Bloque II

UT. 5 . Sistemas operativos gestión de recursos

1. Planificación de Memoria

Profesor: Baldomero Sánchez Pérez.

1.1. ADMISTRACIÓN DE LA MEMORIA

- *Organización y gestión de la memoria. Conceptos generales*
- *Gestión de la memoria en los sistemas monoprogramados*
- *Gestión de la memoria en los sistemas multiprogramados*
- *Asignación de memoria contigua*
  - *Particiones estáticas*
  - *Particiones dinámicas*
  - *Estrategias de colocación*
  - *Intercambio*
- *Asignación de memoria no contigua*
  - *Esquema general de traducción*
  - *Paginación*
  - *Segmentación*
  - *Segmentación Paginada*

1.1. ADMISTRACIÓN DE LA MEMORIA

- *Asignación de memoria no contigua*
  - *Esquema general de traducción*
  - *Paginación*
    - *Memoria asociativa*
    - *Páginas compartidas*
    - *Protección*
    - *Dos visiones de la memoria*
  - *Segmentación*
    - *Visión del usuario de la memoria*
    - *Hardware*
    - *Implementación de las Tablas de Segmentos*
    - *Compartición y Protección*
    - *Fragmentación*
  - *Segmentación Paginada*

1.2. Jerarquía de la memoria

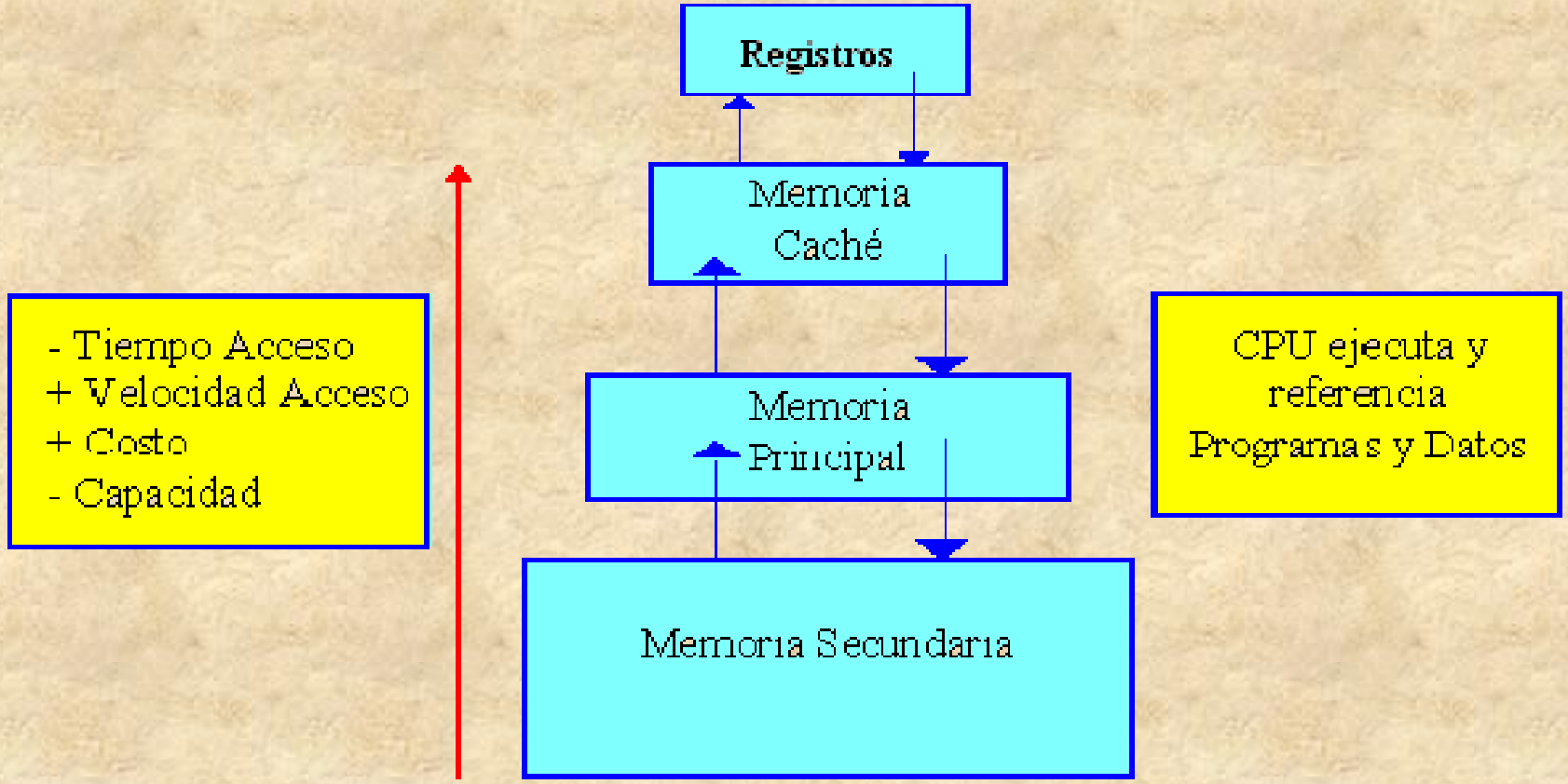


Figura 6.1. Jerarquía de Memoria

1.3. Gestión de la memoria en los sistemas monoprogramados

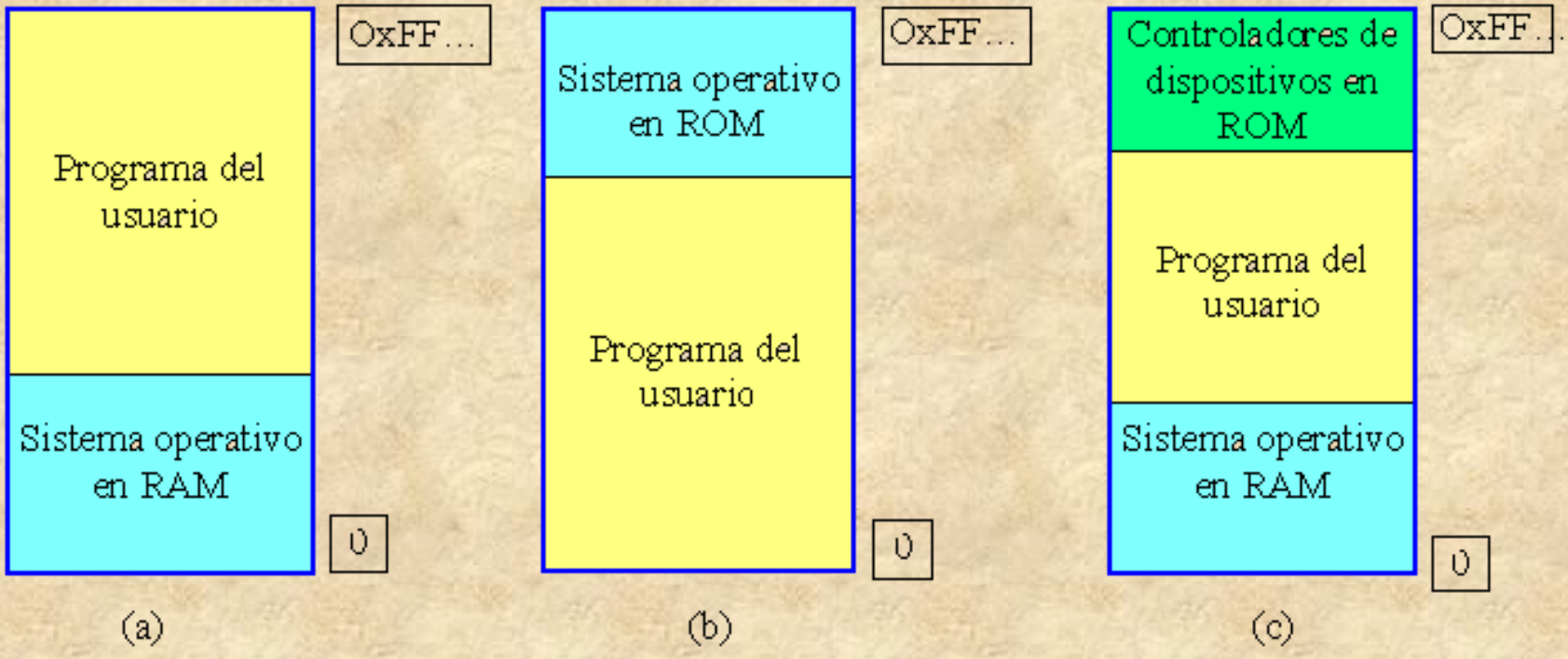


Figura 6.2. Tres formas de organización de la memoria, con un sistema operativo y un proceso de usuario.

