

**Universidad de Guadalajara**



**Centro Universitario de Ciencias Exactas e Ingeniería**

División de Tecnologías para la Integración Ciber-humana

**Materia:** Sistemas Operativos

**Profesor:** Violeta Del Rocío Becerra Velázquez

**Alumno:** Denice Estefania Rico Morones

**Código:** 219421171

**Carrera:** Ingeniería en Computación

**Sección:** D04

**Título:** Administración de memoria

**Fecha:** 27/10/2024

**Contenido**

Técnicas para el manejo de memoria .....	3
Paginación simple: .....	3
Particiones estáticas .....	3
Particiones dinámicas .....	3
Memoria virtual .....	3
Paginación con memoria virtual: .....	4
Segmentación con memoria virtual .....	4
Tabla de paginas.....	4
Buffers: importancia y manejo .....	5
Conclusión:.....	5
Bibliografía: .....	6

## **Técnicas para el manejo de memoria**

- **Particionamiento:** dividimos la memoria física en diferentes partes las cuales serán asignadas a distintos procesos la cual será atendida de manera exclusiva.
- **Paginación:** la memoria se divide y cada división es del mismo tamaño, son más pequeñas ya que se trata de corregir el problema de la fragmentación interna, se carga en los marcos disponibles.
- **Segmentación:** el proceso se divide en segmentos ya sean los que se necesiten y del tamaño que se requieran, los procesos deben de entrar en su totalidad si no, no pueden ingresar.

### **Paginación simple:**

La memoria se divide en distintos fragmentos del mismo tamaño y se cargan en todos sus espacios disponibles; puede llegar a existir la fragmentación interna ya que ese espacio puede no ser del tamaño exacto del tamaño de la página, aun así tiene sus ventajas como lo es el no tener fragmentación externa ya que las páginas no se encuentran siguiendo un orden en la memoria o no necesitan estar juntas.

### **Particiones estáticas**

La memoria principal se divide en distintos fragmentos estáticos, solamente se dispone de un proceso por partición, al ser unas de las primeras técnicas era muy común que se contara con un tamaño muy reducido en memoria. Aquí no disponemos de toda la memoria; estos procesos se traducían con ensambladores y compiladores para poder ejecutarlos dentro de una partición específica; si un proceso quiere ingresar debe de haber una partición disponible, se desperdicia demasiado espacio en memoria (fragmentación).

### **Particiones dinámicas**

La memoria se asigna conforme se tienen los procesos aun así, no se debe superar el tamaño en memoria que se tiene, existía un grado mínimo de fragmentación conforme los procesos iban saliendo, y estos espacio podría ser utilizados por otros procesos pero esto ocasionaba un problema, al tener procesos pequeños y liberarse ese espacio quedaba inmerso para las particiones de más tamaño, al ser fragmentos pequeños de memoria no se podían unir para formar particiones más grandes.

## **Memoria virtual**

El sistema operativo tiene la habilidad de asignar la memoria suficiente a cierto número de procesos pero cuando se excede físicamente esa capacidad se tiene a lo que llamamos *memoria virtual*, para esto se tiene en cuenta los espacios libres que no están utilizando en su momentos para asignarlos a ese almacenamiento secundario. Esto surge gracia a la aparición de los simuladores de SO y máquinas virtuales.

### **Paginación con memoria virtual:**

Esta es similar a la paginación simple solamente que la diferencia está en que no es necesario cargar todas las páginas de un proceso; las páginas necesarias solamente se cargan en memoria física cuando se requieren mientras que las que están inactivas se mantienen en el disco.

### **Segmentación con memoria virtual**

Es similar a la segmentación, pero de igual forma no es necesario cargar todos los segmentos de un proceso, en esta misma no existe la fragmentación interna y por ello mismo hay un mayor grado de multiprogramación. El sistema operativo mantendrá una tabla de segmentos que registra la dirección y el límite de cada segmento para así tener un acceso controlado a estos y aislando a los procesos entre sí.

### **Tabla de paginas**

El sistema operativo mantiene una tabla de páginas para mostrar la ubicación del marco por cada página del proceso y está conformada por un número de páginas y un desplazamiento dentro de esta, Con la paginación el procesador debe conocer cómo acceder a la tabla del proceso actual esto se hace gracias a una dirección lógica (**número de página, desplazamiento**), y el procesador utiliza esta tabla para producir una dirección física (**número de marco, desplazamiento**).

Podemos resumir esto a que en dicha tabla indicamos el número de página, el proceso que se ha asignado y los atributos de la misma administración también se puede indicar si la memoria es compartida y los datos para los algoritmos de intercambio de almacenamiento secundario y el valor para el registro base del CPU; si se desea hacer cualquier operación de memoria y en otra página diferente debemos de modificar el valor en el registro de su tabla de páginas.

Con la tabla de páginas podemos traducir los identificadores de página los cuales son utilizados por programas en sus direcciones lógicas y por las direcciones de marco de página esto al usarse en las direcciones físicas al momento de querer

acceder a la memoria, Otras cosas a las cuales también podemos acceder son las siguientes:

- Una bandera para indicar si ha sido referenciada.
- Una bandera para indicar si la página ha modificado su contenido desde La última vez que se ha escrito.
- Información de protección lo cual es una serie de bits para controlar los aspectos ya sea si queremos que la página sea solamente para lectura, si se trata de un código de programa, si permitimos modificaciones o si será de memoria compartida.

### **Buffers: importancia y manejo**

los buffers al ser partes temporales de almacenamiento mientras existen un intercambio de datos entre diferentes partes de un sistema son de suma importancia si queremos mejorar la eficiencia y estabilidad en el procesamiento de los datos; los buffers realizan sincronización entre las velocidades para poder trabajar juntos sin interrupciones, además de reducir la latencia, evitar la pérdida de datos y la eficiencia en operaciones ya sean de entrada/salida.

- Determinar el tamaño óptimo del buffer.
- Es muy común que los buffers funcionen con estructuras FIFO.
- Utilizar mecanismos de sincronización es muy importante cuando múltiples procesos acceden al buffer.
- Es importante liberar el espacio del buffer para evitar los problemas de rendimiento o overflow.

### **Conclusión:**

Cómo conclusión puedo decir que esta actividad ha sido muy importante ya que he aprendido nuevos conceptos y cómo trabajan todos estos en conjunto, cómo es que han ido evolucionando los métodos de gestión de memoria conforme el tiempo y las nuevas necesidades de los sistemas; es de suma importancia conocer cómo poder realizar procesos de una manera más óptima para así aprovechar los recursos cómo se debería, estos conceptos son aplicables en muchas áreas o así lo considero yo así que esta información será de mucha ayuda estar implementarlo tanto en esta materia como en alguna otra.

**Bibliografía:**

- Llaven, D. S. (2015). *Sistemas operativos. Panomarama para ingeniería en computación e informática*. México: Patria.
- Martinez, D. L. (s.f.). *Sistemas Operativos*. Argentina: Universidad Nacional del Noroeste.
- Stallings, W. (2005). *Sistemas operativos. Aspectos internos y principios de diseño*. Madrid: Pearson Education.