

Preparador Informática

www.preparadorinformatica.com

TEMA 17. INFORMÁTICA / S.A.I.

**SISTEMAS OPERATIVOS:
GESTIÓN DE MEMORIA**

TEMA 17 INF / SAI: SISTEMAS OPERATIVOS: GESTIÓN DE MEMORIA.

1. INTRODUCCIÓN

2. GESTIÓN DE MEMORIA

2.1. MONOPROGRAMACIÓN

2.2. MULTIPROGRAMACIÓN EN MEMORIA REAL

2.2.1. PARTICIONES FIJAS (ESTÁTICAS)

2.2.2. PARTICIONES VARIABLES (DINÁMICAS)

2.2.3. PAGINACIÓN

2.2.4. SEGMENTACIÓN

2.3. MULTIPROGRAMACIÓN CON MEMORIA VIRTUAL

2.3.1. MEMORIA VIRTUAL PAGINADA

2.3.2. MEMORIA VIRTUAL SEGMENTADA

2.3.3. MEMORIA VIRTUAL SEGMENTADA PAGINADA

3. ESTRATEGÍAS DE GESTIÓN DE MEMORIA

3.1. POLÍTICAS DE SUSTITUCIÓN

3.2. POLÍTICAS DE CARGA

3.3. POLÍTICAS DE UBICACIÓN

4. CONCLUSIÓN

5. BIBLIOGRAFÍA

1. INTRODUCCIÓN

Hoy en día, los ordenadores constan de uno o más procesadores, memoria principal, discos, interfaces de red, periféricos y otros dispositivos de entrada/salida. En general se tratan de sistemas complejos, por lo que si los programadores tuvieran que comprender el funcionamiento de todos estos componentes les resultaría muy difícil desarrollar programas.

Por esta razón, los ordenadores están equipados con una capa de software llamada sistema operativo, que interacciona directamente con el hardware del equipo encargándose de controlar todos los recursos del mismo y presentar al usuario una interfaz más fácil de entender y programar.

El mercado de los sistemas operativos está ampliando sus horizontes para dirigirse también a otros equipos. Prueba de ello, es que Microsoft apuesta por estar presente en todos los dispositivos con el lanzamiento de **Windows 10 IoT Core**. Se trata de una versión reducida de Windows 10 optimizada para pequeños Mini-PCs (Raspberry Pi 2 y 3, Dragonboard 410c, etc.) que son productos relacionados con IoT (*Internet of Things*) y, en general, del sector de embebidos.

De igual forma, actualmente existen distribuciones Linux para una amplia variedad de dispositivos y fines, como por ejemplo versiones de propósito general (Red Hat, Fedora, Debian, etc.), de auditoría informática (BlackArch), para pequeños Mini-PCs (Raspbian), para entornos educativos (Guadalinex, Lliurex, MAX...), etc.

El presente tema está dedicado a estudiar el sistema operativo como responsable de las actividades relacionadas con la gestión de memoria, como son la asignación de memoria a los procesos que lo requieren, control de la memoria que está asignada, etc.

2. GESTIÓN DE MEMORIA

El sistema operativo lleva a cabo la gestión de la memoria por medio de un componente llamado gestor de memoria.

Las principales funciones del gestor de memoria son:

- En un sistema multitarea la memoria está compartida entre varios procesos, y el gestor de memoria debe decidir qué zonas de memoria asigna a cada proceso y que zonas descarga de memoria.
- Evitar que los procesos cargados en memoria interfieran unos con otros accediendo a zonas de memoria que no les corresponden.
- Controlar las zonas de memoria que están asignadas y las que están libres, además de saber las zonas de memoria que corresponden a cada proceso.
- Gestionar el intercambio de información entre memoria principal y memoria secundaria en los casos en que la memoria principal está completamente ocupada.

El sistema de gestión de memoria empleado dependerá del ordenador y del sistema operativo que se utilice. Las opciones en la gestión de memoria se dividen en función del número de procesos albergados en memoria (monotarea/multitarea) y de si se utiliza memoria real o virtual.

