

Trabajo Practico N° 3.

Alumno: Tomas Alaluf, Gonzalo Errandonea, Gonzalo Romero.

Profesor: Osvaldo Aguiar.

Instituto: UADER FCYT.

Carrera: Licenciatura en sistemas de información.

Materia: Sistemas Operativos.

1. ¿Cómo se divide la memoria en un sistema monoprogramado, y en uno multiprogramado?

- En un sistema monoprogramado la memoria se divide en dos partes: una parte para el sistema operativo (monitor residente, núcleo) y una parte para el programa actualmente en ejecución.
- En un sistema multiprogramado la parte de <usuario> de la memoria se debe subdividir posteriormente para acomodar múltiples procesos.

2. ¿Quién se encarga de la subdivisión de la memoria? ¿Cómo se denomina esta tarea?

- El sistema operativo se encarga de subdividir y a esta tarea se le denomina gestión de memoria.

3. ¿Cuáles son los requisitos que debe satisfacer la gestión de memoria? Explique cada uno de ellos.

- Reubicación: En un sistema multiprogramado la memoria principal disponible se comparte generalmente entre varios procesos. El programador no puede anticipar qué programas estarán en memoria

principal cuando se ejecute su programa por lo que poder intercambiar procesos en memoria principal para maximizar la utilización del procesador es algo positivo.

Una vez que un programa se ha llevado al disco, sería bastante limitante tener que colocarlo en la misma región de memoria donde se hallaba anteriormente, cuando este se trae de nuevo a la memoria. Por el contrario, podría ser necesario reubicar el proceso a un área de memoria diferente. A este proceso se le llama reubicación.

- **Protección:** Cada proceso debe protegerse contra interferencias no deseadas por parte de otros procesos, sean accidentales o intencionadas. Un programa de otro proceso debe poder ser capaz de referenciar sin permiso a posiciones de memoria de un proceso, tanto en modo lectura como escritura. Todas las referencias de memoria generadas por un proceso deben comprobarse en tiempo de ejecución para poder asegurar que se refieren sólo al espacio de memoria asignado a dicho proceso. Los requisitos de esta protección de memoria deben ser satisfechos por el procesador (hardware) y no por el sistema operativo (software).
- **Compartición:** Cualquier mecanismo de protección debe tener la flexibilidad de permitir a varios procesos acceder a la misma porción de memoria principal. Procesos que estén cooperando en la misma tarea podrían necesitar compartir el acceso a la misma estructura de datos. Por lo tanto, el sistema de gestión de la memoria debe permitir el acceso controlado a áreas de memoria compartidas sin comprometer la protección esencial.
- **Organización lógica:** La memoria principal de un computador se organiza como un espacio de almacenamiento lineal o unidimensional, compuesto por una secuencia de bytes o palabras. Mientras que esta organización es similar al hardware real de la máquina, no se corresponde a la forma en la cual los programas se construyen normalmente, ya que éstos se organizan en módulos, algunos de los cuales no se pueden modificar y algunos contienen datos que se pueden modificar. El tratar a los programas de usuarios y datos en la forma de módulos permite varias ventajas:
 1. Los módulos se pueden escribir y compilar independientemente, con todas las referencias de un módulo desde otro resueltas por el sistema en tiempo de ejecución.
 2. Con una sobrecarga adicional modesta, se puede proporcionar diferentes grados de protección a los módulos (sólo lectura, sólo ejecución).
 3. Es posible introducir mecanismos por los cuales los módulos se pueden compartir entre los procesos. La ventaja de proporcionar compartición a nivel de módulo es que se corresponde con la forma en la que el usuario ve el problema, y por tanto es fácil para éste especificar la compartición deseada. La

herramienta que más adecuadamente satisface estos requisitos es la segmentación

- Organización física: la memoria del computador se organiza en al menos dos niveles conocidos como memoria principal y memoria secundaria. La memoria principal proporciona acceso rápido a un coste relativamente alto. Adicionalmente, es volátil, lo que quiere decir que el almacenamiento no es permanente. Por otro lado, la memoria secundaria de larga capacidad puede proporcionar almacenamiento para programas y datos a largo plazo, mientras que una memoria principal más pequeña contiene programas y datos actualmente en uso. En este esquema de dos niveles la organización del flujo de información entre la memoria principal y secundaria supone una de las preocupaciones principales del sistema. La responsabilidad para este flujo podría asignarse a cada programador en particular, pero no es practicable por dos motivos.
 1. La memoria principal disponible para un programa más sus datos podría ser insuficiente. En este caso, el programador debería utilizar una técnica conocida como superposición (overlaying). Esta programación con overlays malgasta tiempo del programador.
 2. En un entorno multiprogramado el programador no conoce en tiempo de codificación cuánto espacio estará disponible o dónde se localizará dicho espacio.

