

# GESTIÓN DE MEMORIA: TEORÍA

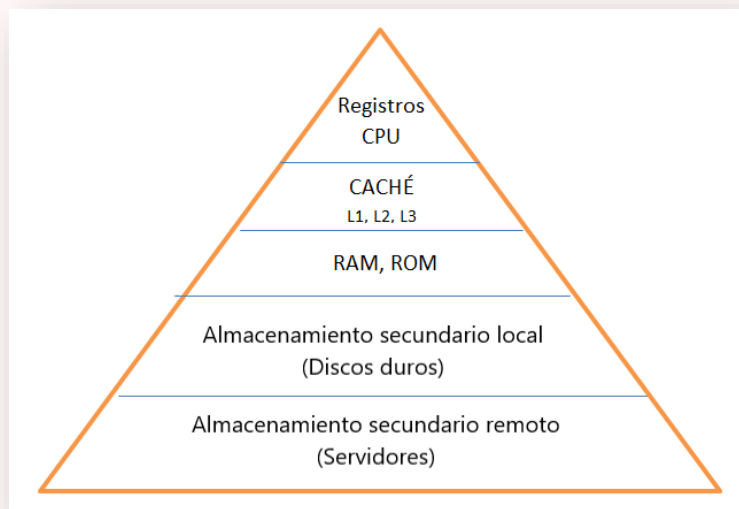
---



---

IMPLANTACIÓN DE SISTEMAS  
JUAN CARLOS NAVIDAD GARCÍA

### 1. Jerarquía de la memoria. Dibújalo.



### 2. Gestión de la memoria en los sistemas de monoprogramación.

En los sistemas con unidad de control monoprogramada solo existe un proceso de usuario y este tendrá todos los recursos del equipo, esto en pocas palabras, facilita la gestión de la memoria, ya que ésta sólo debe ser compartida por los programas del sistema operativo, y por el único proceso de usuario existente.

### 3. Gestión de la memoria en los sistemas multiprogramación. Esquema resumen.

En los sistemas con unidad de control multiprogramada la memoria es compartida por los diferentes procesos que existan, optimizando el uso de los recursos del equipo, lo que hace que la gestión de la memoria se complique.

#### 4. Estrategias de colocación en los sistemas multiprogramación de asignación de memoria con particiones dinámicas.

- **Primero en ajustarse (first fit):** El algoritmo más simple, Se revisa la lista de huecos hasta encontrar un espacio lo suficientemente grande. El espacio se divide entonces en dos partes, una para el proceso, y otra para la memoria no utilizada, excepto en el caso poco probable de un ajuste perfecto. Este algoritmo es rápido, ya que busca lo menos posible.
- **Mejor en ajustarse (best fit):** busca en toda la lista, y elige el mínimo hueco suficientemente grande como para ubicar al proceso. Este algoritmo intenta que los huecos que se creen en la memoria sean lo más pequeños posible.

#### 5. Memoria virtual.

La memoria virtual es una técnica para proporcionar la simulación de un espacio de memoria mucho mayor que la memoria física de una máquina, en otras palabras, se permite hacer programas de tamaño mayor al de la memoria principal.

## 6. Estrategias de la administración de la memoria virtual.

- **Estrategia para cargar páginas (o segmentos) a la memoria RAM.** Se usan más comúnmente dos estrategias:
  - **Cargado de páginas por demanda:** La estrategia de cargado por demanda consiste en que las páginas solamente son llevadas a RAM si son solicitadas, es decir, si se hizo referencia a una dirección que cae dentro de ellas.
  - **Cargado de páginas anticipada:** La carga anticipada consiste en tratar de adivinar qué páginas serán solicitadas en el futuro inmediato y cargarlas de antemano, para que cuando se pidan ya no ocurran fallos de página. Ese 'adivinar' puede ser que se aproveche del fenómeno de localidad y que las páginas que se cargan por anticipado sean aquellas que contienen direcciones contiguas a la dirección que se acaba de referenciar.
- **Estrategias de colocación.** Determinan en qué lugar de la memoria principal se debe colocar una página o un segmento entrante.
- **Estrategias de reemplazo.** Sirven para decidir qué página o segmento se debe desplazar para dejar espacio a una página o segmento entrante cuando está completamente ocupada la memoria principal

## 7. Algoritmos de reemplazo. Explícalos.

**Reemplazo FIFO:** El algoritmo de reemplazo más sencillo es el primero en entrar, primero en salir. Este algoritmo asocia a cada página el instante en el que se trajo a memoria. Cuando hay que reemplazar una página, se elige la más antigua.