1.4. Gestión de la memoria en los sistemas multiprogramados

- En un sistema de multiprogramación la memoria debe ser compartida por varios procesos para mejorar los recursos del ordenador.
- Primero: hay que llevar un recuento de las zonas de memoria ocupadas por los procesos.
  - Cuando un nuevo proceso entre en la memoria se le asignará una zona que estaba libre.
- Al escribir un programa no se sabe en qué zona de memoria se ubicará, siendo posible que durante la vida de un proceso éste cambie varias veces de emplazamiento.
- Se debe proteger las zonas de memoria ocupadas por los procesos, máxime en sistemas **multiusuario** donde los procesos pueden pertenecer a distintos usuarios.

1.5. Asignación de memoria contigua

- **La asignación de memoria contigua un proceso se ubica en su totalidad en posiciones consecutivas de memoria.**
- **Ej. los sistemas de monoprogramación**
- **Dos métodos de asignación contigua empleados históricamente en sistemas multiprogramados.**
  - **Particiones estáticas.**
  - **Particiones dinámicas.**

1.5.1. Particiones estáticas

- Esta forma de gestión consiste en dividir la memoria en varias zonas, pudiendo ser cada zona de un tamaño diferente.
- El tamaño de las zonas podrá ser modificado eventualmente por algún usuario responsable de la administración del ordenador.
- Los trabajos se traducían mediante compiladores y **ensambladores** absolutos, para ejecutarse en una **partición específica y contiguos**
  - **Una vez introducido un proceso en una partición, permanece en ella hasta su finalización.**
  - **Si un trabajo se iniciaba, y la partición para la que estaba compilado estaba ocupada, tenía que esperar, aunque estuvieran libres otras particiones. Esto provoca una pérdida de eficiencia.**
- **Fragmentación interna:** Se produce , al carga un proceso, en una partición fija, cuyo tamaño es superior al proceso cargado, con lo que se aprovecharía solo parte del tamaño de la partición asignada a ese proceso, el resto no se utiliza.(ej: proceso 5k, particion 10k)

