

MÓDULO:

Sistemas informáticos

UT2_3_Gestión de memoria

ÍNDICE DE CONTENIDO DE LA UNIDAD DIDÁCTICA

1	Gestión de memoria.....	2
1.1	Gestión de la memoria real	2
1.1.1	Mapa de bits	2
1.1.2	Lista enlazada de segmentos de memoria.....	2
1.2	Gestión de la memoria virtual	3
1.2.1	Paginación.....	3
1.2.2	Segmentación.....	4

1 Gestión de memoria

La memoria es una de los recursos más importantes del ordenador. El sistema operativo debe proporcionar mecanismos para asignarla a los procesos cuando se ejecutan, liberándola cuando terminan su ejecución.

Cada proceso en ejecución necesita un área de memoria para ubicar tres tipos de datos:

- El código del proceso (las instrucciones).
- Los datos (variables inicializadas).
- La pila (Zona de trabajo durante la ejecución).

Existe una parte del sistema operativo encargada de la administración de la memoria que se denomina **Administrador de memoria (MMU – Memory Management Unit)**. Controla las partes de la memoria en uso para así poder asignarla a nuevos procesos y liberarla cuando terminan su ejecución.

Otra de sus funciones es realizar el intercambio de un proceso completo entre la memoria principal y el disco, y viceversa, dado que en un sistema multiusuario no pueden estar almacenados todos los procesos en la memoria y parte de ellos deben estar en el disco.

1.1 Gestión de la memoria real

Como en el caso del uso de CPU, existen diferentes algoritmos para la gestión de la memoria libre. Para poder gestionar el espacio de memoria libre se utilizan principalmente dos métodos: mapa de bits y lista enlazada de segmentos de memoria.

1.1.1 Mapa de bits

Se asigna un bit a cada unidad en la que se divide la memoria. Dicho valor valdrá 0 si esa zona de la memoria está libre, y 1 si está ocupada.

Un inconveniente de este sistema es que para insertar un programa de K unidades de asignación se deben buscar en el mapa K unidades libres consecutivas, operación que no es demasiado rápida.

1.1.2 Lista enlazada de segmentos de memoria

Cada segmento es, o bien un proceso o un hueco libre de memoria, y para diferenciarlos cada elemento de la lista tiene tres campos: el primero indica si es un hueco o un proceso, el segundo el comienzo del segmento y el tercero la longitud del segmento.

Para enlazar bloques de memoria libre a los procesos que la requieran se utilizan las siguientes técnicas:

a) Primer ajuste

Se recorre secuencialmente la memoria hasta encontrar el primer hueco de tamaño suficiente para el proceso. Es un mecanismo sencillo de implementar.

b) Siguiente ajuste

Es como el primer ajuste, pero se guarda un registro con el último hueco encontrado, para comenzar la búsqueda a partir de ahí.

c) Mejor ajuste

Se trata de buscar en toda la lista el hueco cuyo tamaño se ajusta mejor al tamaño requerido. Es un mecanismo más lento que los anteriores, ya que se debe buscar en toda la lista.

d) Peor ajuste

Este algoritmo se comporta al contrario que el anterior, es decir, se busca el hueco de mayor tamaño posible para el tamaño que requiere el proceso, de manera que la parte sobrante de memoria sea lo suficientemente grande para poder ser utilizada por otro proceso.

Se ha demostrado que es poco eficiente.

e) Optimización adicional

Una optimización que se podría hacer a los cuatro algoritmos anteriores es mantener una lista separada de procesos y huecos. Si se ordenan los huecos en su lista de menor a mayor tamaño, en cuanto se encuentra un hueco que sirva, automáticamente es el mejor, luego el algoritmo del primer ajuste y el mejor ajuste son iguales. De igual forma y en consecuencia, el algoritmo del siguiente ajuste deja de tener sentido.

1.2 Gestión de la memoria virtual

El concepto de memoria virtual implica poder ejecutar programas que requieren una cantidad de memoria mayor que la memoria real disponible. Para ello, los programas se dividen en partes denominadas **capas**, de tal forma que en memoria principal solo residen algunas de las capas de un programa, estando el resto en disco. De esta forma se **usa disco como memoria RAM** (memoria virtual).

Vinculado con la memoria virtual está el concepto de **intercambio (swapping)**. Los procesos en espera pueden ser llevados al disco y dejar libre la parte de memoria que ocupan para que otros procesos entren en ejecución.

Cuando la ejecución del programa requiera una capa que no está en memoria, la MMU detiene la ejecución, envía a disco la capa que se estaba ejecutando y trae la capa que contiene el código a ejecutar. Los sistemas operativos multiusuario y multitarea actuales de Windows usan este tipo de técnica, para la que es recomendable disponer de entre un 2'5% y un 5% de espacio de disco libre.

Existen dos variantes de memoria virtual en función de si el tamaño de los bloques de memoria es fijo o variable: **la paginación y la segmentación**.

1.2.1 Paginación

Consiste en dividir la memoria, tanto virtual como física, en **bloques de memoria de tamaño fijo**. En la memoria virtual estos bloques se denominan **páginas**, y en la memoria física se denominan **marcos de página o frames**.

El administrador de memoria (MMU) llevará un control de los marcos que estén libres y de la página que esté cargada en cada uno. Cuando el proceso solicita un dato que se encuentra en una página que no está en memoria, se produce un **fallo de página** y el sistema operativo carga la página solicitada.

Cuando ocurre un fallo de página existen algoritmos de reemplazo de páginas para decidir qué página debe abandonar la memoria: la primera en entrar, la que hace más tiempo no es referenciada, la que no se va a volver a referenciar, etc.

Existe también una **tabla de páginas** donde la MMU asigna las direcciones físicas de los marcos a las páginas en que ha dividido un programa. Las páginas de un mismo programa no tienen por qué estar en marcos consecutivos.

1.2.2 Segmentación

Es similar a la paginación, pero permitiendo definir los **bloques de memoria de tamaño variable**, denominándose **segmentos**.

Puede haber segmentos de distintas longitudes y además un mismo segmento puede variar la suya en función de las necesidades del programa. Todo esto permite la compactación de memoria y un mejor aprovechamiento de la misma.

Los sistemas operativos Windows y Linux utilizan ambas técnicas de gestión de memoria virtual. Para implementar la memoria virtual en el disco utilizan distintas estrategias: Windows implementa la memoria virtual por medio de un archivo (**PageFile.sys**), mientras que en los sistemas Linux se reserva una partición en exclusiva para el intercambio (**partición de swap**).