y datos (como las tablas de traducción usadas para memoria virtual) de corromperse por una aplicación, y de proteger a las aplicaciones que podrían causar problemas entre sí.

Referencias

Memoria Virtual. (s. f.). SYSTOPE. Recuperado 14 de octubre de 2021, de

<http://systope.blogspot.com/2012/06/memoria-virtual.html>