1.5.1. Particiones estáticas

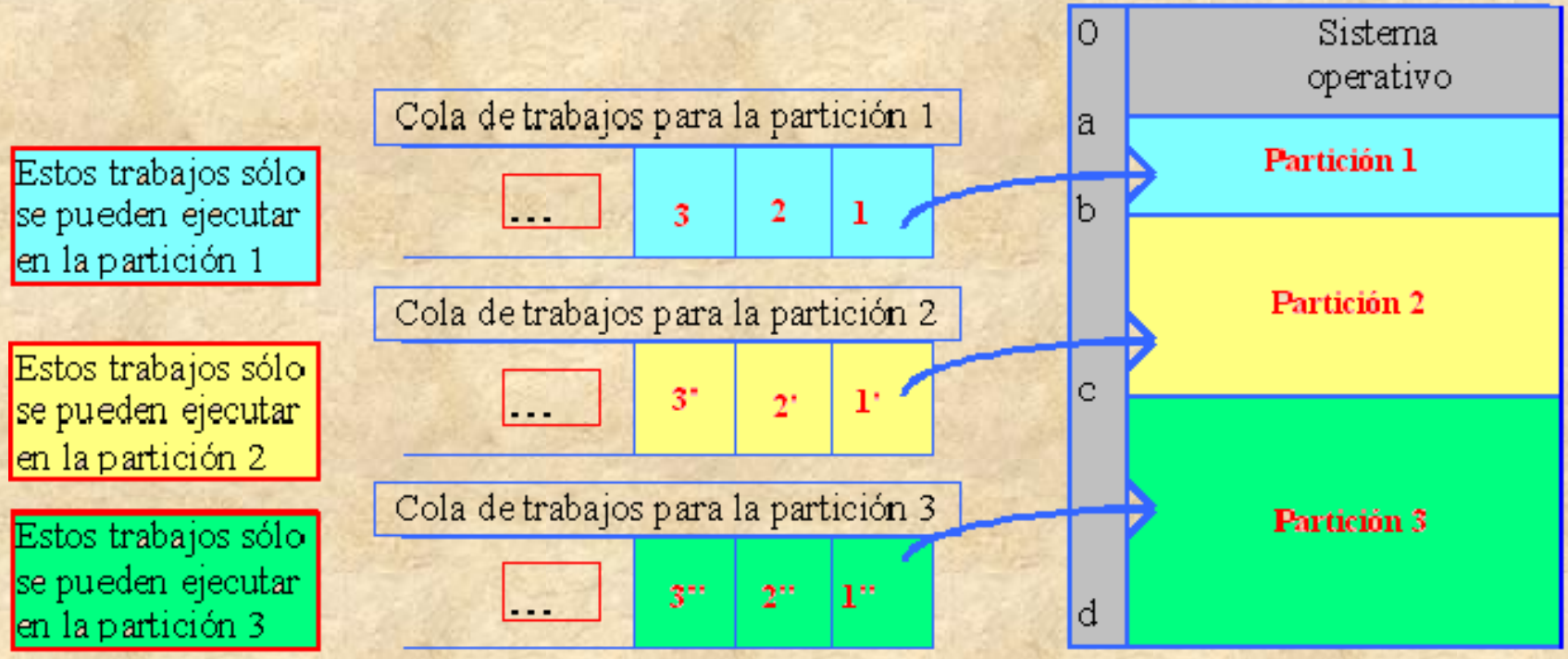


Figura 6.3. Multiprogramación con particiones estáticas, con traducción y carga absolutas

1.5.2. Protección

- Si se tiene el esquema *hardware* del registro base, para lograr la protección de las zonas de memoria basta con añadir un nuevo registro, denominado **registro límite**.
- Registro límite : este registro guarda la última dirección de la partición, y forma también parte del PCB del proceso.
  - El *hardware*, después de sumar el registro base a la dirección relativa, comprueba que la dirección obtenida no supere el valor del registro límite.
  - Si se supera el valor, se está intentando acceder a una zona que no corresponde al proceso; en esta situación, el *hardware* genera una **interrupción**.
  - El sistema operativo sirve a la interrupción, lo normal es que mande una señal al proceso por violación de memoria.
  - Si el proceso no tiene definido atrapar esa señal, lo cual es lo más probable, se eliminará al proceso.

1.5.3. Particiones dinámicas

- Se va asignando la memoria dinámicamente a los procesos, conforme se introducen en la memoria.
- A cada proceso se le asigna exactamente la memoria que necesita.
- Con este método de gestión de la memoria se evita el problema de la fragmentación interna.
- La **fragmentación externa** entre particiones, consiste en que se creen huecos libres demasiado pequeños como para que quepan procesos, aunque la unión de todos esos huecos produciría un hueco considerable, lo que acarrea el desperdicio de la memoria.
- Una posible solución es la **compactación** de la memoria, que consiste en desplazar todos los procesos hacia la parte inferior de la memoria mientras sea posible.
  - Como la compactación lleva mucho tiempo, a veces no se realiza, Solo se realiza en grandes equipos, con proceso en ejecución continuo: por la noche, en horas de poco uso del ordenador.
    - El sistema debe detener todas sus actividades mientras realiza la compactación.
    - Puede ocasionar tiempos de respuesta irregulares para usuarios **interactivos**, y podría ser devastador en un sistema de **tiempo real**.

