Gestión de Memoria

6

Sistemas Operativos y Distribuidos

Prof. Javier Echaiz

D.C.I.C. – U.N.S.

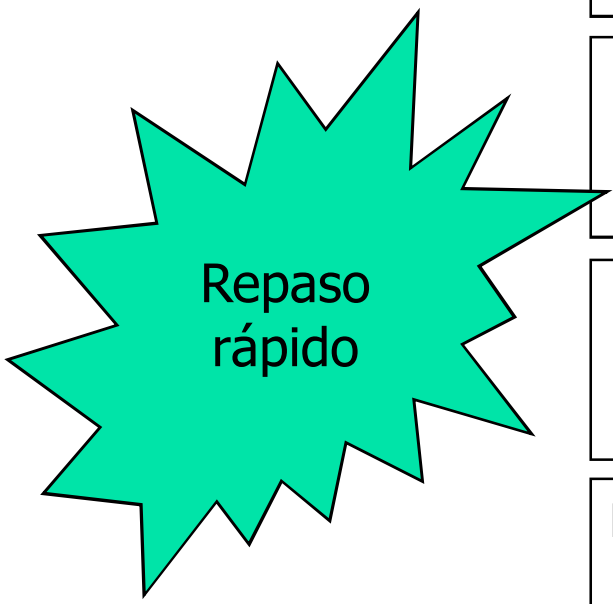
http://cs.uns.edu.ar/~jechaiz

je@cs.uns.edu.ar

Mapa Conceptual de esta parte de la clase

<table>
<tr><td>Real</td><td colspan="2">Real</td><td colspan="2">Real</td><td colspan="2">Virtual</td></tr>
<tr><td>Mono Usuario</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td></tr>
<tr><td rowspan="3"></td><td colspan="2">Particionamiento</td><td>Paginación Simple</td><td>Segmentación Simple</td><td>Paginación Virtual</td><td>Segmentación Virtual</td></tr>
<tr><td>Fija</td><td>Dinámica</td><td colspan="2">Combinación</td><td colspan="2">Combinación</td></tr>
<tr><td colspan="6">Reubicación, Protección</td></tr>
</table>

Organización Física de la Memoria



Registros de la CPU

Cache
Nivel 1
Nivel 2

RAM
Memoria Física
Memoria Virtual

Almacenamiento
Temporal o
de acceso aleatorio

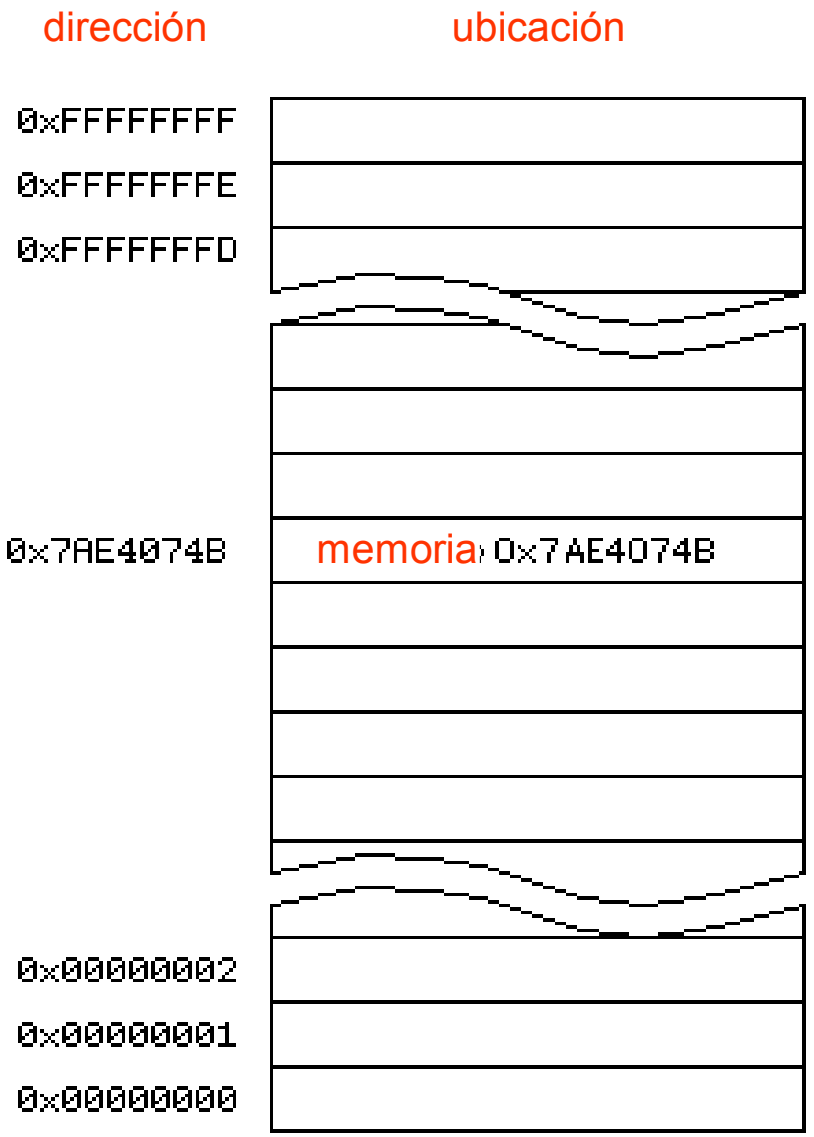
Hard Disk
Drive
Flash Memory

Almacenamiento
Permanente

Dispositivos Externos
Teclado
Mouse
Scaner

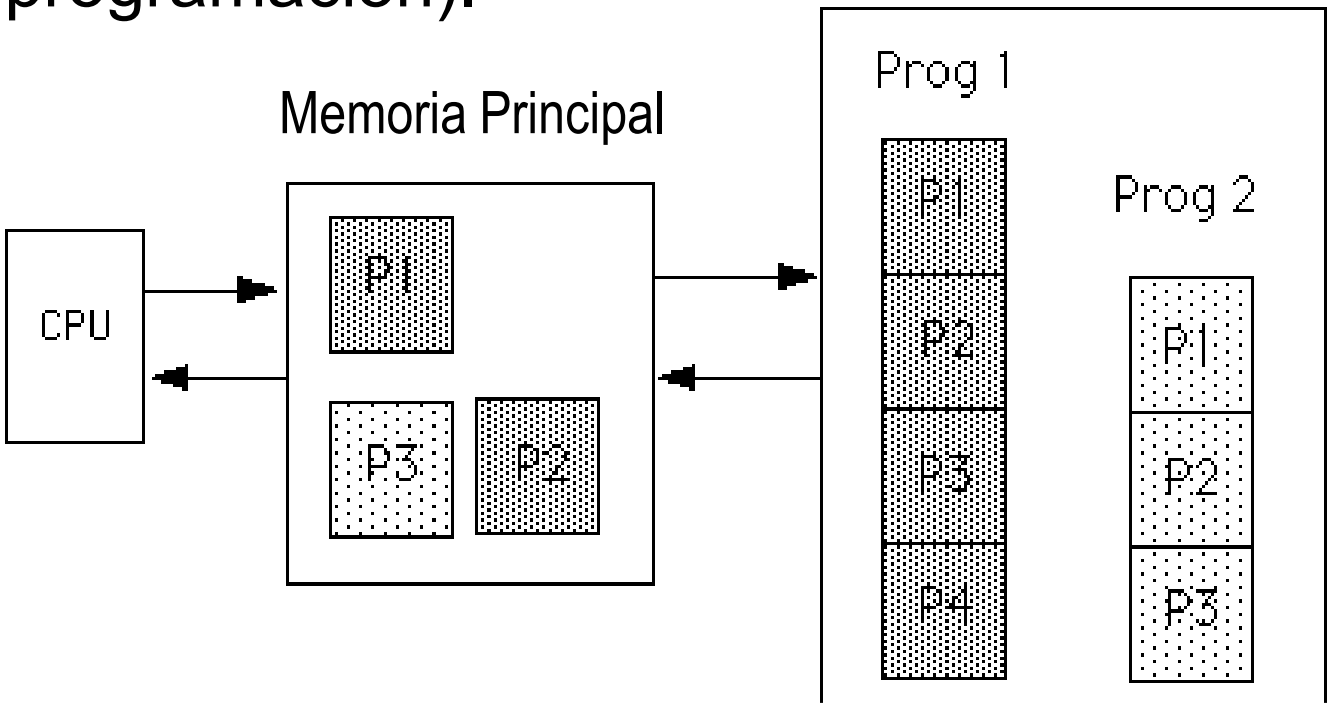
Organización Lógica de la Memoria

- La memoria principal es un arreglo de palabras o bytes, cada uno de los cuales tiene una dirección (espacio de direcciones).
- La interacción es lograda a través de un conjunto de lecturas y escrituras a direcciones especificas realizadas por los procesos.