**Reemplazo óptimo:** El algoritmo de reemplazo de páginas óptimo sería aquel que eligiera la página de la memoria que vaya a ser referenciada más tarde.

**Reemplazo menos recientemente usado o LRU (Least Recently Used):** se basa en el principio de localidad de referencia de los programas. Las páginas de uso frecuente en las últimas instrucciones se utilizan con cierta probabilidad en las siguientes instrucciones. De manera recíproca, es probable que las páginas que no hayan sido utilizadas durante mucho tiempo permanezcan sin ser utilizadas durante cierto tiempo. Es decir, estamos utilizando el pasado reciente como aproximación de lo que sucederá en el futuro. El algoritmo LRU explota esta idea: al ocurrir un fallo de página se utiliza la página que no haya sido utilizada hace más tiempo.

**Algoritmos de bits adicionales de referencia:** se basa en el principio de localidad de referencia de los programas. Las páginas de uso frecuente en las últimas instrucciones se utilizan con cierta probabilidad en las siguientes instrucciones. De manera recíproca, es probable que las páginas que no hayan sido utilizadas durante mucho tiempo permanezcan sin ser utilizadas durante cierto tiempo. Es decir, estamos utilizando el pasado reciente como aproximación de lo que sucederá en el futuro. El algoritmo LRU explota esta idea: al ocurrir un fallo de página se utiliza la página que no haya sido utilizada hace más tiempo.

**Algoritmos de la segunda oportunidad:** evita deshacerse de una página de uso frecuente, inspecciona el bit R de la página más antigua. Si es 0, la página es antigua y no utilizada, por lo que se reemplaza de manera inmediata. Si el bit es 1, R se pone a cero, la página se coloca al final de la lista de páginas, como si hubiera llegado en ese momento a la memoria. Después continúa la búsqueda siguiendo la lista.

**Algoritmo del reloj:** consiste en mantener las páginas en una lista circular, con la forma de un reloj. Al ocurrir un fallo de página, se inspecciona la página a la que apunta la manecilla. Si su bit R vale 0, la página se retira de la memoria, se inserta la nueva página en su lugar en el reloj, y la manecilla avanza una posición. Si R vale 1, este bit se pone a 0 y la manecilla avanza a la página siguiente. Este proceso continúa hasta encontrar una página con R a cero.

**Algoritmo LFU (Last Frequently Used):** mantiene un contador del número de referencias que se han hecho para cada página. Se reemplaza la página con el menor recuento. La razón para esta selección es que una página que se usa activamente debe tener un alto número de referencias. Este algoritmo tiene problemas cuando una página se usa mucho en la fase inicial de un proceso, pero después ya no se utiliza. Como se usó bastantes veces, tiene un recuento alto y permanece en memoria, aunque ya no se necesite.

**Algoritmo MFU (Most Frequently Used):** se basa en el argumento de que la página con el menor recuento probablemente acaba de llegar y aún tiene que usarse.

#### 8. ¿Qué memoria es más rápida, la RAM o la cache?

La memoria caché.

#### 9. ¿En un sistema con monoprogramación, puede compartirse entre el sistema operativo y varios procesos del usuario?

No, en un sistema con unidad de control monoprogramada solo existe un proceso de usuario.

**10. En los sistemas de monoprogramación ¿puede un programa de usuario ocupar cualquier zona de la memoria principal?**

No, el proceso tiene que asegurarse que el espacio que se le asigne es el que ocupe.

**11. ¿Qué significa asignación de memoria continua?**

Un proceso se ubica en su totalidad en posiciones consecutivas de memoria.

**12. ¿En qué se basa el proceso de asignación de memoria no continua?**

En este caso, los procesos pueden dividirse en bloques, y estos bloques pueden situarse en posiciones no contiguas de memoria principal.

**13. ¿Qué significa gestión de la memoria en particiones dinámicas? ¿y estáticas?**

**La gestión de memoria en particiones dinámicas** gestiona la memoria de manera que permite que se asigne la memoria dinámicamente a los procesos, conforme se introducen en la memoria. A cada proceso se le asigna exactamente la memoria que necesita.