1.5.3. Particiones dinámicas

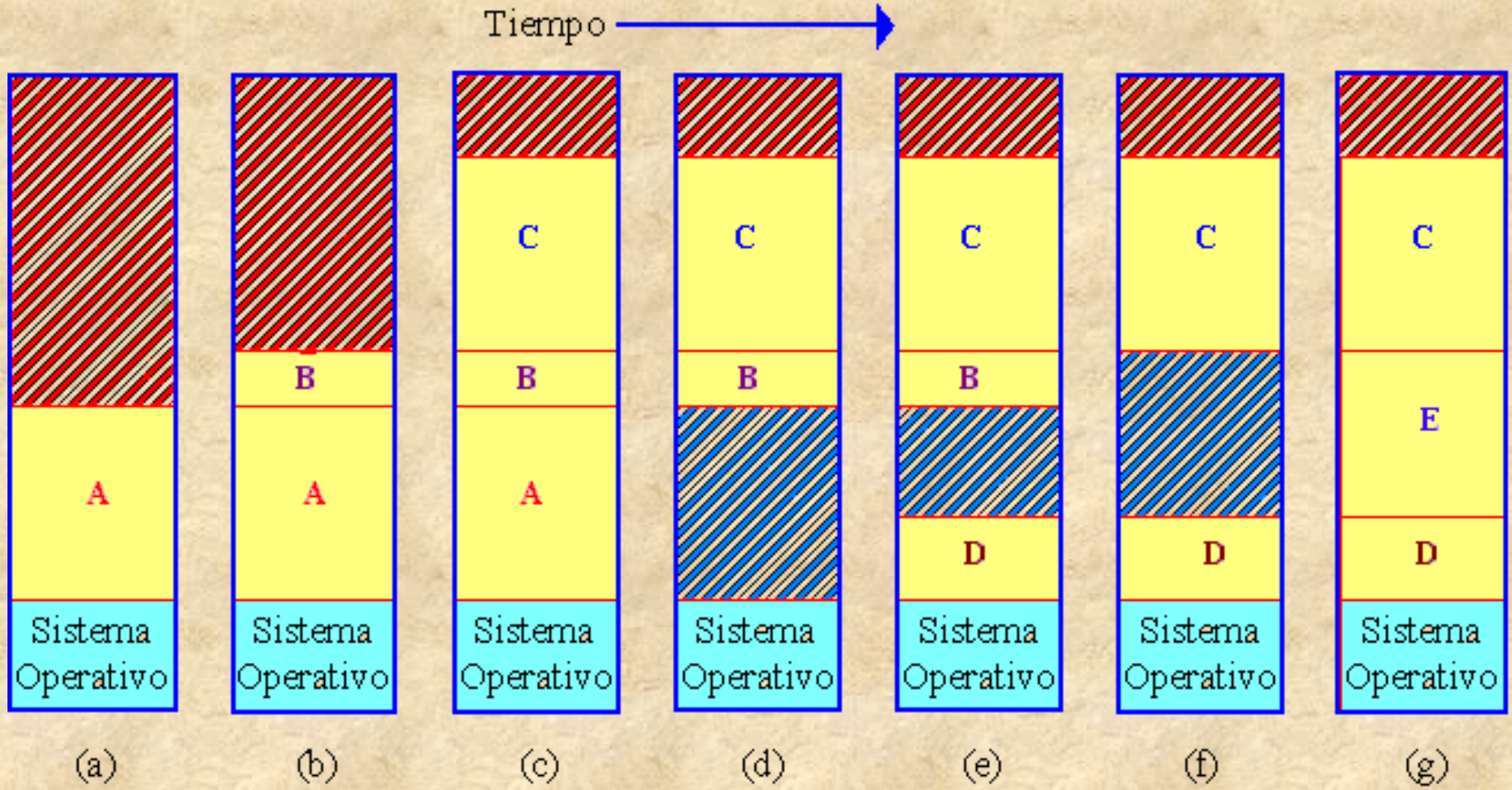


Figura 6.4. La asignación de memoria cambia cuando el proceso llega o sale de la memoria. Las regiones sombreadas son memoria libre.

1.5.3. Particiones dinámicas

- El esquema de los registro base y límite sigue siendo válido para la reasignación y la protección.
- La cantidad de memoria por asignar a un proceso recién creado.
- Si los procesos se crean con un tamaño fijo invariante, la asignación es muy sencilla, se asigna exactamente lo que se necesite.

1.5.3. Particiones dinámicas (2)

- Si los segmentos de datos de los procesos pueden crecer, como es el caso de la asignación dinámica de memoria a partir de una pila.
- Si hay un hueco adyacente al proceso, éste puede ser asignado, y el proceso podrá crecer hacia el hueco.
- Si el proceso es adyacente a otro proceso, el proceso de crecimiento deberá ser desplazado a un hueco de la memoria lo suficientemente grande; o bien, habrá que eliminarlo.