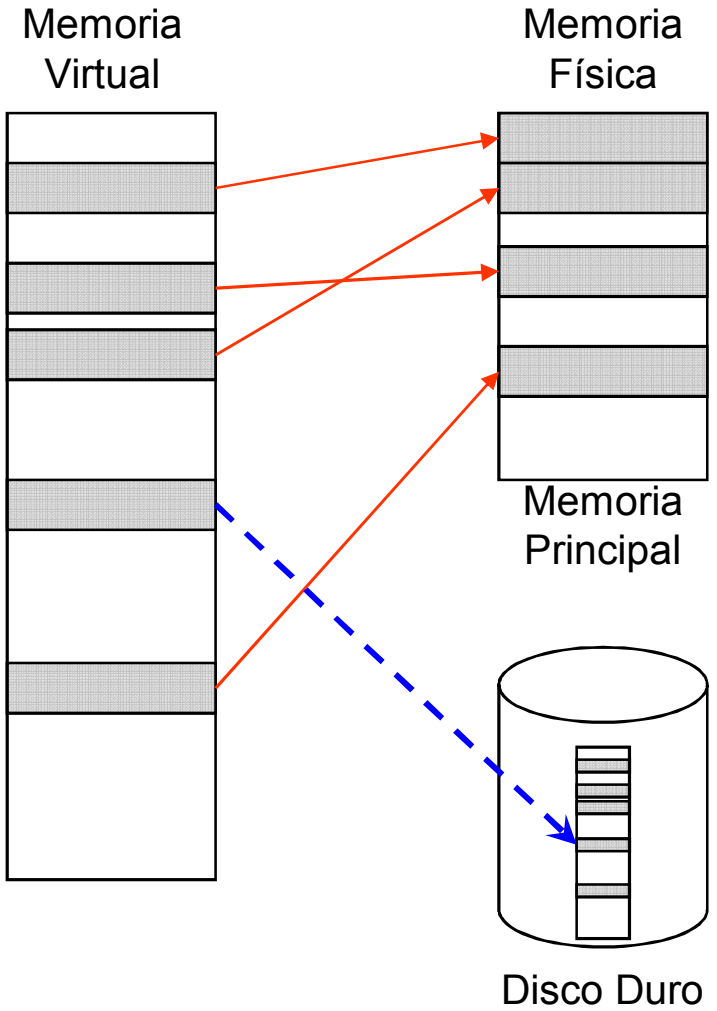
Procesos y Memoria

- Para que un proceso se ejecute se requiere ubicarlo en memoria principal junto con los datos que direcciona.
- Para optimizar el uso de la computadora se requiere tener varios procesos en memoria principal (grado de multiprogramación).

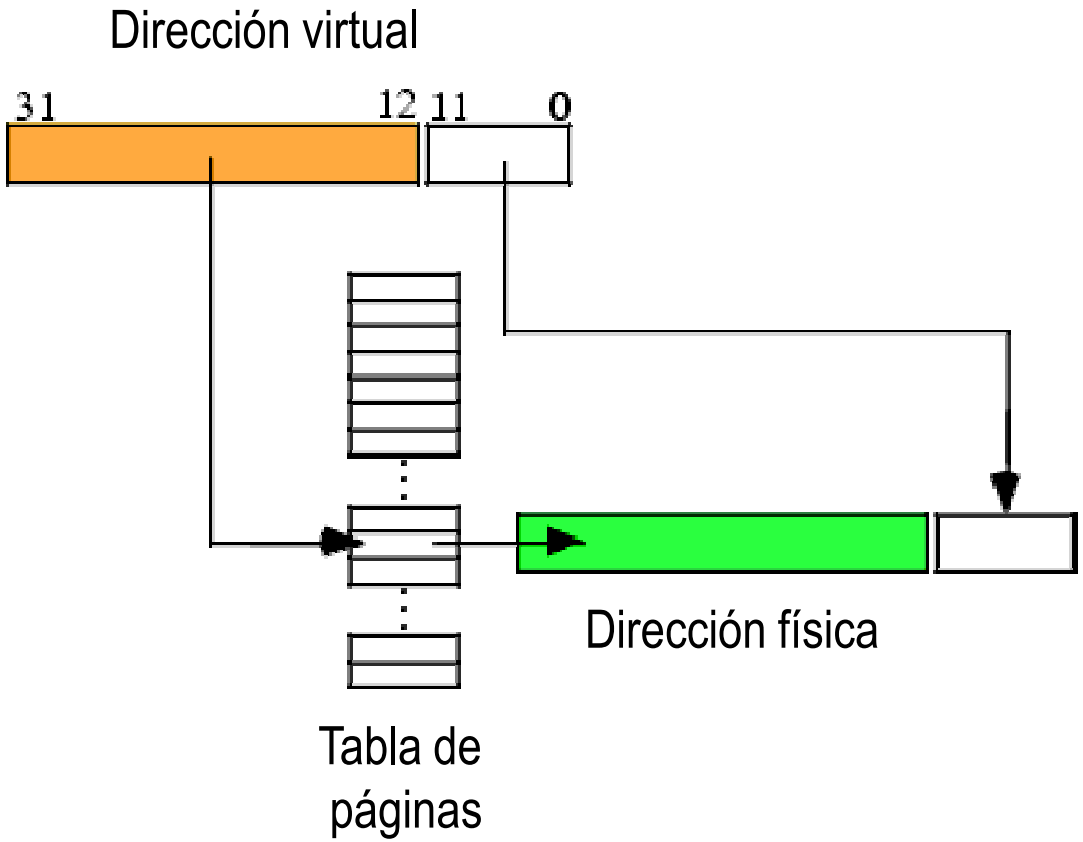


Memoria Virtual

- La memoria principal es pequeña como para acomodar todos programas y datos permanentemente.
- Por lo que es necesario implementar mecanismos de memoria virtual.
- La memoria virtual es una técnica para dar la ilusión de tener más memoria que la memoria principal.

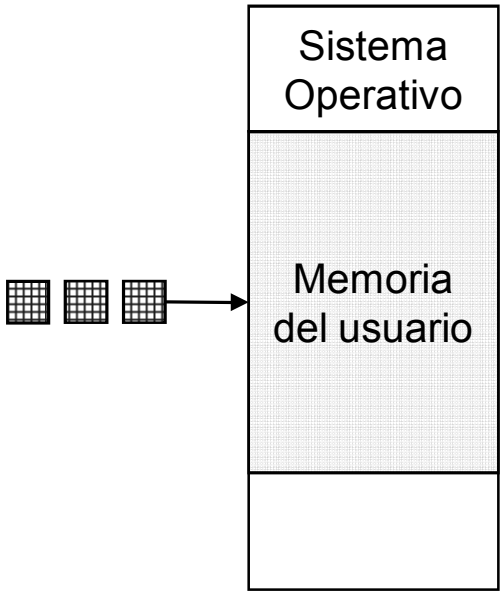


Administrador de memoria



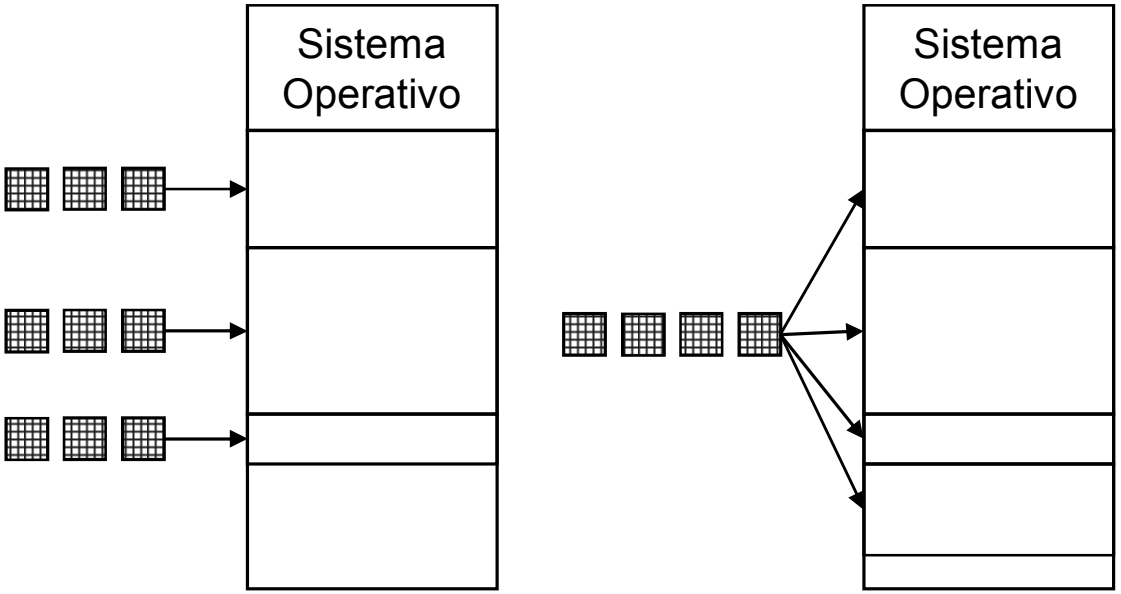
Administrador de Memoria

Sistema monoprogramado



Un programa puede o no ingresar a una única partición de memoria

Sistema multiprogramado



Múltiples programas comparten diversas particiones de memoria
Particiones de tamaño fijo
Particiones de tamaño variable

Administrador de Memoria

- El administrador de memoria tiene como objetivos:
  - Ubicar, reemplazar, cargar y descargar procesos en la memoria principal.
  - Proteger la memoria de acceso indeseados (accidentales o intencionados).
  - Permitir la compartición (*sharing*) de zonas de memoria (indispensable para lograr la cooperación de procesos).

Requisitos del administrador de memoria

1. **Reubicación.** Permitir el recalculo de direcciones de memoria de un proceso reubicado.
2. **Protección.** Evitar el acceso a posiciones de memoria sin el permiso expreso. (no direcciones absolutas).
3. **Sharing.** Permitir a procesos diferentes acceder a la misma porción de memoria.
4. **Organización Lógica.** Permitir que los programas se escriban como módulos compilables y ejecutables por separado.
5. **Organización Física.** Permitir el intercambio de datos en la memoria primaria y secundaria

Estrategias

Están dirigidas a la obtención del mejor uso del recurso memoria principal, estas pueden ser:

1. Estrategia de solicitud (búsqueda)

   (cuando obtener un fragmento de programa)

   - Estrategias de búsqueda por demanda.
   - Estrategias de búsqueda anticipada.

