**EXPOSICIÓN NO 9. SISTEMA OPERATIVO 1**

**MEMORIA REAL**

La memoria real o principal es en donde son ejecutados los programas y procesos de una computadora y es el espacio real que existe en memoria para que se ejecuten los procesos. Por lo general esta memoria es de mayor [costo](http://www.monografias.com/trabajos7/coad/coad.shtml#costo) que la memoria secundaria, pero el acceso a la información contenida en ella es de más rápido acceso. Solo la [memoria cache](http://www.monografias.com/trabajos5/sisope/sisope.shtml#cache) es más rápida que la principal, pero su costo es a su vez mayor.

Ventajas:

\*Tiempo de ciclo, que representa el intervalo de tiempo mínimo entre dos accesos   sucesivos.

\*Rendimiento, que define el volumen de información intercambiado por unidad de      tiempo, expresado en bits por segundo.

\*No volatilidad, que caracteriza la capacidad de una memoria para almacenar datos      cuando no recibe más electricidad.

Direcciones físicas, lineales y lógicas.

Espacio De Direcciones: son aquellos que apuntan a la memoria para acceder a los datos de esta.

Los espacios de direcciones involucrados en el manejo de la memoria son de tres tipos:

* Direcciones físicas: son aquellas que referencian alguna posicion en la memoria [física](http://www.monografias.com/Fisica/index.shtml). En la memoria física, la dirección accede a la memoria RAM real (los chips RAM incorporados a la placa madre) .
* Direcciones lógicas : son las direcciones utilizadas por los procesos. Sufren una serie de transformaciones , realizadas por el procesador (la MMU), antes de convertirse en direcciones físicas.
* Direcciones lineales: direcciones lineales se obtienen a partir de direcciones logicas tras haber aplicado una transformación dependiente de la [arquitectura](http://www.monografias.com/trabajos6/arma/arma.shtml).

La **unidad de gestión de memoria** o **unidad de manejo de memoria** (mmu).

Es un dispositivo de [Hardware](http://es.wikipedia.org/wiki/Hardware) formado por un grupo de [circuitos integrados](http://es.wikipedia.org/wiki/Circuito_integrado), responsable del manejo de los accesos a la [memoria](http://es.wikipedia.org/wiki/Memoria_de_ordenador) por parte de la CPU. De dar permisos a los procesos para que acedan a los espacios de memorias disponibles o hacinados para ellos.

Entre las funciones de este dispositivo se encuentran la traducción de las direcciones lógicas (o virtuales) a direcciones físicas (o reales), la protección de la memoria, el control de caché y, en arquitecturas de computadoras más simples.

Funciones:

Un beneficio fundamental de la MMU es la posibilidad de implementar protección de memoria, evitando que los programas accedan a porciones de memoria prohibidas. Por ejemplo se puede evitar que un programa acceda o modifique sectores de memoria de otros programas.

 Convertir las direcciones lógicas emitidas por los procesos en direcciones físicas.

 Comprobar que la conversión se puede realizar. La [dirección](http://www.monografias.com/trabajos15/direccion/direccion.shtml) [lógica](http://www.monografias.com/trabajos15/logica-metodologia/logica-metodologia.shtml) podría no tener un dirección física asociada. Por ejemplo, la pagina correspondiente a una dirección se puede haber trasladado a una zona de almacenamiento secundario temporalmente.

 Comprobar que el [proceso](http://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml#PROCE) que intenta acceder a una cierta dirección de memoria tiene permisos para ello.

 La MMU se Inicializa para cada proceso del sistema. Esto permite que cada proceso pueda usar el rango completo de direcciones lógicas ([memoria virtual](http://www.monografias.com/trabajos10/gesmem/gesmem.shtml#VIRTU)), ya que las conversiones de estas direcciones serán distintas para cada proceso.

**PARTICIONES**

Consiste en dividir la memoria libre en varias partes de igual tamaño o de diferentes tamaños. En el caso de la partición fija de partes iguales, se plantean dos dificultades:

Un programa puede ser demasiado grande para caber en la partición. En este caso, el programador debe diseñar el programa mediante superposiciones, para que sólo una parte del programa esté en memoria principal. Cuando se necesita un módulo que no está presente, el programa de usuario debe cargar dicho módulo en la partición del programa, superponiéndose a los programas y datos que se encuentren en ella.

El uso de memoria principal es extremadamente ineficiente. Cualquier programa, sin importar lo pequeño que sea, ocupará una partición completa. Supongamos un programa que ocupa 120 Kb y se carga en una partición de 512 Kb, se malgasta el espacio interno de la partición y ésto se denomina fragmentación interna.

En el caso de particiones de igual tamaño, si todas las particiones estuvieran ocupadas con procesos que no están listos para ejecutar y necesita cargarse un nuevo proceso, debe determinarse qué partición expulsarse de memoria.