**La gestión de memoria en particiones estáticas** divide la memoria en varias zonas, pudiendo ser cada zona de un tamaño diferente. El tamaño de las zonas podrá ser modificado eventualmente por algún usuario responsable de la administración del ordenador. El proceso se introduce en esa partición y permanecerá en ella hasta su finalización.

#### 14. ¿Qué problemas presentan la asignación de memoria en particiones estáticas?

- **Fragmentación interna:** Cuando un proceso se introduce en una partición, lo más probable es que su tamaño no sea el mismo (es decir, sea algo menor) que el de la partición.
- **Perdida de eficiencia:** Si un trabajo se iniciaba, y la partición para la que estaba compilado estaba ocupada, tenía que esperar, aunque estuvieran libres otras particiones.

#### 15. ¿Qué significa fragmentación interna? ¿Cuándo se produce?

Cuando un proceso se introduce en una partición, lo más probable es que su tamaño no sea el mismo (es decir, sea algo menor) que el de la partición.

#### 16. ¿Qué problemas plantea la asignación de memoria en las particiones dinámicas?

**Fragmentación externa:** Evita el problema de la fragmentación interna. Pero aparece el problema de la fragmentación externa entre particiones ya que se crean huecos libres demasiado pequeños como para que quepan procesos, aunque la unión de todos esos huecos produciría un hueco considerable, lo que acarrea el desperdicio de la memoria.

#### 17. ¿En qué consiste la asignación de las particiones dinámicas?

**La gestión de memoria en particiones dinámicas** gestiona la memoria de manera que permite que se asigne la memoria dinámicamente a los procesos, conforme se introducen en la memoria. A cada proceso se le asigna exactamente la memoria que necesita.



**18. ¿En qué consiste la fragmentación externa? ¿Cómo se soluciona la fragmentación externa? ¿En qué tipo de asignación de memoria se produce?**

Evita el problema de la fragmentación interna. Pero aparece el problema de la fragmentación externa entre particiones ya que se crean huecos libres demasiado pequeños como para que quepan procesos, aunque la unión de todos esos huecos produciría un hueco considerable, lo que acarrea el desperdicio de la memoria.

**19. Explica el algoritmo primero en ajustarse**

Revisa la lista de huecos hasta encontrar un espacio lo suficientemente grande. El espacio se divide entonces en dos partes, una para el proceso, y otra para la memoria no utilizada, excepto en el caso poco probable de un ajuste perfecto.

**20. Explica el algoritmo mejor en ajustarse.**

Busca en toda la lista, y elige el mínimo hueco suficientemente grande como para ubicar al proceso. Este algoritmo intenta que los huecos que se creen en la memoria sean lo más pequeños posible.

**21. ¿En qué consiste la asignación de memoria no contigua?**

Los procesos pueden dividirse en bloques, y estos bloques pueden situarse en posiciones no contiguas de memoria principal.

**22. ¿Qué algoritmo de remplazo se basa en primero en entrar primero en salir?**

Algoritmo de la segunda oportunidad.

### 23. Explica el algoritmo LRU.

Se basa en el principio de localidad de referencia de los programas. Las páginas de uso frecuente en las últimas instrucciones se utilizan con cierta probabilidad en las siguientes instrucciones. De manera recíproca, es probable que las páginas que no hayan sido utilizadas durante mucho tiempo permanezcan sin ser utilizadas durante cierto tiempo. Es decir, estamos utilizando el pasado reciente como aproximación de lo que sucederá en el futuro. El algoritmo LRU explota esta idea: al ocurrir un fallo de página se utiliza la página que no haya sido utilizada hace más tiempo.

### 24. ¿En qué consiste el swapping?

Técnica que permite el intercambio de páginas (mismo tamaño), segmentos (tamaño variable) o programas completos entre RAM y disco.

### 25. ¿Qué es la memoria virtual?

La memoria virtual es una técnica para proporcionar la simulación de un espacio de memoria mucho mayor que la memoria física de una máquina, en otras palabras, se permite hacer programas de tamaño mayor al de la memoria principal.

### 26. ¿Qué es la paginación? ¿y la segmentación? Diferencias.

La paginación se refiere a páginas de tamaño fijo y la segmentación a segmentos de tamaño variable, La segmentación proporciona soporte directo a las regiones del proceso y la paginación permite un mejor aprovechamiento de la memoria y una base para construir un esquema de memoria virtual ya que utiliza un tamaño fijo de páginas.

**27. ¿En qué método de asignación de memoria se utiliza la paginación? ¿Y la segmentación?**

En la asignación de memoria contigua mediante la memoria virtual.