2. Estrategia de ubicación.

   (donde se colocará (cargar) un fragmento de programa nuevo)

3. Estrategia de reposición.

   (qué fragmento de programa elimina, para cargar uno nuevo)

Administrador de Memoria

- Las técnicas usadas son las siguientes:
  1. Partición Fija
  2. Partición Dinámica
  3. Paginación Simple
  4. Segmentación Simple
  5. Memoria Virtual Paginada
  6. Memoria Virtual Segmentada

TECNICAS DE ADMINISTRACION DE MEMORIA

PARTICIONAMIENTO

Real	Real		Real		Virtual	
Mono Usuario	Multiprogramación		Multiprogramación		Multiprogramación	
	Particionamiento		Paginación Simple	Segmentación Simple	Paginación Virtual	Segmentación Virtual
	Fija	Dinámica	Combinación		Combinación	
	Reubicación, Protección					

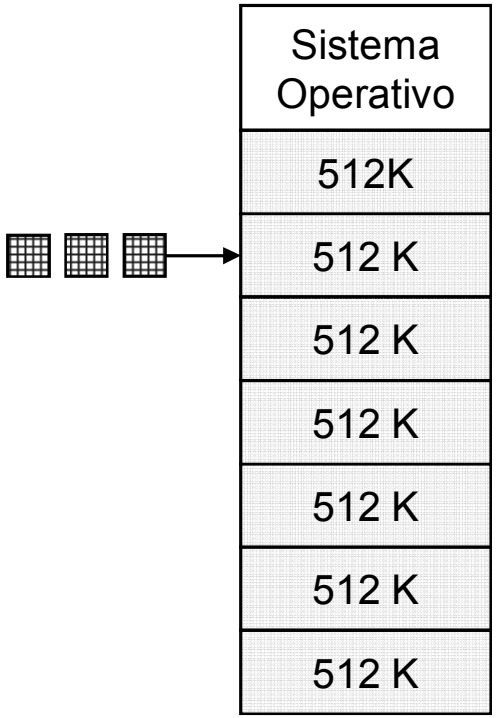
1. Partición Fija

- La memoria principal se divide en un conjunto de particiones de tamaño fijo durante el inicio del sistema.
- Un proceso se puede cargar completamente en una partición de tamaño menor o igual.
- Ventajas. Sencilla de implementar. Poca sobrecarga al SO.
- Desventajas. Fragmentación interna. Nro. fijo de procesos activos.

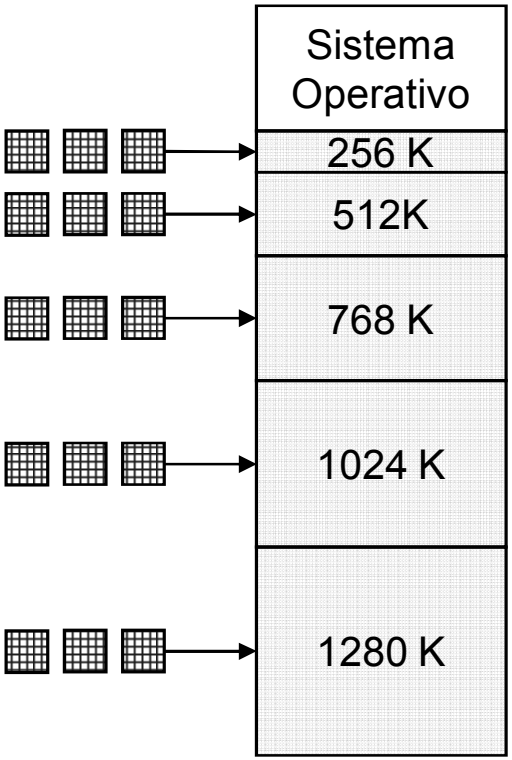
1. Estrategias

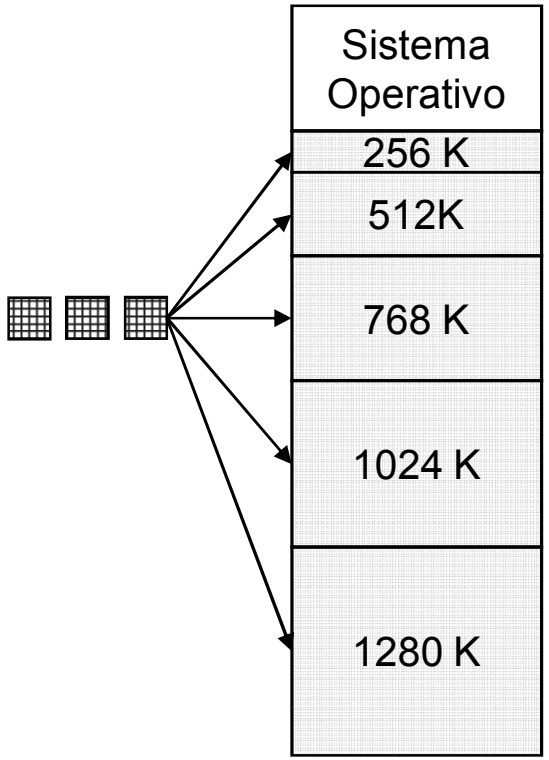
- Solicitud.
  - Por demanda
- Ubicación.
  - Partición de igual tamaño.
    - Si el proceso cabe en una partición se puede cargar
  - Partición de diferente tamaño.
    - Asignar a la partición más pequeña.
    - Se genera dos tipos de colas: una cola, varias colas
- Reemplazo.
  - Uno de los proceso se saca, según el planificador.

1. Estrategia de Ubicación



Particiones del mismo tamaño



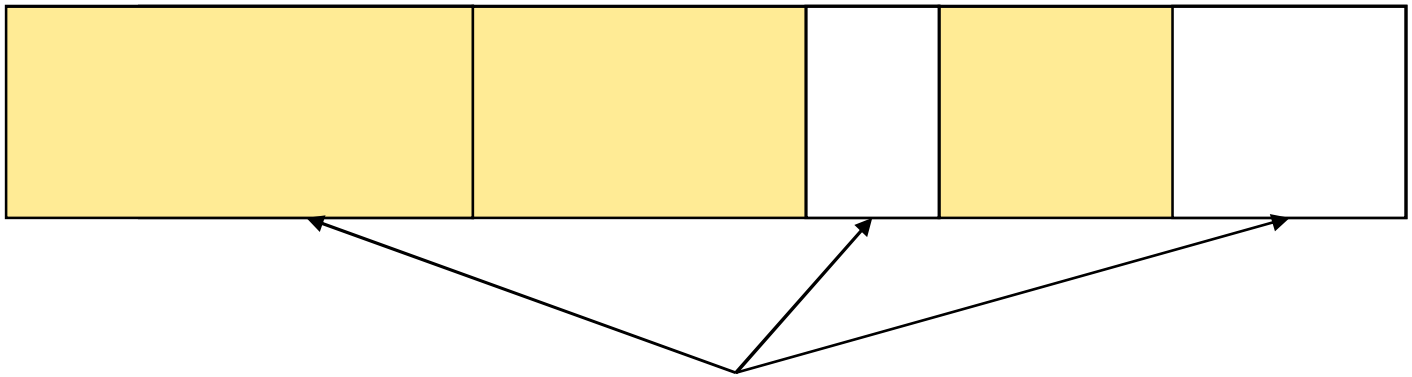


Particiones de distinto tamaño

1. Partición Fija

- Si un programa no cabe en una partición, el programador debe diseñarlo en módulos cargables.

- El uso de la memoria es muy ineficiente, no importa el tamaño del proceso, ocupara toda la partición, se genera fragmentación interna.



fragmentación interna

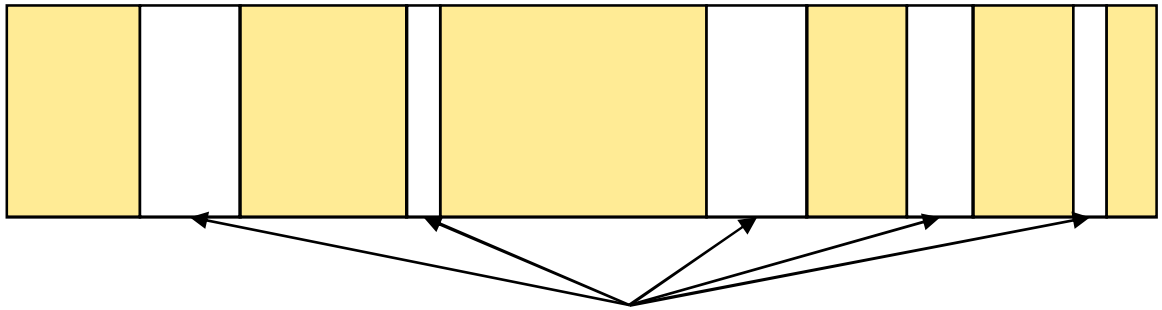
2. Partición Dinámica

- Las particiones se crean dinámicamente por demanda.
- Son variables en tamaño y número.
- Cada proceso se carga completamente en una única partición del tamaño del proceso.

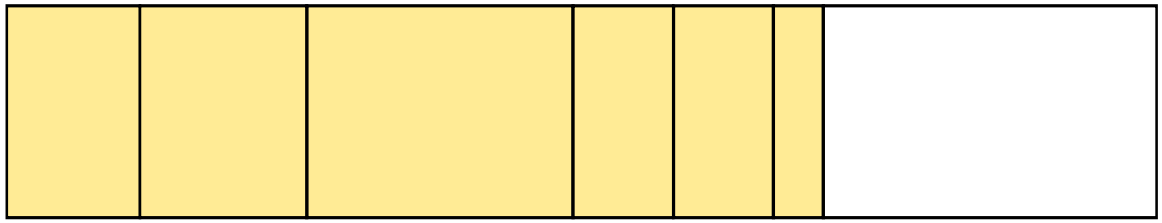
- Ventajas. No existe fragmentación interna.
- Desventajas. Fragmentación externa. Se debe compactar la memoria. El compactado toma tiempo.

2. Partición Dinámica

- El uso de la memoria es muy ineficiente, se generan muchos huecos entre las particiones, cada vez más pequeñas, se genera la fragmentación externa.
- Cada cierto tiempo se debe compactar los segmentos libres, para que estén contiguos.



fragmentación externa



compactación

2. Estrategias

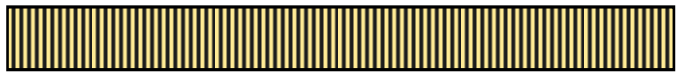
- Solicitud.
  - Por demanda
- Ubicación.
  - Primer ajuste. El primer bloque disponible que ubique (parte del inicio)
  - Siguiente ajuste. El siguiente bloque disponible que ubique (parte desde la ubicación actual)
  - Mejor ajuste. El bloque disponible que deje el menor espacio libre (búsqueda exhaustiva)
- Reemplazo.
  - Uno de los proceso se saca, según el planificador.