Pero en el caso de particiones de distintos tamaños, debe determinarse en que partición (por cuestiones de tamaño) se cargará el nuevo proceso. La forma más simple es asignar cada proceso a la partición más pequeña en que quepa (sin importar si la partición está cargada o no), lo que necesitará de una cola de expulsados para cada partición. Otra forma es asignar el proceso a la partición más pequeña entre aquellas que están libres (usando una única cola para todos los procesos); y si todas las particiones están ocupadas, se debe tomar una decisión de intercambio.

Estamos suponiendo con este método que conocemos la cantidad máxima de memoria que necesitará un proceso, lo que no siempre es cierto.

**PARTICIONES DINÁMICAS**

Este método consiste en particiones de memoria de tamaños variable, o sea, a cada proceso se le asigna la cantidad de memoria que necesita (la cantidad exacta y ni un poco más). Tras el ingreso de nuevos procesos y la expulsión de otras, se generarán huecos de memoria inutilizados. Conforme pasa el tiempo, la memoria comienza a estar más fragmentada y su rendimiento decae; este fenómeno se denomina fragmentación externa.

       Una técnica para superar la fragmentación externa es la compactación. De vez en cuando, el sistema operativo desplaza los procesos para que estén contiguos de forma que toda la memoria libre quede junta en un bloque. Las dificultades que presenta la compactación son:

 ·         Es un procedimiento que consume tiempo de trabajo del procesador

 ·         Debe realizarse los movimientos de una región a otra, sin invalidar las referencias a memoria del programa

A pesar de la compactación, deben utilizarse algunos algoritmos para la ubicación de procesos en las particiones.

**ESTRUCTURAS MAS DINÁMICAS PARA ASIGNAR HUECOS**

El proceso de compactación del punto anterior es una instancia particular del problema de asignación de memoria dinámica, el cual es el cómo satisfacer una necesidad de tamaño n con una lista de huecos libres. Existen muchas soluciones para el problema. El conjunto de huecos es analizado para determinar cuál hueco es el más indicado para asignarse. Las estrategias más comunes para asignar algún hueco de la tabla son:

**PRIMER AJUSTE**

Primer ajuste: Consiste en asignar el primer hueco con capacidad suficiente. La búsqueda puede iniciar ya sea al inicio o al final del conjunto de huecos o en donde terminó la última búsqueda. La búsqueda termina al encontrar un hueco lo suficientemente grande.

**MEJOR AJUSTE**

Mejor ajuste: Busca asignar el espacio más pequeño de los espacios con capacidad suficiente. La búsqueda se debe de realizar en toda la tabla, a menos que la tabla esté ordenada por tamaño. Esta estrategia produce el menor desperdicio de memoria posible.

**PEOR AJUSTE**

Peor ajuste: Asigna el hueco más grande. Una vez más, se debe de buscar en toda la tabla de huecos a menos que esté organizada por tamaño. Esta estrategia produce los huecos de sobra más grandes, los cuales pudieran ser de más uso si llegan procesos de tamaño mediano que quepan en ellos.

Se ha demostrado mediante simulacros que tanto el primer y el mejor ajuste son mejores que el peor ajuste en cuanto a minimizar tanto el tiempo del almacenamiento. Ni el primer o el mejor ajuste es claramente el mejor en términos de uso de espacio, pero por lo general el primer ajuste es más rápido.

**Intercambio swapping**

**Forma de conseguir multiprogramación, utilizando utilizando almacenamiento secundario de apoyo.**

* Fragmentación

***Desaprovechamiento de memoria por haber realizado una mala partición***.

* **Fragmentación interna:**
* **Fragmentación externa:**
* **Desfragmentación**
* **Este proceso consta de ordenar los trozos de información distribuida a través de todo el disco, para mejorar la velocidad de acceso y distribuir de mejor forma el espacio libre del dispositivo.**

Gestión de memoria virtual

Definición

Paginación

Características de la paginación

Se divide la memoria física en bloques de tamaño fijo llamados marcos.

Se divide la memoria Secundaria en bloques de tamaño llamados páginas.

Para correr un programa en páginas de tamaño, se necesitan encontrar n marcos y cargar el programa.

Se establece una tabla de páginas para trasladar las direcciones lógicas a físicas.

Se produce fragmentación interna

Ventajas y Desventajas de la Paginación

Segmentación

SEGMENTACIÓN

SEGMENTACIÓN PAGINADA o PAGINACIÓN Y SEGMENTACIÓN COMBINADA

**Memoria virtual**

* Si el equipo no tiene suficiente memoria de acceso aleatorio (RAM) para ejecutar un programa o una operación, Windows usa la memoria virtual para compensar la falta.

**Protección dentro de una tarea**

* Dentro de una tarea se definen 4 niveles de privilegio de ejecución, para proteger el acceso a parte de la tarea de acuerdo con la sensibilidad de los datos.

**PROTECCIÓN**

* se consigue poniendo cada tarea en espacios de direcciones virtuales diferentes, al asignar un mapa de traducción virtual a física diferente.