Memoria Real	Memoria Real	Memoria Virtual
	<i>Multitarea</i>	<i>Multitarea</i>
<i>Monotarea</i>	<ul style="list-style-type: none"> • Particiones fijas • Particiones variables • Paginación pura • Segmentación pura 	<ul style="list-style-type: none"> • Memoria virtual paginada • Memoria virtual segmentada • Combinación de ambas

2.1. MONOPROGRAMACIÓN

Se trata del esquema más sencillo, ya que la memoria se divide en dos partes: una parte para el sistema operativo que ha de residir en memoria de forma permanente (monitor residente) y la otra se les asigna a los procesos de usuario. Con este tipo de gestión, en la memoria sólo puede encontrarse un proceso en un instante determinado.



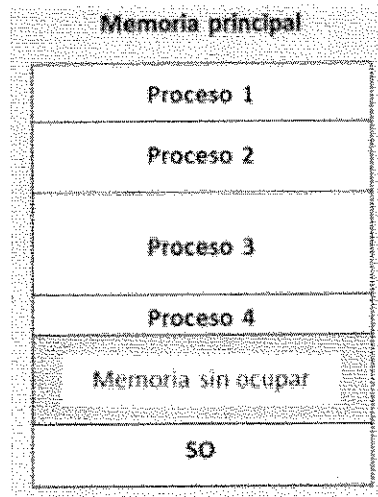
La monoprogramación, ya obsoleta, plantea varios problemas:

- Se desaprovecha memoria principal al no poder cargar más de un programa en memoria.
- Siempre que hay una operación de E/S el procesador no se utiliza, quedando desaprovechado.

Para conseguir mayor eficiencia en la utilización de los distintos recursos que ofrece un computador se desarrolló la multiprogramación.

2.2. MULTIPROGRAMACIÓN EN MEMORIA REAL

En la actualidad la mayoría de los sistemas operativos permiten que haya varios procesos simultáneamente en ejecución. Para que esto sea posible, todos estos procesos deben estar también simultáneamente en memoria. Con la multiprogramación se divide la memoria disponible en partes, a cada una de las cuales se le puede asignar un proceso o partes del mismo. Si todas las partes del proceso se cargan juntas, se dice que la asignación es contigua y, por el contrario, si las partes del proceso se cargan en diferentes espacios, se dice que su asignación es no contigua.



Cuando existen varios procesos que requieren ser cargados en memoria el gestor de memoria tiene que organizar el espacio para ubicarlos. Hay varias alternativas respecto a los posibles esquemas de multiprogramación en memoria real.

Se distinguen:

a) De asignación contigua

- **Particiones fijas (estáticas)**
- **Particiones variables (dinámicas)**

b) De asignación no contigua

- **Paginación**
- **Segmentación**

2.2.1. PARTICIONES FIJAS (ESTÁTICAS)

El esquema de particiones fijas consiste en dividir el espacio de memoria en particiones de tamaño fijo. Estas particiones pueden ser todas del mismo tamaño o tener distintos tamaños. El tamaño de las particiones se establece de forma lógica por el sistema operativo y están predefinidas antes de que lleguen los procesos.

Cada partición puede contener exactamente un proceso. De esta forma, el nivel de multiprogramación está limitado por el número de particiones.

Las particiones fijas presentan las siguientes desventajas:

- Si un proceso es de un tamaño mayor al de las particiones fijas el proceso debe redimensionarse para poder ser cargado.
- Cualquier proceso, por muy pequeño que sea, ocupará una partición completa. Este fenómeno, en el que se malgasta el espacio interno de una partición cuando el proceso cargado es más pequeño que la partición, se denomina **fragmentación interna**.

2.2.2. PARTICIONES VARIABLES (DINÁMICAS)

En estos sistemas la partición de la memoria no se define de forma permanente, sino que se van definiendo particiones del tamaño del proceso solicitante, por lo que no existe fragmentación interna. Es una técnica que aprovecha mejor el espacio de la memoria, ya que se ajusta al hecho de que el número y tamaño de los procesos varía dinámicamente.

El inconveniente que presenta es que conforme los procesos se cargan y extraen de la memoria, el espacio de memoria libre queda fragmentado en pequeños trozos, acabando por ser tan reducidos que no se pueda cargar un nuevo proceso. A este fenómeno se le conoce como **fragmentación externa**.