2. Estrategias

- Primer ajuste. Es bueno, con baja compactación. Puebla el inicio de la memoria.

- Siguiente ajuste. Puebla el final de la memoria, el siguiente bloque libre siempre está al final de la memoria.

- Mejor ajuste. Tiene peores resultados, dado que busca la partición que deje el hueco más pequeño, la memoria se llena de huecos pequeños. Se compacta con más frecuencia



TECNICAS DE ADMINISTRACION DE MEMORIA

PAGINACION Y SEGMENTACION SIMPLE

<table>
<tr><td>Real</td><td colspan="2">Real</td><td colspan="2">Real</td><td colspan="2">Virtual</td></tr>
<tr><td>Mono Usuario</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td></tr>
<tr><td rowspan="3"></td><td colspan="2">Particionamiento</td><td>Paginación Simple</td><td>Segmentación Simple</td><td>Paginación Virtual</td><td>Segmentación Virtual</td></tr>
<tr><td>Fija</td><td>Dinámica</td><td colspan="2">Combinación</td><td colspan="2">Combinación</td></tr>
<tr><td colspan="6">Reubicación, Protección</td></tr>
</table>

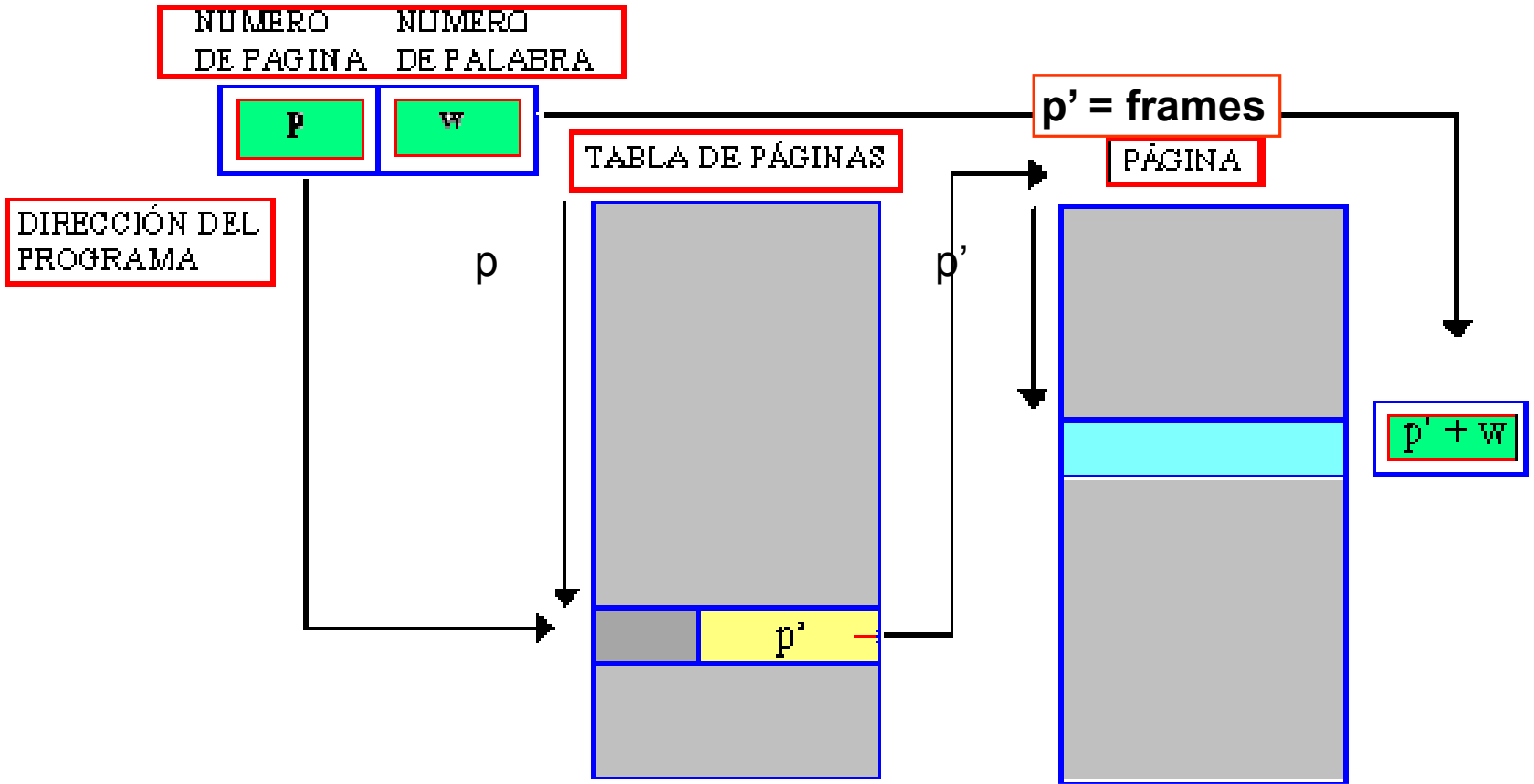
3. Paginación Simple

- La memoria principal se divide en un conjunto de frames de igual tamaño.
- Cada proceso se divide en una serie de páginas del tamaño de los frames.
- Un proceso se carga en los frames que requiera (todas las páginas), no necesariamente contiguos.

- Ventajas. No hay fragmentación externa
- Desventajas. Fragmentación interna pequeña.

3. Paginación Simple

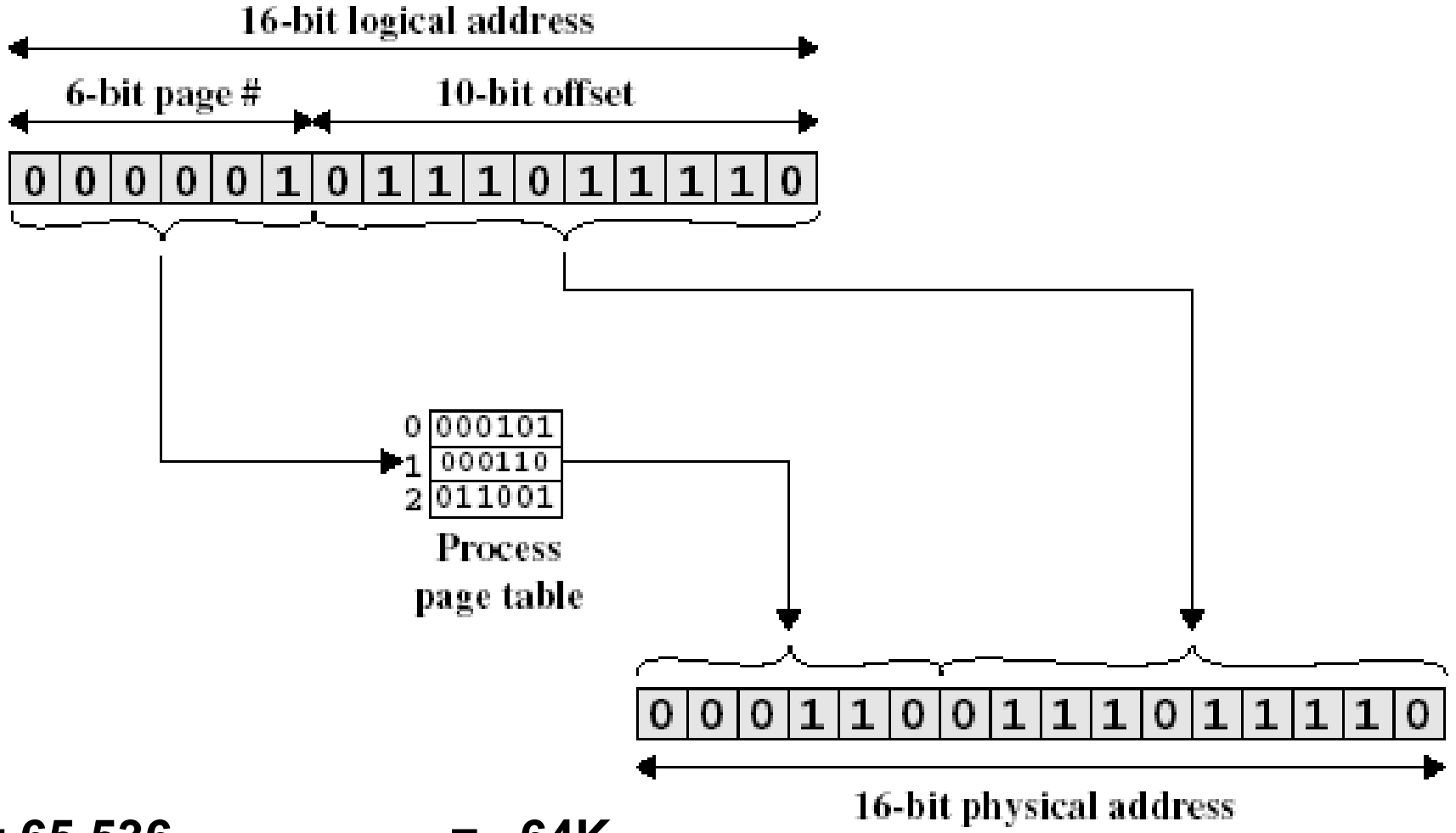
- El SO mantiene una tabla de paginas para cada proceso, que contiene la lista de frames para cada pagina.
- Una dirección de memoria es un número de página (P) y un desplazamiento dentro de la página (W).



3. Estrategias

- Solicitud.
  - Por demanda
- Ubicación.
  - Se cargan todas las páginas de un proceso en los frames libres y se actualiza su tabla de páginas.
- Reemplazo.
  - Una de las páginas se puede sacar y se marca como que no está cargada. Esto es posible por que cada proceso tiene su propia tabla de páginas.
  - No es necesario sacar todas las páginas de un proceso.

