



# Instituto Politécnico Nacional

Escuela Superior de Cómputo

---

*Sistemas Operativos*

## *“Tarea 1. Manejo de memoria e Intercambios”*

**Grupo:** 2CM9

**Integrantes:**

- Martínez Coronel Brayan Yosafat
- Monteros Cervantes Miguel Angel
- Ramírez Olvera Guillermo
- Sánchez Méndez Edmundo Josué

**Profesor:** Cortés Galicia Jorge

# Manejo de memoria

Se puede mejorar tanto el grado de utilización del procesador como la velocidad de respuesta del usuario

## Hardware Básico

La mayoría de los procesadores pueden realizar instrucciones simples con el contenido de los registros

La memoria principal únicamente de accede mediante una transición del bus de memoria

El procesador necesita normalmente detenerse, ya que no dispondrá de los datos requeridos para completar la instrucción, surge la **memoria caché**

Tenemos que asegurarnos de que cada proceso disponga de un espacio de memoria separado, teniendo en cuenta el **Registro Base** y el **Registro Límite**

Cualquier intento de acceso a la memoria del sistema operativo o memoria del usuario hará que se produzca una interrupción en el sistema operativo

## Reasignación de direcciones

La primera dirección del proceso de usuario no tiene que ser la 00000

Un compilador se encarga de reasignar las direcciones simbólicas a direcciones reubicables

Un editor de montaje o cargador se encarga de asignar las direcciones reubicables a direcciones absolutas

Se debe tomar en cuenta:

- **Tiempo de compilación**
- **Tiempo de carga**
- **Tiempo de ejecución**

## Espacio de direcciones lógicas y físicas

Una dirección generada por el CPU se le conoce como **Dirección Lógica** mientras que la que se encarga en el registro de direcciones de memoria es la **Dirección Física**

La correspondencia entre la dirección virtual y física en tiempo de ejecución son establecida por un dispositivo llamado **Unidad de Gestión de Memoria**

El programa del usuario utiliza direcciones lógicas y el hardware convierte esas direcciones a físicas

## Montaje dinámico y bibliotecas

El montaje binario es similar a la carga dinámica, solo que lo que se pospone hasta el momento de ejecución es el montaje en lugar de la carga

Con el montaje dinámico se incluye un **stnd** dentro de la imagen binaria para cada referencia a una rutina de la biblioteca

Un **stnd** es un fragmento de código que indica como localizar la rutina o como cargar una biblioteca

Con este mecanismo todos los procesos que utilicen una determinada biblioteca de lenguaje solo necesitara ejecutar una copia del código de la biblioteca

## Carga Dinamica

El tamaño de un proceso esta limitado por el tamaño de memoria física

Para manejar la utilización del espacio de memoria podemos usar memoria de carga dinamica

Una rutina **no se carga hasta que se invoca**, todas se mantienen en un disco en un formato de carga reubicable

Una memoria no utilizada no se carga nunca en memoria

# Intercambios

A esto se le conoce como fragmentación, y existe una regla para ello

Puede haber memoria para un proceso más, pero puede no estar contigua.

Cada ocasión que se cargan procesos, los huecos son más y más pequeños

También un programa podría entrar en 4Mb siendo de 16Mb, administrando muy bien esos 4 Mb

Regla del 50%: Si tenemos N bloques entonces tendremos 0.5 N fragmentos

Al descomponer la memoria como en el mapa de bits, se le llama fragmentación interna

Problemas de los algoritmos

Entre otros como el siguiente ajuste y el menor ajuste

Primer ajuste: toma el primer hueco que sea lo suficientemente grande

Peor ajuste: toma el más grande para crear huecos grandes.

Mejor ajuste: toma el más pequeño que se ajuste, resulta no ser óptimo.

En UNIX solo se activa el intercambio cuando son muchos procesos

Los procesos deben estar en memoria para su ejecución. A veces se sacan y se ponen en suspensión

Procesos

Eso es en esencia un intercambio, y cuando se hacen quedan huecos de tamaño variable

Suelen regresar a la posición que tenían antes. Se puede usar la compactación de memoria (quitar huecos), pero es mucho trabajo

Los procesos pueden crecer si hay un hueco adjunto, sino puede esperar o morir.

Para eso tiene que estar inactivo. Y el cuanto cuanto debe dar tiempo a las operaciones. Y si se relaciona con E/S se deben tener reglas.

Asignación de Memoria Contigua

La memoria suele dividirse en 2: para el sistema operativo y para los procesos del usuario.

La segunda puede administrarse por listas enlazadas o mapas de bits:

La memoria se divide en unidades de asignación, cada una es un bit, 0 marca que está libre, 1 ocupada. Entre mayor el tamaño de unidad menor el mapa y viceversa.

En una lista se enlazan los huecos (llamados segmentos). Con ellos podemos usar algunos algoritmos existentes.