2.2.3. PAGINACIÓN

La paginación es una técnica que consiste en dividir la memoria principal en zonas iguales llamadas **marcos de página** y el mapa de memoria de cada proceso en partes del mismo tamaño, denominadas **páginas**. La estructura de datos que relaciona cada página con el marco donde está almacenada es la **tabla de páginas**.

Las páginas se almacenan en marcos de página libres independientemente de que estén o no contiguos. Por tanto, se trata de un esquema de asignación no contiguo.

El inconveniente de la paginación pura es que todas las páginas de un proceso deben estar en memoria principal para que se pueda ejecutar. Esto hace que, si los programas son de tamaño considerable, no puedan cargarse muchos a la vez, disminuyendo el grado de multiprogramación del sistema.

2.2.4. SEGMENTACIÓN

La segmentación es una técnica que consiste en dividir los procesos en una serie de segmentos de tamaño variable. Un proceso se carga situando todos sus segmentos en particiones dinámicas que no tienen que estar contiguas en memoria. La estructura de datos que relaciona cada segmento con su ubicación en memoria y su tamaño es la tabla de segmentos.

Al igual que la paginación pura, la segmentación pura presenta el inconveniente de que todos los segmentos de un proceso deben estar en memoria principal para que éste pueda ejecutarse.

2.3. MULTIPROGRAMACIÓN CON MEMORIA VIRTUAL

Con los esquemas de gestión de memoria vistos hasta el momento los procesos se cargan enteros en memoria, ya que es necesario para su ejecución que esté el proceso completo en memoria principal. Pero puede suceder que existan procesos grandes que no quepan en las particiones de la memoria y, por tanto, no puedan ser cargados por completo en la memoria.

La memoria virtual da una solución a este tipo de casos, ya que permite dividir los procesos en partes y cargar solamente algunas de ellas en memoria. La memoria virtual se basa en el uso de las técnicas de paginación o segmentación.

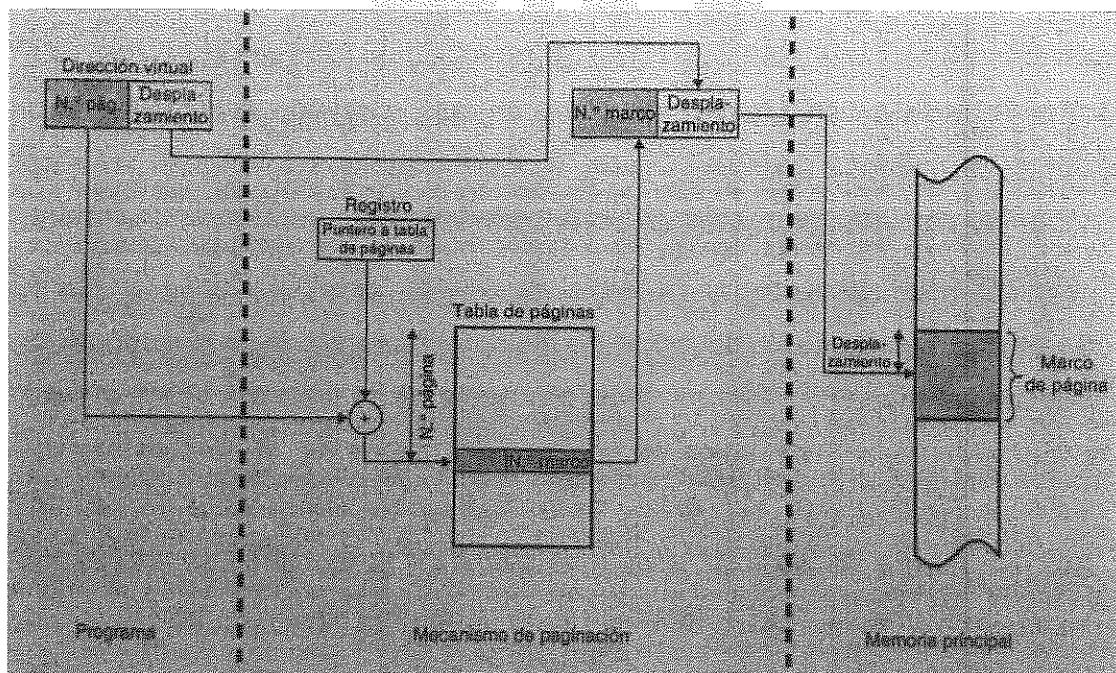
Debido a que no todas las partes de un proceso pueden estar cargadas en memoria en un instante determinado, sucede que cuando un proceso haga referencia a una parte que no se encuentre asignada en memoria provocará un fallo de página o de segmento, y el gestor de memoria traerá dicha parte del proceso de disco a memoria en ese momento.