3. Capacidad de Direccionamiento



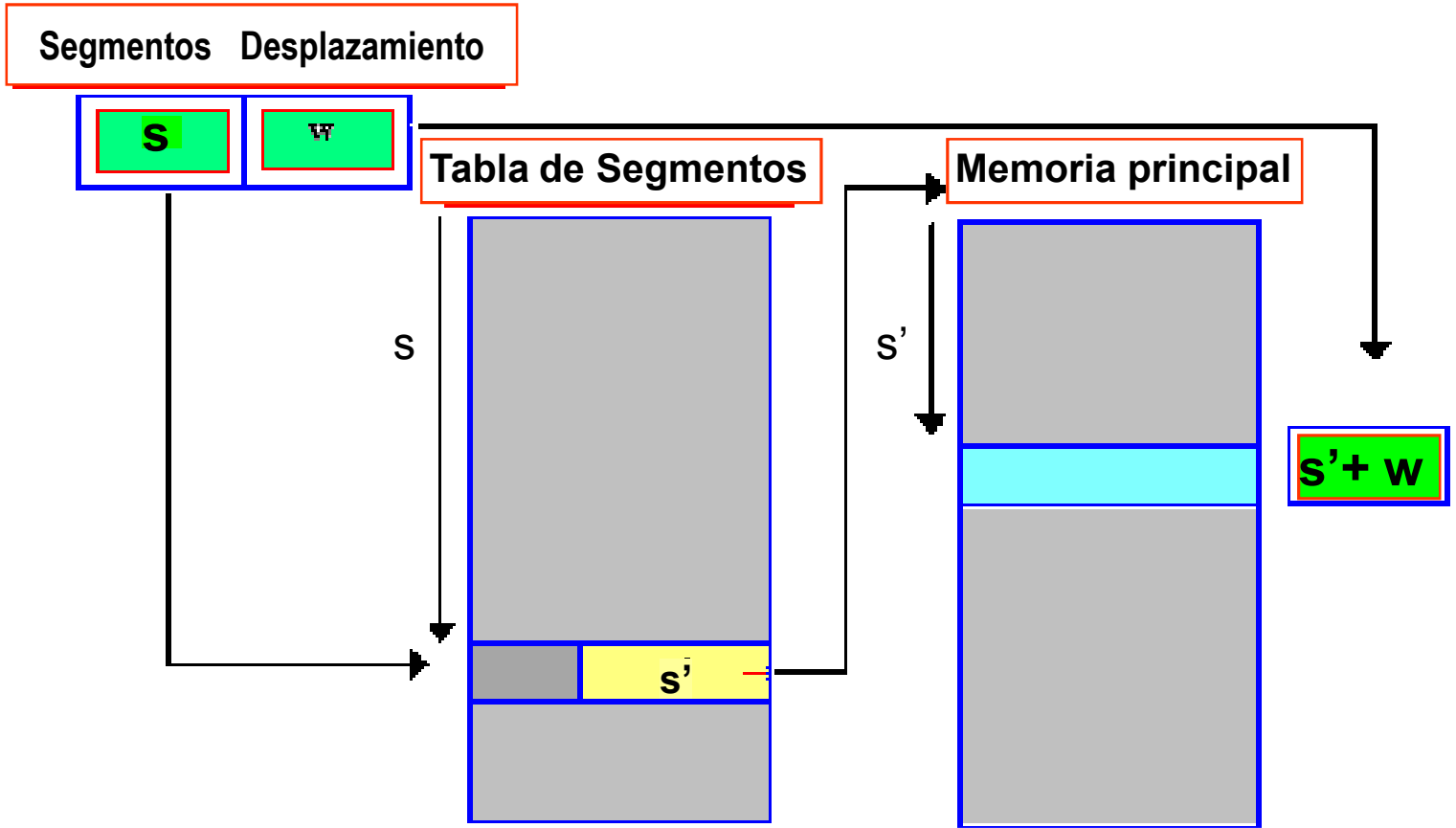
2^{16}= 65,536 = 64K
2^{20}= 1'048,576 = 1MB
2^{24}= 16’777,216 =16MB
2^{32}= 4,294'967,296 = 4GB

4. Segmentación Simple

- Cada proceso y sus datos se dividen en segmentos de longitud variable.
- Un proceso carga sus segmentos en particiones dinámicas no necesariamente contiguas.
- Todos los segmentos de un proceso se deben de cargar en memoria.
- Se diferencia de la partición dinámica en que un proceso puede ocupar más de un segmento.
- Ventajas. No hay fragmentación interna.
- Desventajas. Fragmentación externa, pero menor (compactación).

4. Segmentación Simple

- El SO mantiene una tabla de segmentos para cada proceso y la lista de bloques libres.
- Una dirección de memoria es un número de segmento (S) y un desplazamiento dentro de segmento (W).



4. Estrategias

- Solicitud.
  - Por demanda
- Ubicación.
  - Se cargan los segmentos de un proceso en los bloques libres y se actualiza su tabla de segmentos.
- Reemplazo.
  - Uno de los segmentos se puede sacar y se marca como que no está cargada. Esto es posible por que cada proceso tiene su propia tabla de segmentos.

4. Validación del Direccionamiento

- No hay correspondencia entre dirección lógica y dirección física.
- El SO trabaja con direcciones lógicas.
- El SO debe asegurar que cada dirección lógica esté dentro del rango de direcciones del proceso.
- El SO implementa la tabla de segmentos como un arreglo de registros base-limite.

La segmentación por lo general es invisible al programador. Es el compilador el que define los segmentos.

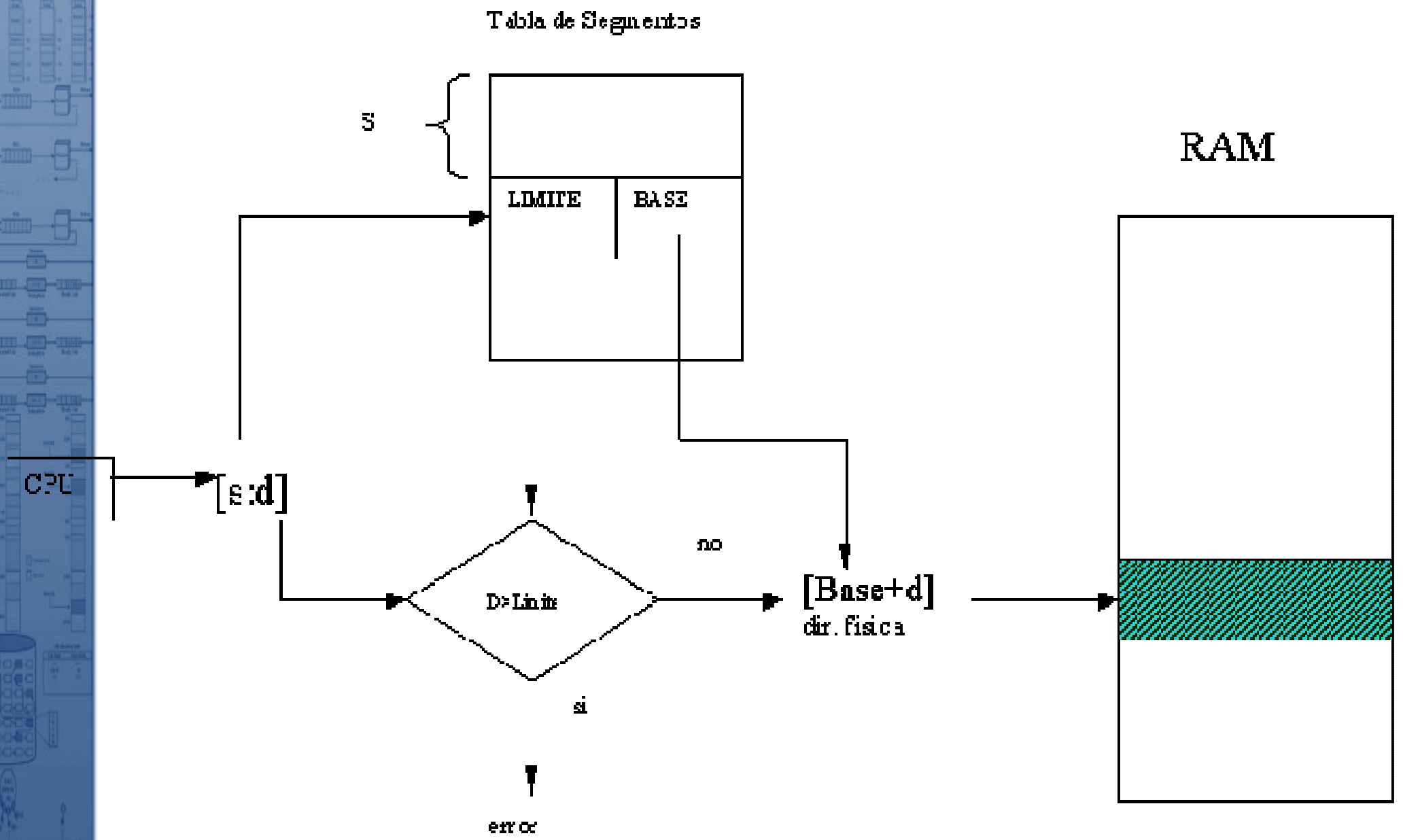


Tabla de Segmentos
S
LIMITE
BASE
RAM
CPU
[s:d]
no
D>Limite
[Base+d]
dir. fisica
si
error

CONCLUSIONES

1. El SAM particionado a diferencia de la paginación o segmentación simple, permite que sólo un proceso se cargue en memoria principal.
2. Cuando se trabaja con bloques de tamaño fijo se genera la fragmentación interna. Si los bloques son de tamaño variable, se genera la fragmentación externa.
3. El SAM de particiones fijas se parece al SAM de paginación simple, diferenciándose en que los primeros requieren que las particiones estén contiguas

Mapa Conceptual de esta parte de la clase

<table>
<tr><td>Real</td><td colspan="2">Real</td><td colspan="2">Real</td><td colspan="2">Virtual</td></tr>
<tr><td>Mono Usuario</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td></tr>
<tr><td></td><td colspan="2">Particionamiento</td><td>Paginación Simple</td><td>Segmentación Simple</td><td>Paginación Virtual</td><td>Segmentación Virtual</td></tr>
<tr><td></td><td>Fija</td><td>Dinámica</td><td colspan="2">Combinación</td><td colspan="2">Combinación</td></tr>
<tr><td></td><td colspan="6">Reubicación, Protección</td></tr>
</table>

