

## 2.4.1 Hardware y Estructuras de Control

- Las referencias de memoria se traducen a direcciones físicas dinámicamente en tiempo de ejecución
  - Un proceso puede ser intercambiado hacia dentro o fuera de la memoria principal ocupando diferentes regiones
- Un proceso se puede dividir en trozos que no sean asignados de forma contigua en memoria principal
  - Todos los trozos de un proceso no necesitan estar cargados en memoria principal durante la ejecución

## Ejecución de un Programa

- El sistema operativo trae a memoria principal unos cuantos trozos del programa
- Conjunto Residente: porción del proceso que está en memoria principal
- Se genera una interrupción cuando se necesita una dirección que no está en memoria principal
- El sistema operativo pone al proceso en un estado de bloqueado
- Se trae a memoria principal la parte del proceso que contiene la dirección lógica
  - El S.O. emite una petición de lectura al disco
  - Otro proceso entra en ejecución (se despacha) mientras la operación de E/S tiene lugar
  - Cuando se completa la lectura del disco, se produce una interrupción que hace que el S.O. ponga en estado activo (ready) al proceso afectado

## Ventajas de Dividir un Proceso

- Se puede mantener en memoria principal más procesos
  - Sólo se carga algunas partes de cada proceso
  - Con tantos procesos en memoria principal, más probable es que haya un proceso en estado de listo (ready) en cualquier instante.
- Un proceso puede ser mayor que toda la memoria principal

## Tipos de Memoria

- Memoria Real
  - Memoria principal
- Memoria Virtual
  - Memoria en disco
  - Permite la multiprogramación eficaz y libera al usuario de la limitación en tamaño de la memoria principal

## Soporte Necesario para Memoria Virtual

- El hardware debe soportar paginación y segmentación

- El sistema operativo debe ser capaz de gestionar el movimiento de páginas y/o segmentos entre la memoria secundaria y la memoria principal

## Paginación

- Cada proceso tiene su propia tabla de páginas
- Cada entrada en la tabla de páginas contiene el número del marco en memoria principal de la página correspondiente
- Se necesita un bit para indicar cuando la página está en memoria principal o no

## Segmentación

- Pueden ser diferentes, tamaño dinámico
- Simplifica la gestión de estructuras de datos que crecen
- Permite a los programas ser modificados y recompilados independientemente
- Presta compartición de datos entre procesos
- Presta protección

## Segmentación y Paginación combinada

- La paginación es transparente al usuario
- La paginación elimina la fragmentación externa
- La segmentación es visible al programador
- La segmentación permite estructuras de datos que crecen, modularidad y da soporte a la compartición y protección
- Cada segmento se divide en páginas de tamaño fijo

## 2.4.3 SOFTWARE DEL SISTEMA OPERATIVO

El diseño del gestor de memoria en un sistema operativo depende de tres áreas fundamentales de decisión:

- Si se emplean o no técnicas de memoria virtual.
- El uso de paginación, segmentación o ambas.
- Los algoritmos empleados para los problemas de la gestión de memoria.

Las decisiones tomadas en las dos primeras áreas dependen de la plataforma de hardware disponible. Las decisiones del tercer punto (los algoritmos) son del dominio del software del S.O. En las políticas, elementos claves de diseño, que luego se desarrollarán, el punto clave es el rendimiento: se busca minimizar el porcentaje de fallos de página. Los fallos de página originan una considerable sobrecarga en el software. Como mínimo, la sobrecarga incluye la decisión que el S.O. debe tomar

sobre las páginas residentes o páginas que se deben reemplazar y la E/S de intercambiar páginas. Además, el S.O. debe planificar la ejecución de otro proceso durante el intercambio de página, dando lugar a un cambio de proceso.

Los elementos clave del diseño son los desarrollados a continuación:

## POLÍTICAS DE LECTURA (FETCH)

Es la que está relacionada con la decisión de cuándo se debe cargar una página en la memoria principal. Las dos alternativas más comunes son:

- *Paginación por demanda:* Se trae una página a la memoria principal sólo cuando se hace referencia a una posición en dicha página.
- *Paginación previa:* Se cargan otras páginas distintas a las demandadas debido a un fallo de página. Esta política no es efectiva si la mayoría de las páginas extra que se traen no se referencian.