1.5.3. Particiones dinámicas

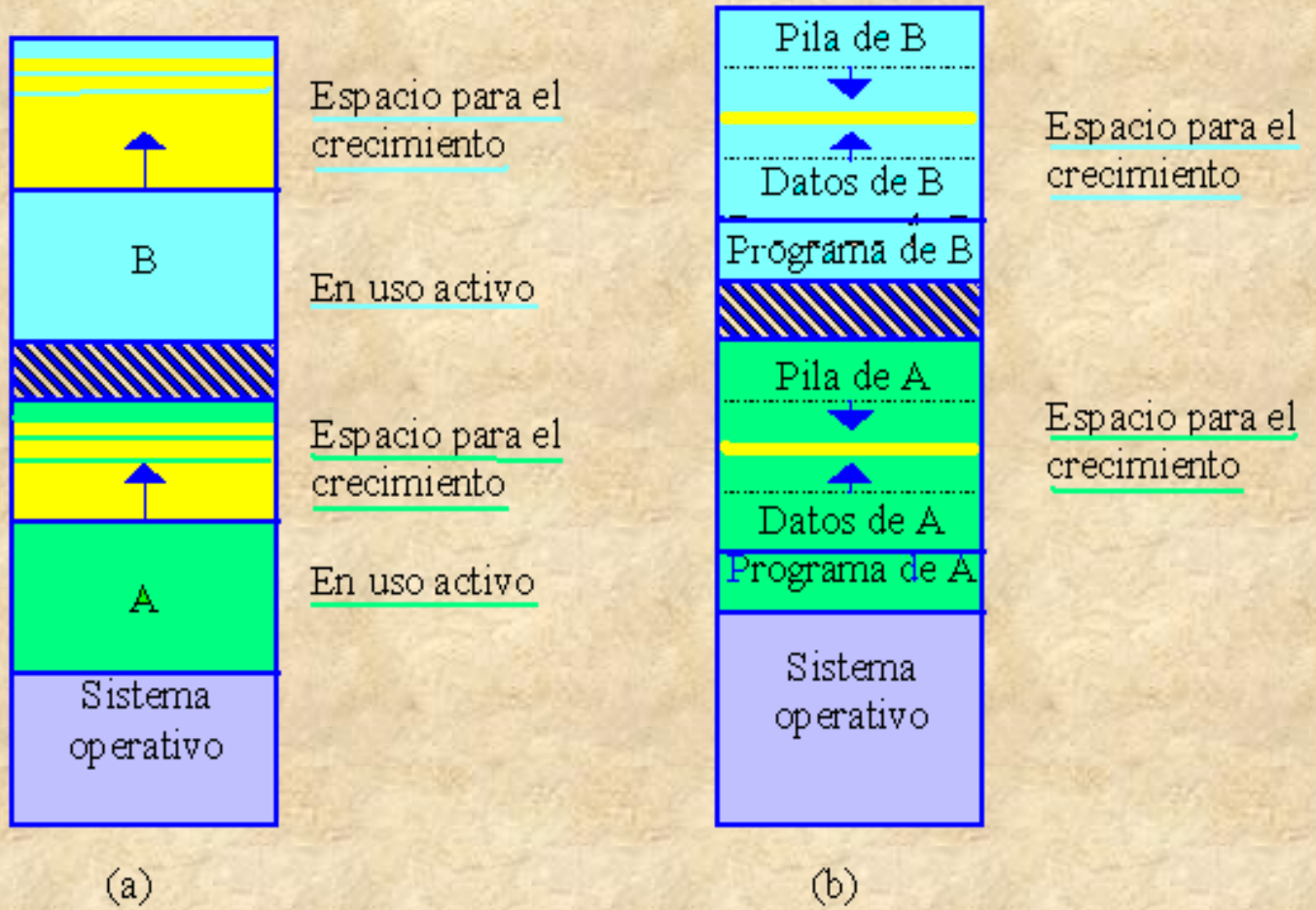