La utilización de las técnicas de paginación o segmentación por parte de la memoria virtual se conocen como:

- Memoria virtual paginada
- Memoria virtual segmentada
- Memoria virtual segmentada paginada

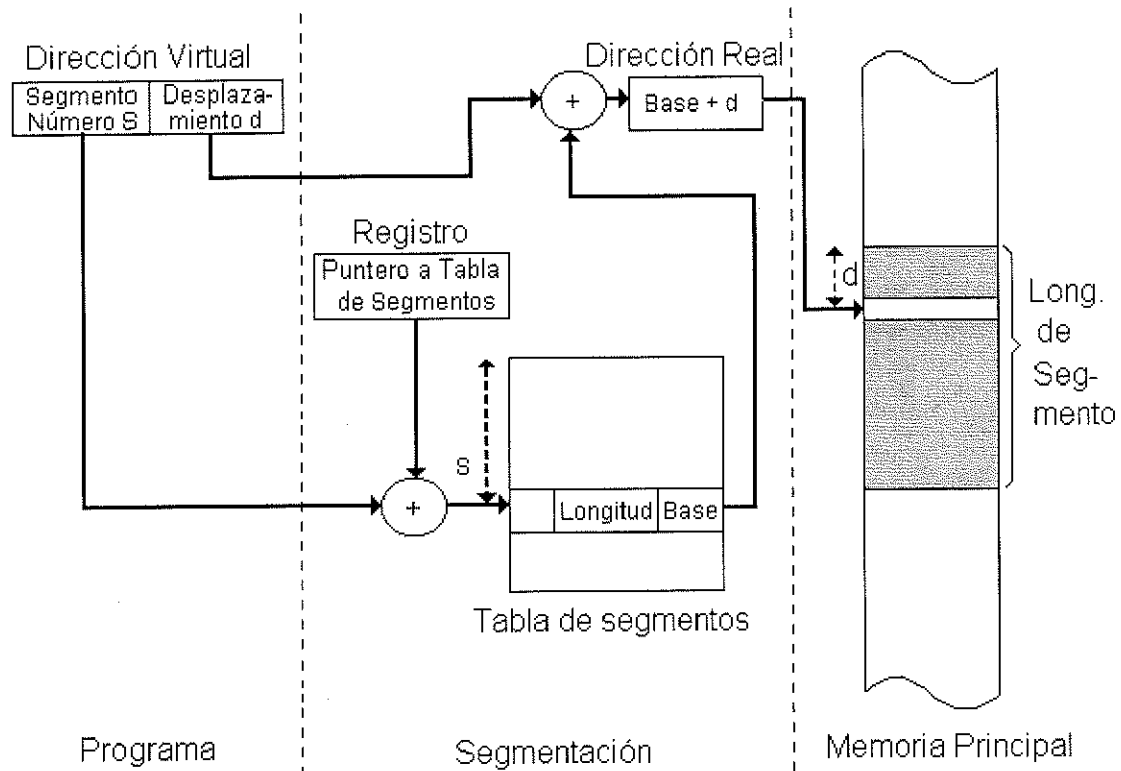
2.3.1. MEMORIA VIRTUAL PAGINADA

Sigue el funcionamiento de la paginación simple, pero no es necesario cargar todas las páginas de un proceso para que este pueda ejecutarse. Las páginas que no se encuentren y se necesiten se traerán posteriormente a memoria de manera automática. Reduce la fragmentación.



2.3.2. MEMORIA VIRTUAL SEGMENTADA

En este caso el funcionamiento sería el mismo que en la segmentación simple, pero tampoco es necesario cargar todos los segmentos de un proceso. Si se necesitan más segmentos no asignados en memoria se traerán en el momento en que sean referenciados.



2.3.3. MEMORIA VIRTUAL SEGMENTADA PAGINADA

La segmentación paginada intenta recoger lo mejor de los dos esquemas anteriores, proporcionando la segmentación soporte directo a las regiones del proceso y la paginación un mejor aprovechamiento de la memoria.