## POLÍTICAS DE UBICACIÓN

Tiene que ver con determinar dónde va a residir una parte de un proceso en la memoria principal. Es un aspecto importante del diseño en un sistema puro de segmentación.

Como posibles alternativas se tiene:

- Políticas del mejor ajuste.
- Políticas del primer ajuste
- Y otras.

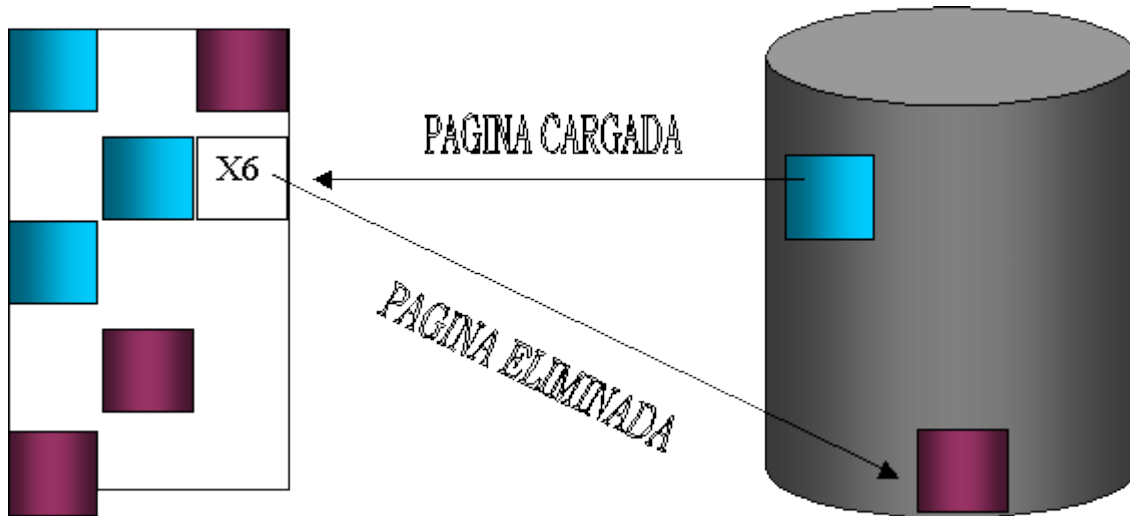
Sin embargo, para un sistema que usa tanto paginación pura como paginación combinada con segmentación, la ubicación carece de importancia puesto que el hardware de traducción de direcciones y el hardware de acceso a la memoria principal pueden desarrollar sus funciones en cualquier combinación de marco de página con idéntica eficiencia.

## POLÍTICAS DE REEMPLAZO

Trata de la selección de la página a reemplazar en la memoria principal cuando se debe cargar una nueva página. Esto resulta difícil de explicar por lo que se debe tener en cuenta algunos conceptos interrelacionados como ser:

- El número de marcos de página a asignar a cada proceso activo.

- Si el conjunto de páginas a considerar para el reemplazo debe limitarse a las del proceso que provocó el fallo de página o abarcar todos los marcos de página situados en la memoria principal.
- De entre el conjunto de páginas consideradas, cuál es la página específica que debe elegirse para el reemplazo.



## Control de Carga

- Determina el número de procesos que estarán residentes en memoria principal
- Demasiado pocos, mayor probabilidad de que todos estén bloqueados y mucho tiempo dedicado al intercambio de procesos (swap)
- Demasiados procesos dan lugar a sobrepaginación (thrashing)

## Política De Vaciado

Una política de vaciado es la contraria a una política de lectura: se preocupa de determinar el momento en que hay que escribir en la memoria secundaria una página modificada. Las dos alternativas son el vaciado por demanda y el vaciado previo. Con **vaciado por demanda**, una página se escribirá en la memoria secundaria sólo cuando haya sido elegida para reemplazarse. Una política de **vaciado previo** escribe las páginas modificadas antes de que se necesiten sus marcos, de forma que las páginas pueden escribirse por lotes.