

# Arquitectura y Sistemas Operativos

---

UTN MAR DEL PLATA

GESTIÓN DE MEMORIA – MEMORIA VIRTUAL

# INTRODUCCIÓN

---

*Comparando la paginación simple y la segmentación simple, tenemos una distinción entre particionamiento estático y dinámico, además tenemos los fundamentos iniciales de la gestión de memoria. Las características tanto de la paginación como la segmentación que son claves para este inicio son:*

- 1. Todas las referencias a la memoria dentro de un proceso se realiza a direcciones lógicas, que se traducen dinámicamente a direcciones físicas durante la ejecución.*
- 2. Un proceso puede dividirse en varias porciones (páginas o segmentos) y estas porciones no tiene que estar localizadas en memoria de forma continua durante la ejecución.*

# INTRODUCCIÓN

---

*Si las dos características anteriores se dan, entonces es necesario que todas las páginas o todos los segmentos de un proceso se encuentren en la memoria principal durante la ejecución. Si la porción (segmento o página) en la que se encuentra la siguiente instrucción a buscar está y si la porción donde se encuentra la siguiente dirección de datos que se va a acceder también está, entonces al menos la siguiente instrucción se podrá ejecutar.*

# INTRODUCCIÓN

---

Supongamos que se tiene que traer un nuevo proceso de memoria. El SO comienza trayendo únicamente una o dos porciones. Que incluye la porción inicial del programa y la porción inicial de datos sobre la cual acceden las primeras instrucciones. Esta parte del proceso que se encuentra en memoria principal se denomina **conjunto residente** del proceso. Cuando el proceso se está ejecutando, las cosas ocurren de forma suave mientras que todas las referencias a la memoria se encuentren dentro del conjunto residente. Usando una tabla de páginas o segmentos, el procesador siempre es capaz de determinar si esto es así.

# INTRODUCCIÓN

---

*Si el procesador encuentra una dirección lógica que no se encuentra en la memoria principal, generará una interrupción indicando un fallo de acceso a la memoria. El SO coloca al proceso interrumpido en estado bloqueado y toma el control. Para que la ejecución del proceso pueda reanudarse más adelante, el SO necesita traer a memoria principal la memoria del proceso que contiene la dirección lógica que ha causado el fallo de acceso.*

*Para eso, el SO realiza una petición de E/S al disco, luego el SO puede activar otro proceso para que se ejecute mientras el disco realiza la operación de lectura. Una vez que la porción solicitada se ha traído a la memoria principal, una nueva interrupción de E/S se lanza, dando el control al SO que coloca al proceso afectado en el estado Listo.*

# INTRODUCCIÓN

---

Aunque se puede cuestionar la eficiencia de realizar esta tarea, ya que el proceso necesita ser interrumpido, veremos dos implicaciones que justifican el uso de este sistema:

1. Pueden mantenerse un mayor número de procesos en memoria principal.
2. Un proceso puede ser mayor que toda la memoria principal.

Debido a que un proceso se ejecuta sólo en memoria principal, esta memoria se denomina **memoria real**. El programador o usuario perciben una memoria potencialmente mucho más grande (la cual se encuentra en disco). Esta última se denomina **memoria virtual**.

# PAGINACIÓN

---

*El término **memoria virtual** se asocia habitualmente con sistemas que emplean paginación, a pesar de que la memoria virtual basada en segmentación también se utiliza.*

*Cuando vimos paginación simple, indicamos que cada proceso dispone de su propia tabla de páginas, y que todas las páginas se encuentran localizadas en memoria principal. Cada entrada en la tabla consiste en un número de marco de la correspondiente página. Para la memoria virtual basada en paginación, también se necesita una tabla de páginas, aunque es un poco más compleja.*