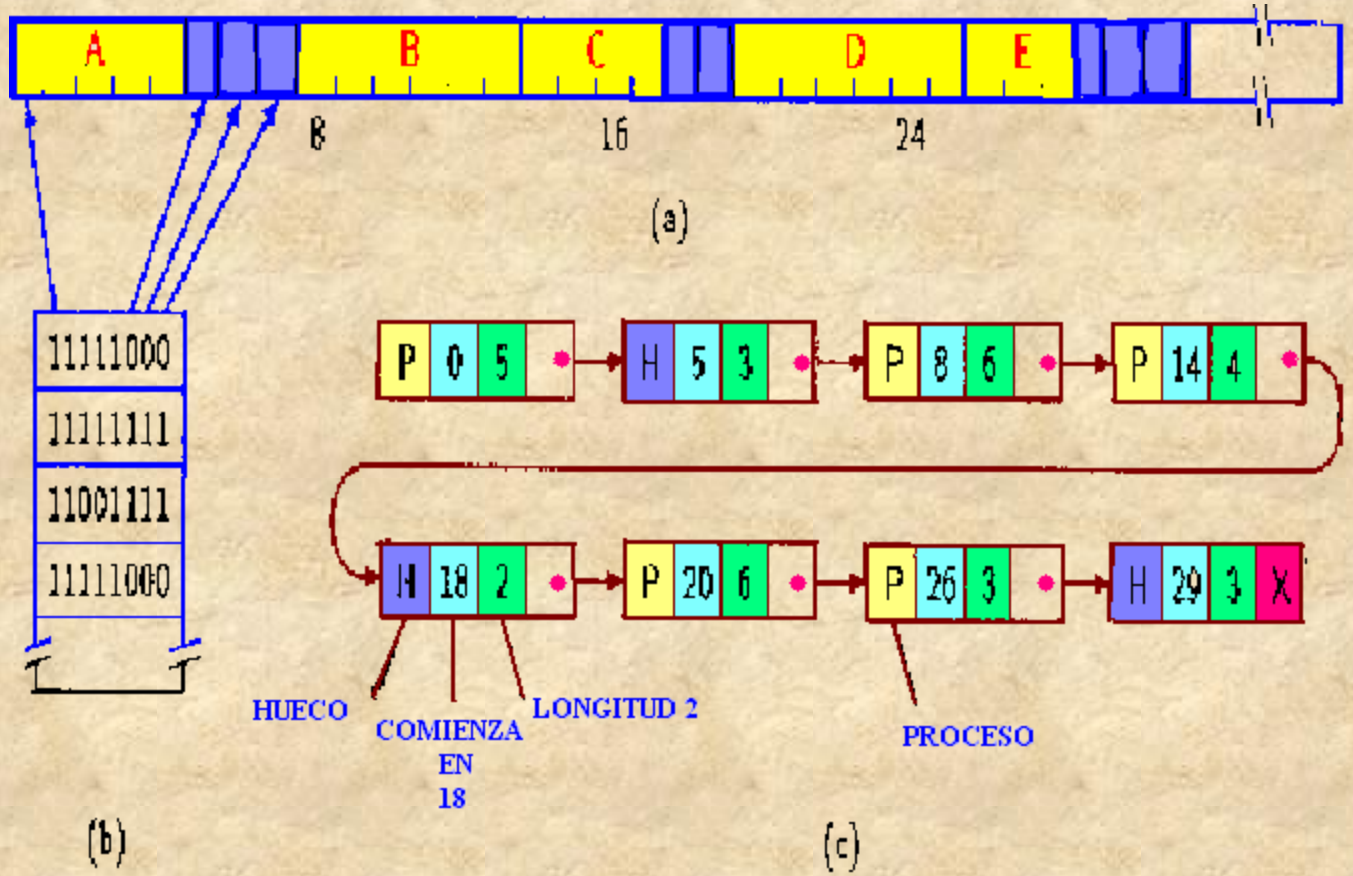