**28. ¿Calcular el número de fallos para los algoritmos FIFO y Optimo de las siguientes secuencias de referencias a páginas**

a. 1, 3, 0, 3, 5, 6, 3, FIFO con 3 marcos

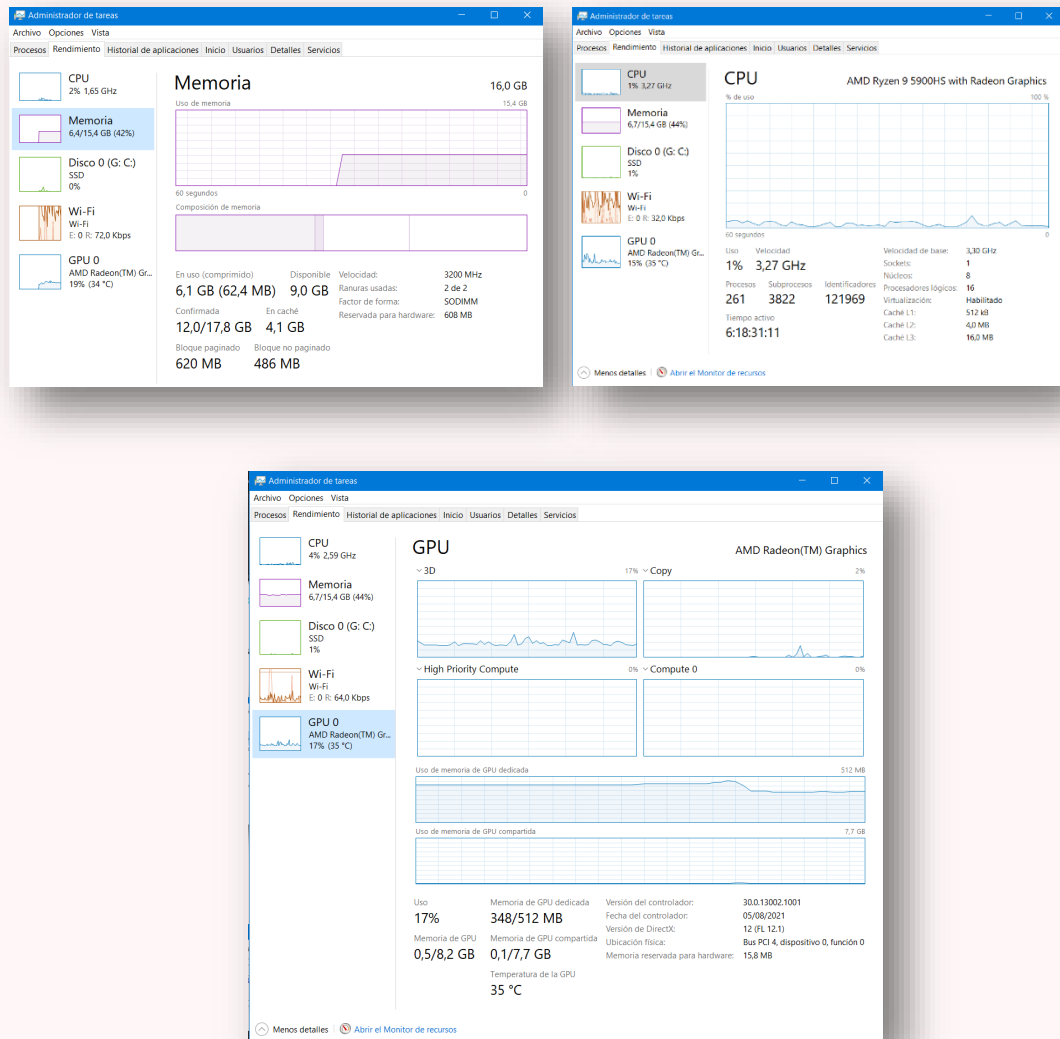
1	3	0	3	5	6	3
1	1	1	1	5	5	5
	3	3	3	3	6	6
		0	0	0	0	3
F	F	F		F	F	F

b. 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 Optimo con 3 marcos

7	0	1	2	0	3	0	4	2	3	0	3	2
7	7	7	2		2		2			2		
	0	0	0		0		4			0		
		1	1		3		3			3		
F	F	F	F		F		F			F		

## 29. Identifica las características de tu PC:

- Administrador de tareas



## ○ CPU-Z