TECNICAS DE ADMINISTRACION DE MEMORIA

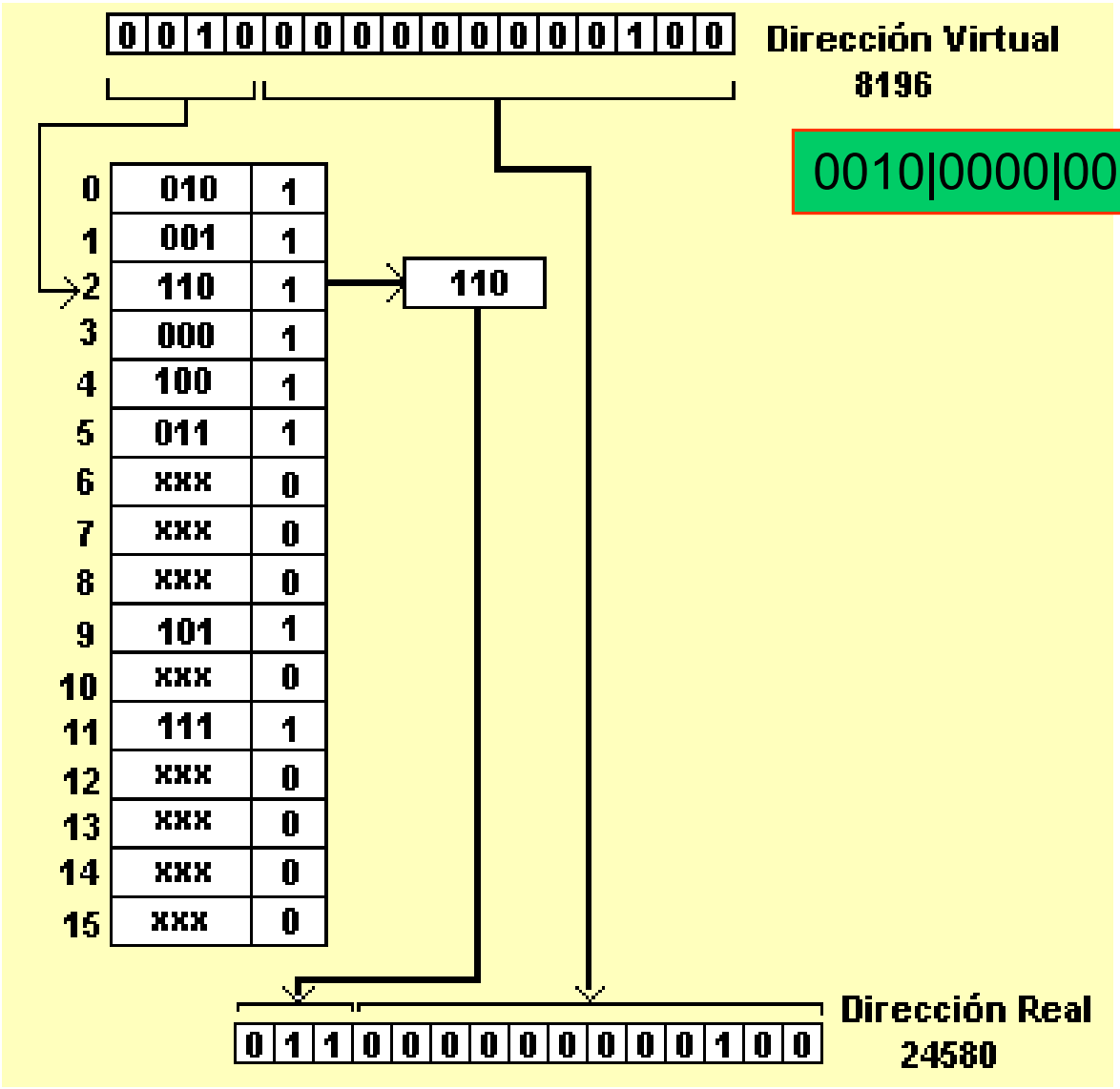
PAGINACION Y SEGMENTACION VIRTUAL

<table>
<tr><td>Real</td><td colspan="2">Real</td><td colspan="2">Real</td><td colspan="2">Virtual</td></tr>
<tr><td>Mono Usuario</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td><td colspan="2">Multiprogramación</td></tr>
<tr><td rowspan="3"></td><td colspan="2">Particionamiento</td><td>Paginación Simple</td><td>Segmentación Simple</td><td>Paginación Virtual</td><td>Segmentación Virtual</td></tr>
<tr><td>Fija</td><td>Dinámica</td><td colspan="2">Combinación</td><td colspan="2">Combinación</td></tr>
<tr><td colspan="6">Reubicación, Protección</td></tr>
</table>

Memoria Virtual

- La memoria virtual es una técnica para proporcionar la **ilusión** de un espacio de memoria mayor que la memoria física, sin tener en cuenta el tamaño de la memoria física.
- Está soportada por el mecanismo de **traducción de memoria**, junto con un almacenamiento rápido en disco duro (**swap**).
- El **espacio de direcciones virtual**, está mapeado de tal forma que una pequeña parte de él, está en memoria real y el resto almacenado en el disco.

Traducción de Memoria



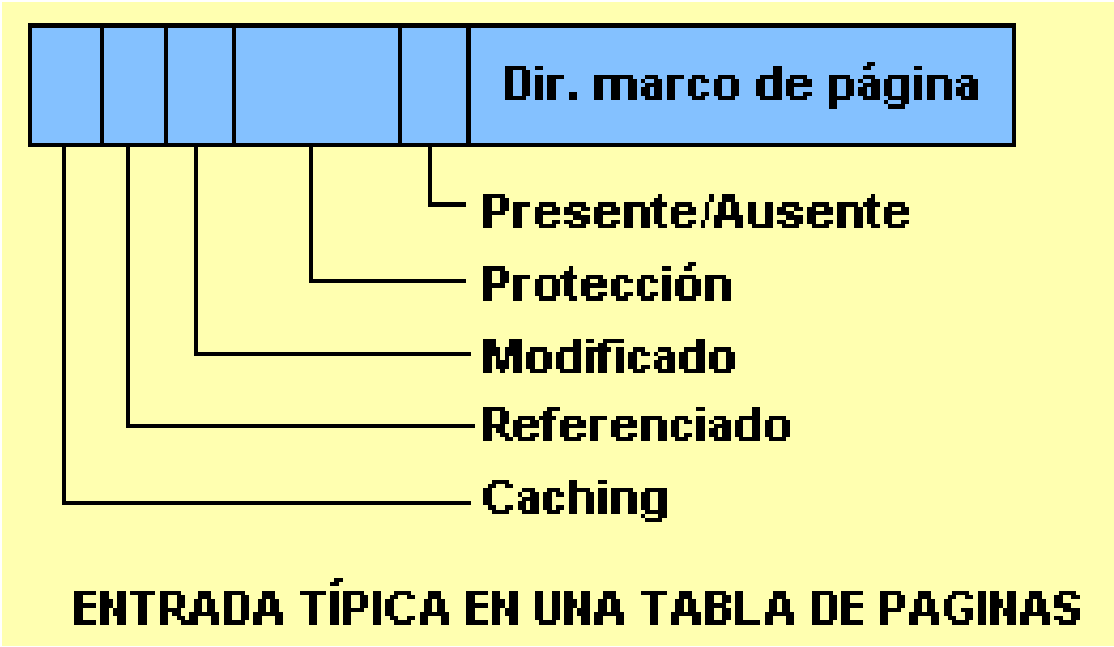
5. Memoria Virtual Paginada

- Igual que la paginación simple.
- No es necesario cargar todas las páginas.
- Las páginas no residentes se cargan por demanda.

- Ventajas. No fragmentación externa. Alto grado de multiprogramación. Gran espacio virtual para el proceso.
- Desventaja. Sobrecarga por gestión compleja de memoria.

5. Memoria Virtual Paginada

- Cada proceso tiene su propia tabla de paginas.