Por tanto está claro que la tarea de mover la información entre los dos niveles debería ser una responsabilidad del sistema, Esta tarea es la esencia de la gestión de memoria.

4. ¿En qué consiste el particionamiento fijo de memoria? Explique las dos alternativas que existen.

- El particionamiento fijo de memoria consiste en repartir la memoria en regiones con límites fijos.
- Paginación sencilla: La memoria principal se divide en marcos del mismo tamaño. Cada proceso se divide en páginas del mismo tamaño que los marcos. Un proceso se carga a través de la carga de todas sus páginas en marcos disponibles, no necesariamente contiguos.
- Segmentación sencilla: Cada proceso se divide en segmentos. Un proceso se carga cargando todos sus segmentos en particiones dinámicas, no necesariamente contiguas.

5. ¿Qué dificultades existen con el uso de particiones fijas del mismo tamaño?

- Existen dos dificultades:

- Un programa podría ser demasiado grande para caber en una partición, por lo que el programador debería diseñar su programa con el uso de overlays.
- La utilización de la memoria principal es extremadamente ineficiente. Cualquier programa por pequeño que sea ocupa una partición completa.

6) Explique cómo funcionaría el algoritmo de ubicación en particiones fijas y en particiones de diferente tamaño.

Con particiones del mismo tamaño o fijas, la ubicación de los procesos en memoria es trivial. En cuanto haya una partición disponible, un proceso se carga en dicha partición. Debido a que todas las particiones son del mismo tamaño, no importa qué partición se utiliza. Si todas las particiones se encuentran ocupadas por procesos que no están listos para ejecutar, entonces uno de dichos procesos debe llevarse a disco para dejar espacio para un nuevo proceso.

Con particiones de diferente tamaño, hay dos formas posibles de asignar los procesos a las particiones. La forma más sencilla consiste en asignar cada proceso a la partición más pequeña dentro de la cual cabe. En este caso, se necesita una cola de planificación para cada partición, que mantenga procesos en disco destinados a dicha partición. La ventaja de esta técnica es que los procesos siempre se asignan de tal forma que se minimiza la memoria malgastada dentro de una partición (fragmentación interna).

7) ¿En qué consiste la técnica de particionamiento dinámico?

Con particionamiento dinámico, las particiones son de longitud y número variable. Cuando se lleva un proceso a la memoria principal, se le asigna exactamente tanta memoria como requiera y nomás.

8) ¿Cuántos algoritmos de ubicación se consideran en el particionamiento dinámico? Explique cada uno de ellos.

Existen tres algoritmos de ubicación que pueden considerarse como lo son mejor-ajuste (best fit), primer-ajuste(first-fit) y siguiente-ajuste (next-fit). Todos, por supuesto, están limitados a escoger entre los bloques libres de la memoria principal que son iguales o más grandes que el proceso que va a llevarse a la memoria.

Mejor-ajuste escoge el bloque más cercano en tamaño a la petición.

Primer-ajuste comienza a analizar la memoria desde el principio y escoge el primer bloque disponible que sea suficientemente grande.

Siguiente-ajuste comienza a analizar la memoria desde la última colocación y elige el siguiente bloque disponible que sea suficientemente grande.

9) ¿Cuál es la diferencia entre dirección lógica y física?

Una dirección lógica es una referencia a una ubicación de memoria independiente de la asignación actual de datos a la memoria; se debe llevar a cabo una traducción a una dirección física antes de que se alcance el acceso a la memoria.

Una dirección física, o dirección absoluta, es una ubicación real de la memoria principal.