3. ESTRATEGÍAS DE GESTIÓN DE MEMORIA

Las estrategias o políticas para la gestión de la memoria pueden clasificarse en:

- Políticas de sustitución
- Políticas de carga
- Políticas de ubicación

3.1. POLÍTICAS DE SUSTITUCIÓN

Las políticas de sustitución son las que determinan qué bloque hay que sacar de la memoria principal. Es decir, son las responsables de la creación de zonas de memoria libres.

Las estrategias de sustitución más habituales son:

- a) **FIFO (First In First Out):** sustituye el bloque que haya entrado primero en memoria.
- b) **NRU (Not Recently Used):** se escoge el bloque que no haya sido usado (referenciado) en el ciclo anterior. Pretende aprovechar el hecho de la localidad en el conjunto de trabajo.
- c) **LRU (Least Recently Used):** es similar a la anterior, pero sustituye el bloque que se usó hace más tiempo, suponiendo que, como lleva mucho sin usarse es muy probable que siga sin usarse en los próximos ciclos. Necesita de una búsqueda exhaustiva.
- d) **NFU (Not Frequently Used):** este algoritmo toma en cuenta no tanto el tiempo, sino el número de referencias. En este caso cualquier bloque que se use muy poco, menos veces que algún otro.
- e) **LFU (Least Frequently Used):** es similar a la anterior, pero aquí se busca de forma exhaustiva aquel bloque que se ha usado menos que todos los demás.

3.2. POLÍTICAS DE CARGA

Las políticas de carga son las que determinan cuándo hay que cargar la información en la memoria principal. Se pueden distinguir dos categorías.

- a) **De demanda:** la falta de un bloque provoca una petición de carga, con lo que los algoritmos de sustitución y ubicación harán sitio en la memoria para el nuevo bloque.

b) **Anticipatorias:** cargan los bloques por adelantado, por lo que deben basarse en predicciones del comportamiento futuro del programa. Estas predicciones pueden realizarse en base a dos criterios:

- La naturaleza de la construcción del programa, refiriéndose esto al principio de localidad.
- La inferencia basada en el comportamiento pasado.

3.3. POLÍTICAS DE UBICACIÓN

Las políticas de ubicación son las que determinan en qué lugar de la memoria principal hay que colocar la información entrante.

Las políticas de ubicación más habituales son:

- Primer ajuste:** se asigna el primer hueco que sea mayor al tamaño deseado.
- Mejor ajuste:** se asigna el hueco cuyo tamaño exceda en la menor cantidad al tamaño deseado. Requiere de una búsqueda exhaustiva.
- Peor ajuste:** se asigna el hueco cuyo tamaño exceda en la mayor cantidad al tamaño deseado. Requiere también de una búsqueda exhaustiva.
- Siguiente ajuste:** es igual que el “primer ajuste” con la diferencia que se deja un apuntador en el lugar en donde se asignó el último hueco para realizar la siguiente búsqueda a partir de él.
- Ajuste rápido:** se mantienen listas enlazadas separadas de acuerdo a los tamaños de los huecos, para así buscarle a los procesos un hueco más rápido en la cola correspondiente.

4. CONCLUSIÓN

En este tema se ha presentado una visión global de la gestión de memoria, la cual es una tarea fundamental en los sistemas operativos. Hemos visto que para que un proceso se pueda ejecutar no solo requiere de tiempo de procesamiento sino también estar cargado en memoria principal. De este modo, la memoria se convierte en otro recurso clave que gestiona el sistema operativo y la parte encargada se denomina gestor de memoria.

5. BIBLIOGRAFÍA

- Tanenbaum A. **Sistemas operativos modernos**. Editorial Prentice Hall
- Stallings W. **Sistemas operativos**. Editorial Prentice Hall
- Deitel, H. **Introducción a los Sistemas Operativos**. Addison-Wesley.
- Prieto, A. y otros. **Introducción a la informática**. Editorial McGraw-Hill.
- http://atc.ugr.es/APrieto_videoclasas Departamento de Arquitectura y Tecnología de Computadores. Universidad de Granada.
- www.microsoft.com/windows
- www.linux.com (Web oficial de Linux Foundation)
- www.xataka.com (Web de actualidad sobre tecnología e informática).