Figura 6.5. (a) Asignación de espacio a un segmento de datos que crece.
(b) Asignación de espacio a una pila y un segmento de datos que crece.

1.5.3. Particiones dinámicas

- Lista doblemente enlazada



1.6. Estrategias de colocación

- Cuando en un sistema de particiones dinámicas se debe asignar memoria principal para un nuevo proceso, y los procesos y huecos se mantienen en una lista ordenada por direcciones, se pueden utilizar diversos algoritmos para la elección del hueco de memoria donde ubicar al proceso.

1.6. Estrategias de colocación

- El algoritmo
  - El primero en ajustarse (*first fit*). Se revisa la lista de huecos hasta encontrar un espacio lo suficientemente grande. El espacio se divide entonces en dos partes, una para el proceso, y otra para la memoria no utilizada, excepto en el caso poco probable de un ajuste perfecto. Este algoritmo es rápido, ya que busca lo menos posible.
  - Es el mejor en ajustarse (*best fit*), el cual busca en toda la lista, y elige el mínimo hueco suficientemente grande como para ubicar al proceso. Este algoritmo intenta que los huecos que se creen en la memoria sean lo más pequeños posible.
  - El peor ajuste (worst fit), busca en la lista, y elige el máximo hueco.

1.6. Estrategias de colocación

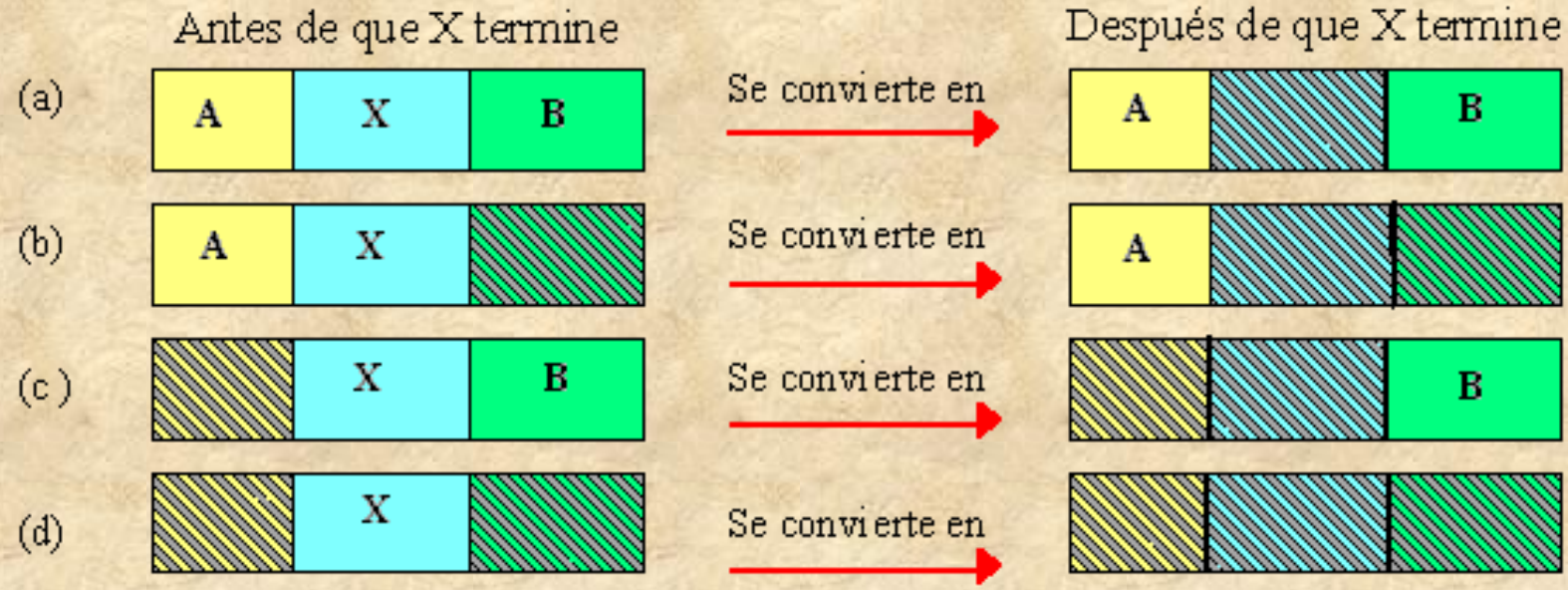


Figura 6.7. Cuatro combinaciones de vecinos para el proceso X que concluye

1.6.1. Ejemplos de carga en memoria

Lista de tareas:

Número de tarea	Memoria solicitada
J1	10K
J2	20K
J3	30K*
J4	10K

Lista de memoria:

Localidad de memoria	Tamaño del bloque de memoria	Número de tarea	Tamaño de tarea	Estado	Fragmentación interna
10240	30K	J1	10K	Ocupado	20K
40960	15K	J4	10K	Ocupado	5K
56320	50K	J2	20K	Ocupado	30K
107520	20K			Libre	
Total disponible	115K	Total utilizado	40K		