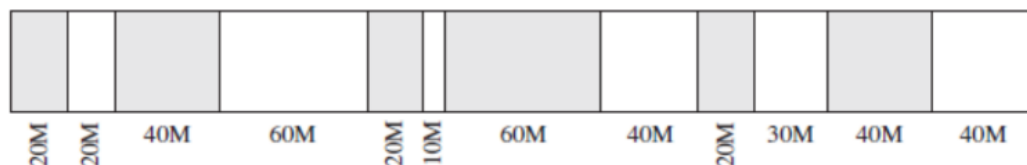
10) Explique cómo funciona paginación de memoria.

Con la paginación simple, la memoria principal se divide en muchos marcos pequeños de igual tamaño. Cada proceso se divide en páginas de igual tamaño; los procesos más pequeños requieren menos páginas, los procesos mayores requieren más. Cuando un proceso se trae a la memoria, todas sus páginas se cargan en los marcos disponibles, y se establece una tabla de páginas. Esta técnica resuelve muchos de los problemas inherentes en el particionamiento.

11) Explique cómo funciona la segmentación de memoria.

Con la segmentación simple, un proceso se divide en un conjunto de segmentos que no tienen que ser del mismo tamaño. Cuando un proceso se trae a memoria, todos sus segmentos se cargan en regiones de memoria disponibles, y se crea la tabla de segmentos.

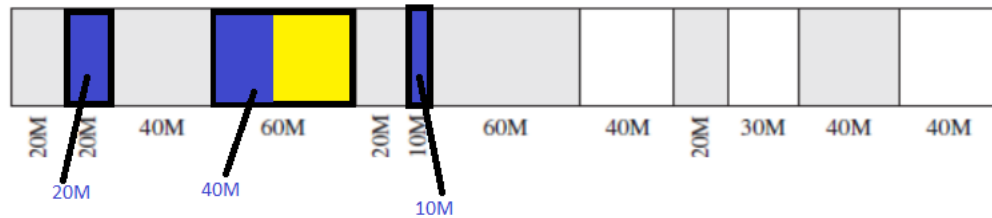
12) Si se utiliza un esquema de particionamiento dinámico y en un determinado momento la configuración de la memoria es la siguiente:



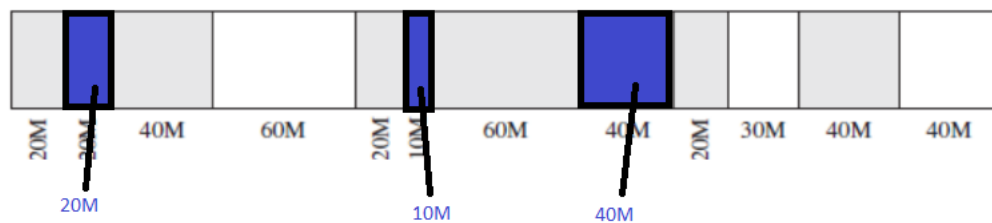
Las áreas sombreadas son bloques asignados; las áreas blancas son bloques libres. Las siguientes tres peticiones de memoria son de 40M, 20M, y 10M.

Indique la dirección inicial para cada uno de los tres bloques utilizando los siguientes algoritmos de colocación:

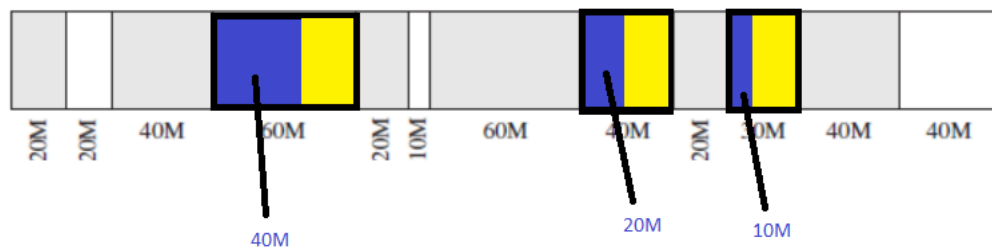
a) Primer ajuste



b) Mejor ajuste



c) Siguiendo ajuste. Asímbase que el bloque añadido más recientemente se encuentra al comienzo de la memoria.



d) Peor ajuste

