SISTEMAS INFORMÁTICOS

SISTEMAS OPERATIVOS

GESTIÓN DE MEMORIA

Alberto Martínez Pérez

1º CFGS DESARROLLO DE APLICACIONES WEB (daw)

**ÍNDICE DE PREGUNTAS**

[1.¿QUÉ ES LA FRAGMENTACIÓN INTERNA? ¿QUÉ ES LA FRAGMENTACIÓN EXTERNA? 3](#_Toc120214373)

[2.¿QUÉ ES LA MEMORIA VIRTUAL? 3](#_Toc120214374)

[3.¿QUÉ ES EL TLB? 3](#_Toc120214375)

[4.¿QUÉ ES EL ESPACIO DE INTERCAMBIO O SWAP? 4](#_Toc120214376)

[5.¿CÓMO SE TRADUCE UNA DIRECCIÓN VIRTUAL EN DIRECCIÓN FÍSICA EN LA PAGINACIÓN? 4](#_Toc120214377)

[6.¿EN QUÉ CONSISTE LA PAGINACIÓN? 5](#_Toc120214378)

[7.¿EN QUÉ CONSISTE LA SEGMENTACIÓN? 5](#_Toc120214379)

[8.PRINCIPAL DIFERENCIA ENTRE SEGMENTACIÓN Y PAGINACIÓN 6](#_Toc120214380)

[9.¿DE QUÉ SE COMPONE LA DIRECCIÓN VIRTUAL DE MEMORIA? 6](#_Toc120214381)

[10.FORMAS DE PARTICIÓN DE MEMORIA EN SISTEMAS MULTIPROGRAMADOS 6](#_Toc120214382)

1. ¿QUÉ ES LA FRAGMENTACIÓN INTERNA? ¿QUÉ ES LA FRAGMENTACIÓN EXTERNA?

La fragmentación se puede definir de forma general como espacios de memoria que no pueden ser utilizados por un proceso debido a que son demasiado pequeños para que puedan ser ocupados por ese proceso.

* **Fragmentación interna:** Se produce en las memorias que tienen una partición estática, es decir, la memoria se encuentra dividida en partes de un tamaño prefijado (sea este fijo o variable). Sucede cuando una partición es ocupada por un proceso de un tamaño menor a la misma de modo que deja un espacio libre, pero como estas particiones sólo pueden ser usadas por un proceso, es un espacio libre no utilizable por otro proceso.
* **Fragmentación externa:** Se producen en las memorias que tienen una partición dinámica, es decir, memorias que no tienen unas particiones prefijadas, sino que cada proceso irá cogiendo el tamaño de memoria que necesite para completar sus funciones. Sucede cuando un proceso abandona memoria tras terminar su función, pero el espacio que libera no puede ser utilizado por otro proceso al ser un espacio de menor tamaño que el del nuevo proceso.

1. ¿QUÉ ES LA MEMORIA VIRTUAL?

Es una técnica de gestión de la memoria del equipo que consiste en la utilización de una parte del almacenamiento secundario del sistema (HDD, SSD) como si se tratara de un módulo más de memoria.

Esto permite que no todo el programa tenga que estar permanentemente cargado en memoria, es decir, sólo se cargarán en RAM aquellas partes del programa que sean necesarias en ese momento y en los inmediatamente posteriores; y, además, nos permite que varias páginas residan en ella en lugar de en RAM lo que se va a traducir en un mayor número/espacio de direcciones disponibles.

El problema que tiene esto es que el acceso a disco es lento por lo que el proceso repetido de acceder al disco y llevar una página desde este hasta memoria afectará al rendimiento del sistema.

1. ¿QUÉ ES EL TLB?

El TLB o búfer de traducción de direcciones es una memoria que va a residir cerca del procesador y que va a almacenar las N últimas traducciones de direcciones virtuales a direcciones físicas que se hayan realizado de forma que si se necesita el acceso a una determinada dirección primero se comprobará si esta traducción se ha realizado y existe en el TLB:

* Si existe, el TLB mandará automáticamente la traducción, es decir, la dirección de memoria física.
* Si no existe, habrá que acceder a la tabla de páginas de la memoria física y realizar la traducción de forma normal.

Por tanto, el TLB nos va a permitir que en lugar de tener que acceder siempre 2 veces a memoria (principal problema de la paginación) haya ocasiones en las que sólo haya que acceder una vez (para acceder a la dirección física).

1. ¿QUÉ ES EL ESPACIO DE INTERCAMBIO O SWAP?

Es un espacio que el almacenamiento secundario tiene para intercambiar la memoria física con la memoria virtual, es decir, para mandar páginas de procesos que no se están usando en la memoria física para almacenarlos en la memoria virtual y, de esa manera, liberar de carga a la RAM o poder intercambiarlas con páginas siguientes del mismo proceso que hasta ese momento residían en memoria virtual.

En Windows lo conocemos como espacio de intercambio es un fichero que está en la raíz del disco duro de instalación o en la carpeta Windows. Este fichero cambiará de tamaño en función de los datos (páginas) que se almacenen en él.

En Linux lo conocemos como swap y en este caso se usa una participación de intercambio, aunque también puede ser un fichero como sucede en Windows.

1. ¿CÓMO SE TRADUCE UNA DIRECCIÓN VIRTUAL EN DIRECCIÓN FÍSICA EN LA PAGINACIÓN?