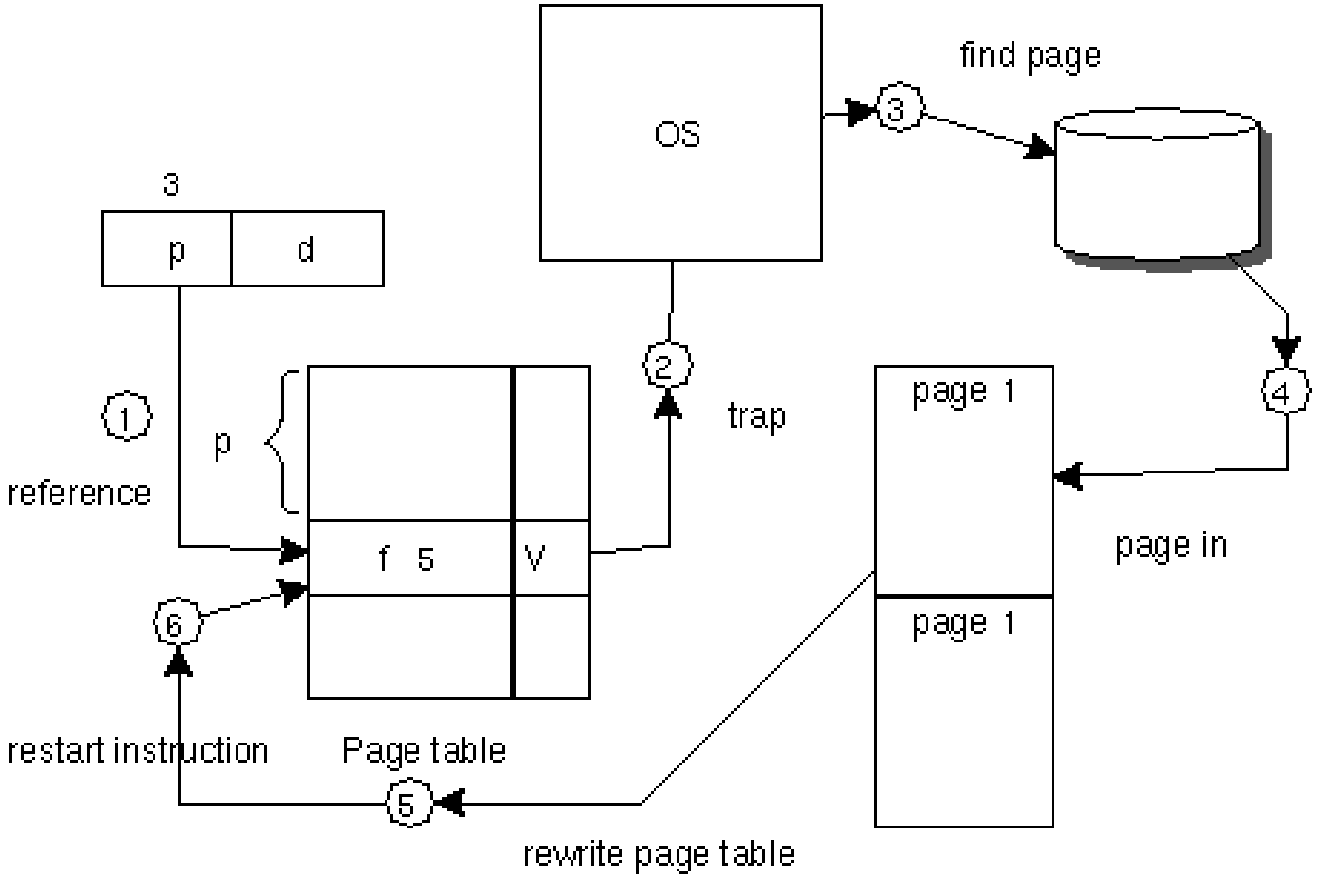


ENTRADA TÍPICA EN UNA TABLA DE PAGINAS

- Si la pagina no se modifica, al realizarse el swap a disco no se necesitara copiar desde la memoria principal a la memoria secundaria.

5. Fallo de Página (page fault)

- Ocurre cuando se referencia a una dirección virtual y ella no reside en la memoria real, se presenta una interrupción ***fallo de página.***



5. Tamaño de Página

- Páginas pequeñas
  - Menos fragmentación interna.
  - Más páginas para el proceso.
  - Muchas páginas por proceso.
  - La tabla de paginas crecerá en tamaño.
  - Se necesita mas MV para carga la tabla.
  - El fallo de página se reduce.
- Páginas grandes
  - Mas fragmentación interna.
  - C/página contiene mas porciones del proceso.
  - Se ocupa memoria innecesariamente.
  - El fallo de página se incrementa.

6. Memoria Virtual Segmentada

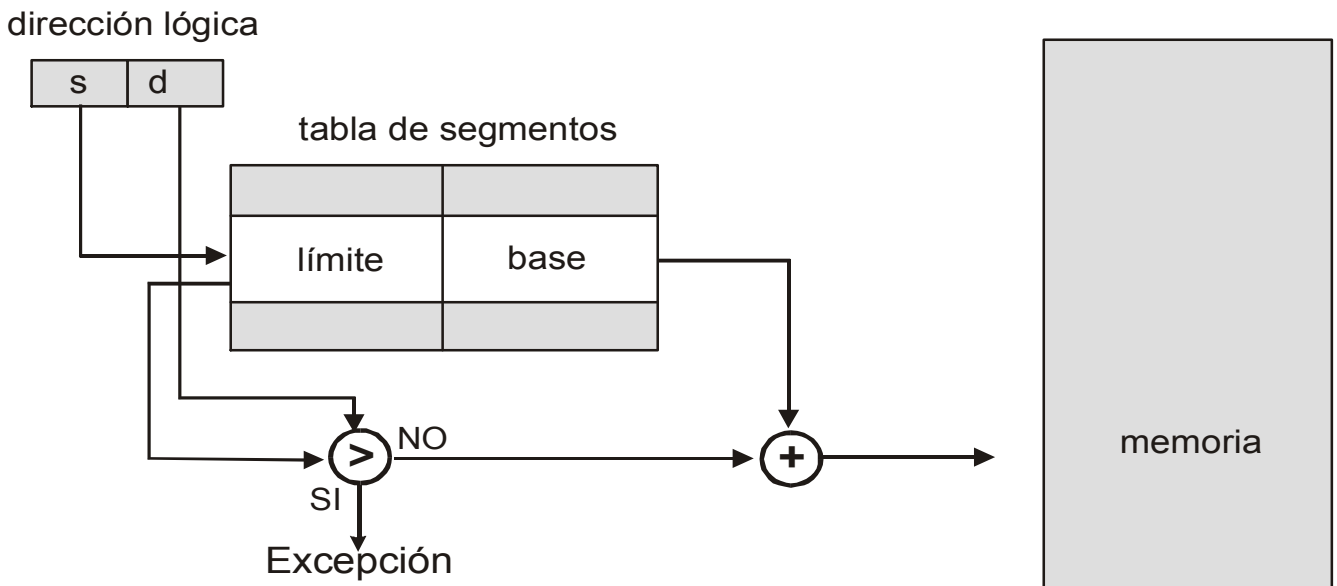
- Igual que la segmentación simple.
- No es necesario cargar todos los segmentos.
- Las segmentos se cargan por demanda.
- Segmentos de tamaño dinámico, según la demanda.
- Se puede alterar los programas y recompilarlos independientemente.

6. Memoria Virtual Segmentada

- Permite compartir datos entre procesos, mediante el uso segmentos compartibles.
- Permite la protección de datos, el administrador otorgar permisos a este segmento.
- Ventajas. No hay fragmentación interna. Alto grado de multiprogramación. Gran espacio virtual para el proceso. Soporte de protección y compartición (sharing).
- Desventajas. Sobrecarga por gestión compleja de memoria.

6. Tabla de Segmentos

- El SO debe mantener una lista de huecos libres.
- Un bit expresa si el segmento se encuentra ya en memoria.
- Un bit expresa si el segmento ha sido modificado.



CONCLUSIONES

1. El SAM particionado a diferencia de la paginación o segmentación simple, permite que sólo un proceso se cargue en memoria principal.
2. Cuando se trabaja con bloques de tamaño fijo se genera la fragmentación interna. Si los bloques son de tamaño variable, se genera la fragmentación externa.
3. El SAM de particiones fijas se parece al SAM de paginación simple, diferenciándose en que los primeros requieren que las particiones estén contiguas

Consistencia, Replicación y Memoria Compartida

6

Sistemas Distribuidos (INTRO!)

Prof. Javier Echaiz
D.C.I.C. – U.N.S.
http://cs.uns.edu.ar/~jechaiz
je@cs.uns.edu.ar

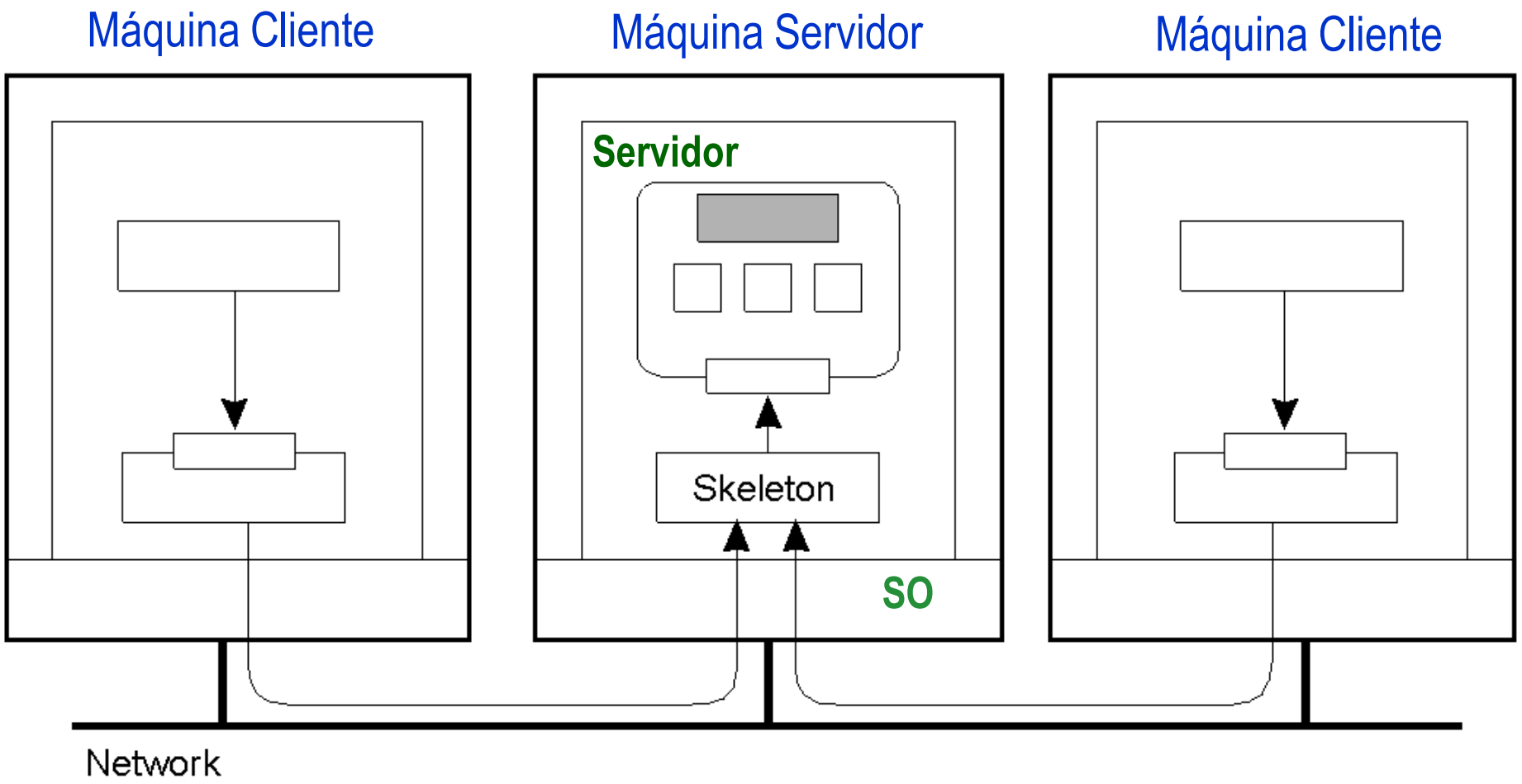
Razones para la Replicación

Hay dos razones principales para la replicación de datos:

- Confiabilidad
  - Continuidad de trabajo ante caída de una réplica.
  - Mayor cantidad de copias mejor protección contra la corrupción de datos (copias para “desempatar”).
- Rendimiento (Performance)
  - El SD escala en número (e.j. réplica de web servers).
  - Escala en área geográfica (disminuye el tiempo de acceso al dato, acerca dato con procesamiento).
  - Consulta simultánea de los mismos datos.