# PAGINACIÓN

---

*Debido a que sólo algunas páginas se encuentran en memoria principal, se necesita que cada entrada en la tabla de páginas indique si la correspondiente página esta presente (P) en memoria principal. Además, la entrada de la tabla de páginas incluye un bit de modificado (M), que indica si los contenidos de la correspondiente página han sido alterados desde que la página se cargó por última vez en la memoria principal.*



# PAGINACIÓN

## *Estructura de la tabla de páginas*

Dirección virtual

Número de página	Desplazamiento
------------------	----------------

Entrada de la tabla de páginas

P	M	Otros bits de control	Número de marco
---	---	-----------------------	-----------------

# SEGMENTACIÓN

*La segmentación permite al programador ver la memoria como si se tratase de diferentes espacios de direcciones o segmentos.*

---

*Esta organización tiene un gran número de ventajas para el programador sobre los espacios de direcciones no segmentados:*

- 1. Simplifica el tratamiento de estructuras de datos que puedan crecer. Con memoria virtual segmentada, a una estructura de dato se le puede asignar su propio segmento, y el SO expandirá y reducirá el segmento bajo demanda.*
- 2. Permite programas que se modifican o recopilan de forma independiente.*
- 3. Da soporte a la compartición entre procesos.*
- 4. Soporta mecanismos de protección.*

# SEGMENTACIÓN

---

*En segmentación simple, indicamos que cada proceso tiene su propia tabla de segmentos. Cada entrada en la tabla contiene la dirección de comienzo del correspondiente segmento en la memoria principal, así como la longitud del mismo.*

*En el caso de memoria virtual basada en segmentación, son un poco más complejas. Debido a que sólo algunos de los segmentos del proceso pueden encontrarse en la memoria principal, se necesita un bit para cada entrada de la tabla para indicar si el correspondiente segmento se encuentra presente en la memoria principal.*

*Otro bit de control es el bit de modificado, que tiene la misma funcionalidad que en la tabla de paginas.*

# SEGMENTACIÓN

## *Estructura de la tabla de segmentos*

Dirección virtual

Segmento de página	Desplazamiento
--------------------	----------------

Entrada de la tabla de segmentos

P	M	Otros bits de control	Longitud	Comienzo de segmento
---	---	-----------------------	----------	----------------------

# PAGINACIÓN Y SEGMENTACIÓN COMBINADAS

---

*Paginación y segmentación, cada una tiene sus propias ventajas. La paginación es transparente al programador y elimina la fragmentación externa, y por lo tanto proporciona un uso eficiente de la memoria principal. La segmentación sí es visible al programador y tiene la capacidad de manejar estructuras de datos que crecen, modularidad y dar soporte a la compartición y a la protección. Para combinar las ventajas de ambos, algunos sistemas por medio del hardware del procesador y del soporte del SO son capaces de proporcionar ambos.*

# PAGINACIÓN Y SEGMENTACIÓN COMBINADAS

---

*En un sistema combinado de segmentación /paginación, el espacio de direcciones del usuario se divide en un número de segmentos. Cada segmento es dividido en un número de páginas de tamaño fijo, que son del tamaño de los marcos de la memoria principal. Desde el punto de vista del programador, una dirección lógica sigue conteniendo un número de segmento y un desplazamiento dentro del segmento. Desde el punto de vista del sistema, el desplazamiento dentro del segmento es visto como un número de página y un desplazamiento dentro de la página incluida en el segmento.*

# PAGINACIÓN Y SEGMENTACIÓN COMBINADAS

## *Estructura de la tabla de segmentación/paginación*

Dirección virtual

Segmento de página	Número de página	Desplazamiento
--------------------	------------------	----------------

Entrada de la tabla de segmentos

Bits de control	Longitud	Comienzo de segmento
-----------------	----------	----------------------

Entrada de la tabla de páginas

P	M	Otros bits de control	Número de marco
---	---	-----------------------	-----------------

P = bit de presente  
M = bit de modificado

# SEGMENTACIÓN Y PAGINACIÓN COMBINADAS

*Traducciones de dirección virtual o lógica a dirección física:*