Una vez que llega una dirección de memoria virtual esta se divide en dos partes, una primera parte que es la página y otra que es el offset. Por ejemplo, si tenemos esta dirección de memoria virtual 0x010CA00B, 010 sería la página y CA00B sería el offset. Este offset lo que hace es referencia al desplazamiento que deberemos de hacer dentro del marco de páginas.

Lo que el sistema va a hacer con la página es un primer acceso a memoria física para ir a la tabla de páginas y comprobar a qué marco de página hace referencia esa página, por ejemplo, ese 010 hace referencia al marco 101. Este proceso lo lleva a cabo la MMU.

A continuación, se hará un segundo acceso a memoria en este caso ya para acceder a la dirección física, para ello el sistema accederá al marco 101 y se desplazará hasta la dirección CA00B, dando como resultado una dirección como esta 0x101FA00C.

1. ¿EN QUÉ CONSISTE LA PAGINACIÓN?

En organizar la memoria de forma que se divida en porciones de igual tamaño a las cuales se les llama marcos de página. Este modo de organizar la memoria facilita su gestión.

Estos marcos se van a identificar por un número, la página y va a existir una tabla de páginas que va a guardar el orden de los marcos, de forma que estos van a poder estar desordenados dentro de memoria.

Cada página se va a asignar a un proceso de forma exclusiva, de forma que con la paginación podríamos encontrarnos con fragmentación interna ya que un proceso podría requerir de 65KB y si cada página tuviera 64KB, requeriría de 2 páginas utilizando un único KB de la 2ª página, pero impidiendo que otro proceso pudiera utilizarla.

1. ¿EN QUÉ CONSISTE LA SEGMENTACIÓN?

En organizar la memoria en segmentos de tamaño variable. Estos segmentos van a contener direcciones de memoria a las cuales también vamos a poder acceder a través de una dirección virtual como ocurre en la paginación, pero en este caso las direcciones no se van a dividir en página y offset sino en segmento y offset. El segmento nos dará dos datos:

* Base: Que se refiere al límite inferior del segmento.
* Límite: Que se refiere al límite superior del segmento.

Con estos datos lo que va a hacer el sistema es mandar a la MMU el límite para valorar si el offset es mayor que el límite, es decir, si se quiere acceder a una dirección de memoria que está fuera del segmento:

* Si el resultado de la comparación es un true, es decir, el offset es mayor, se recibirá un error de segmentación al estar intentando acceder a una dirección fuera de los límites.
* Si el resultado de la comparación es un false, es decir, el offset es igual o menor, se hará el segundo acceso a memoria para conseguir la dirección.

1. PRINCIPAL DIFERENCIA ENTRE SEGMENTACIÓN Y PAGINACIÓN

La principal diferencia que podemos encontrar entre estos dos modos de organización de la memoria es que la paginación utiliza particiones estáticas (las páginas) mientras que la segmentación utiliza particiones dinámicas (los segmentos).

1. ¿DE QUÉ SE COMPONE LA DIRECCIÓN VIRTUAL DE MEMORIA?

De una primera parte referente a la página (la cual a través de la tabla de páginas nos servirá para averiguar el marco de página donde se encuentra la dirección real) y del offset o desplazamiento (el cual nos servirá para, una vez dentro del marco, desplazarnos por él hasta llegar a la dirección real).

1. FORMAS DE PARTICIÓN DE MEMORIA EN SISTEMAS MULTIPROGRAMADOS

En los sistemas multiprogramados encontramos los siguientes tipos de particiones:

* **Particiones estáticas de tamaño fijo**: Se parte la memoria en n partes llamadas particiones. Estas particiones son estáticas (es decir, no pueden variar su tamaño porque está prefijado) y son de un tamaño fijo (es decir, todas las particiones son del mismo tamaño).

A cada partición le vamos asignando un proceso, es decir, nos llega un proceso P1 que se colocará en el primer espacio de memoria, un proceso P2 que se colocará en el segundo…

En este modelo de particiones el tamaño de memoria que ocupa cada proceso es indiferente, cada proceso va a una partición. Salvo si llega un proceso que sea demasiado grande (como es el caso del proceso P4) que obviamente no podría entrar en ninguna de las particiones y no podría ejecutarse.

* **Particiones estáticas de tamaño variable:** Volvemos a tener una memoria dividida en n particiones. En este caso las particiones vuelven a ser estáticas, pero tienen un tamaño variable (es decir, el tamaño de cada partición no es siempre el mismo, sino que habrá unas más grandes y otras más pequeñas).

En este caso el tamaño de los procesos sí es importante para decidir qué partición de memoria se le asigna según las necesidades de memoria que tenga.

Podemos volvemos a tener el mismo problema que en las de tamaño fijo si nos llega un P4 que o bien sea demasiado grande o bien el hueco de memoria donde entraría está ocupado, se quedaría en espera o podría no llegar a ejecutarse nunca.

* **Particiones dinámicas**: En este modelo no tenemos unas particiones estáticas, sino que son dinámicas, es decir, cada proceso irá cogiendo la cantidad de memoria que necesita para llevar a cabo su trabajo.

En este sistema si bien no utiliza particiones de tamaño fijo también nos podemos encontrar con problemas de espacio. Por ejemplo, tenemos 3 procesos que han ido cogiendo la cantidad de memoria necesaria para hacer sus funciones, llegado un momento termina el P2 liberando ese especio en memoria y llega un P4 que no entraría en el espacio liberado por P2.

En los 2 primeros vamos a tener problemas de fragmentación interna, en el último vamos a tener problemas de fragmentación externa.