Precio a pagar por la replicación de datos:
Problemas de Consistencia

Replicación de Objetos



- Organización de un objeto remoto distribuido compartido por dos clientes diferentes.

Replicación de Objetos (cont)

Cuando se replican objetos remotos en varias máquinas es necesario resolver el problema de como proteger al objeto contra el acceso simultáneo de múltiples clientes.

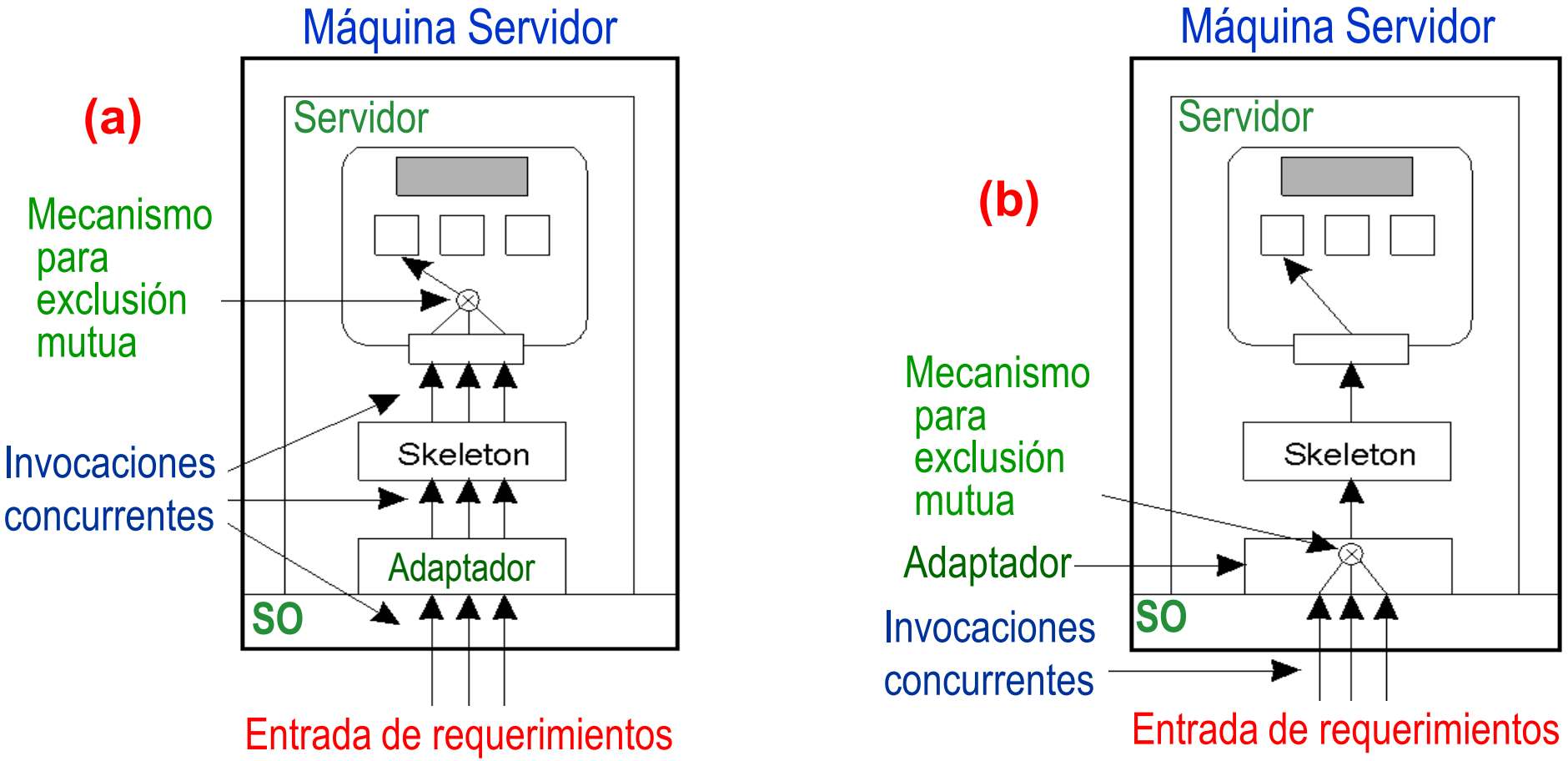
Básicamente hay dos soluciones [Briot et al,1998].

Una solución es que el mismo objeto maneje las invocaciones concurrentes (a). Ej. Declaración de métodos Java como *synchronized*.

Otra solución es que el objeto esté totalmente desprotegido y del control de concurrencia se ocupe el servidor en el cual el objeto reside (b).

Dado que el objeto está replicado debe extremarse el cuidado de modo que cada réplica esté permanentemente actualizada.

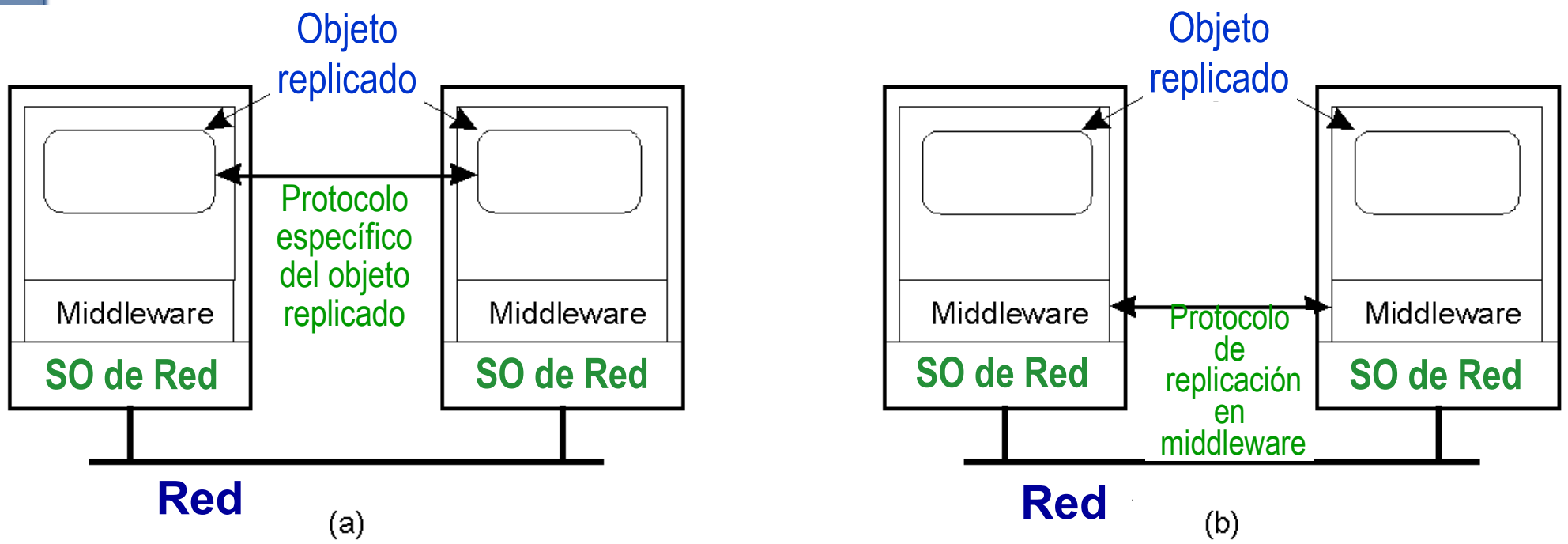
Replicación de Objetos (cont)



a) Un objeto remoto capaz de manejar invocaciones concurrentes por sus propios medios.

b) Un objeto remoto para el cual se requiere un adaptador para manejar las invocaciones concurrentes.

Replicación de Objetos (cont)



a) Un sistema distribuido para manejo propio de la replicación por los objetos distribuidos.

b) Un sistema distribuido responsable del manejo de las réplicas.

Replicación como Técnica de Escalabilidad

En general **lograr escalabilidad va en detrimento del rendimiento**.

Como técnicas para facilitar la escalabilidad se utiliza la **replicación** y el **caching**.

Ubicar copias de datos u objetos cercanos a los procesos que los usan mejora el rendimiento por la reducción del tiempo de acceso y resuelve el problema de escalabilidad.

Problemas:

La actualización de las réplicas consume más ancho de banda de la red.

Mantener múltiples copias consistentes resulta a su vez un serio problema de escalabilidad y más en un contexto de consistencia estricta (*strict consistency*).

La idea es que la actualización se realice con una única operación atómica. Se necesita sincronizar todas las réplicas.

Dilema: Por un lado la replicación tiende a resolver el problema de la escalabilidad (aumenta el rendimiento); por otro mantener consistentes las copias requiere sincronización global. La cura puede ser peor que la enfermedad.

Modelos de Consistencia

Un modelo de consistencia es esencialmente un contrato entre procesos y el almacenamiento de datos. Es decir: si los procesos *acuerdan* obedecer ciertas reglas, el almacenamiento *promete* trabajar correctamente.

Normalmente un proceso que realiza una operación de lectura espera que esa operación devuelva un valor que refleje el resultado de la última operación de escritura sobre el dato.

Los modelos de consistencia se presentan divididos en dos conjuntos:

- **Modelos de consistencia centrados en los datos.**
- **Modelos de consistencia centrados en el cliente.**

Coming Next