UNIVERSIDAD DE
Belgrano
BUENOS AIRES - ARGENTINA

Lic. en Sistemas de Información

SISTEMAS OPERATIVOS

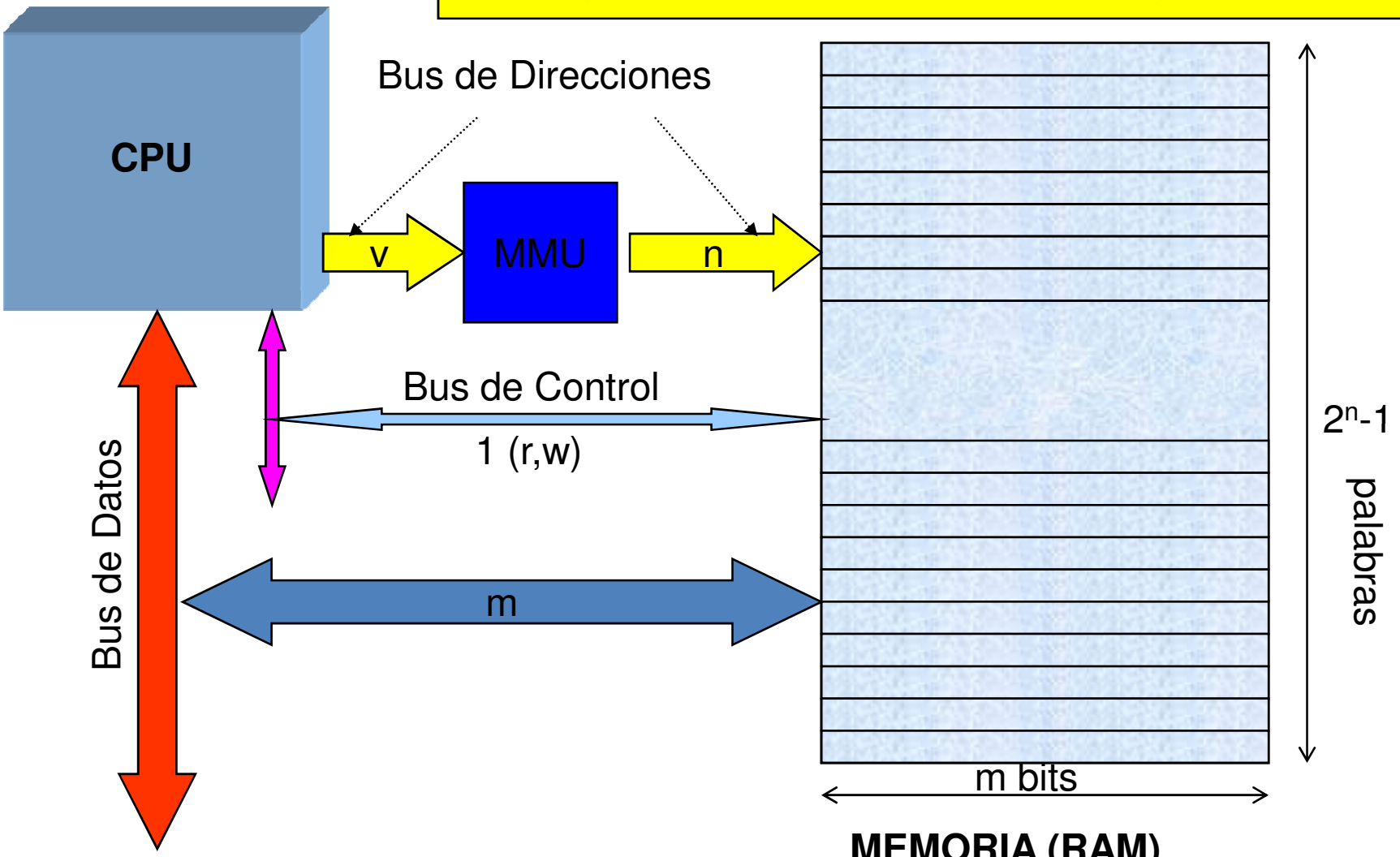
Ciclo 2014 – Plan 2012

UNIDAD 6 Parte A

ADMINISTRACION DE MEMORIA

ING. AGUILERA SERGIO OMAR

ESTRUCTURA DE MEMORIA CENTRAL



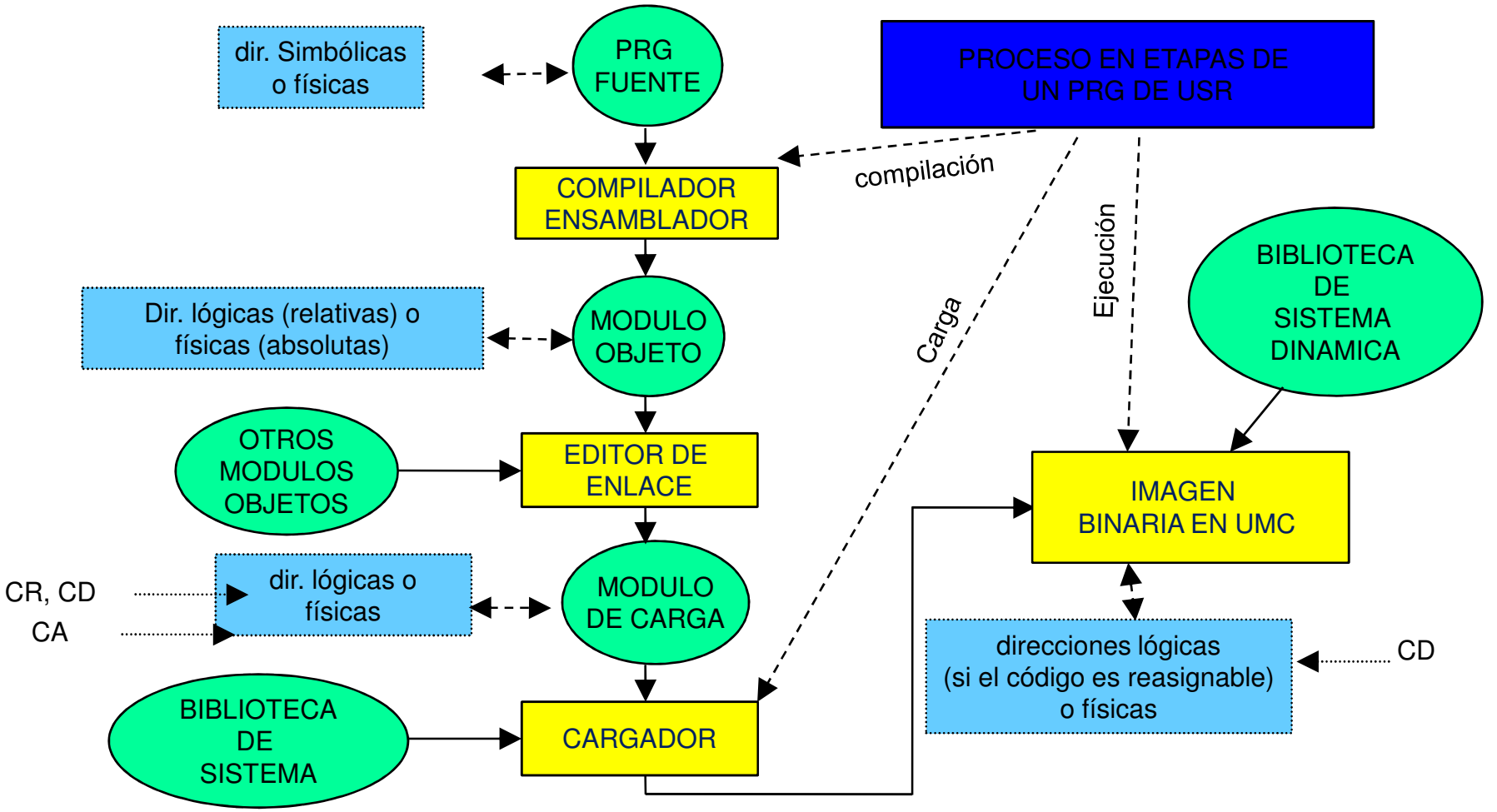
Tamaño en (bytes) = $(2^n-1)*(m/8)$

UNIDAD DE MEMORIA CENTRAL

- La memoria es una matriz de palabras o bytes direccionables (accesibles mediante una dirección única) por la CPU

- - Las direcciones de un proceso pueden ser representadas de modo diferente en las sucesivas etapas del ciclo de un programa de usuario (compilación, carga y ejecución)

- - Las direcciones que entiende el controlador de la memoria son direcciones absolutas. La conversión entre las distintas representaciones de las direcciones y las direcciones absolutas se denomina vinculación.



CA: Carga Absoluta

CR: Carga Reubicable

CD: Carga Dinámica

ETAPAS DE DIRECCIONAMIENTO

OBJETIVOS DEL ADM. DE MEMORIA

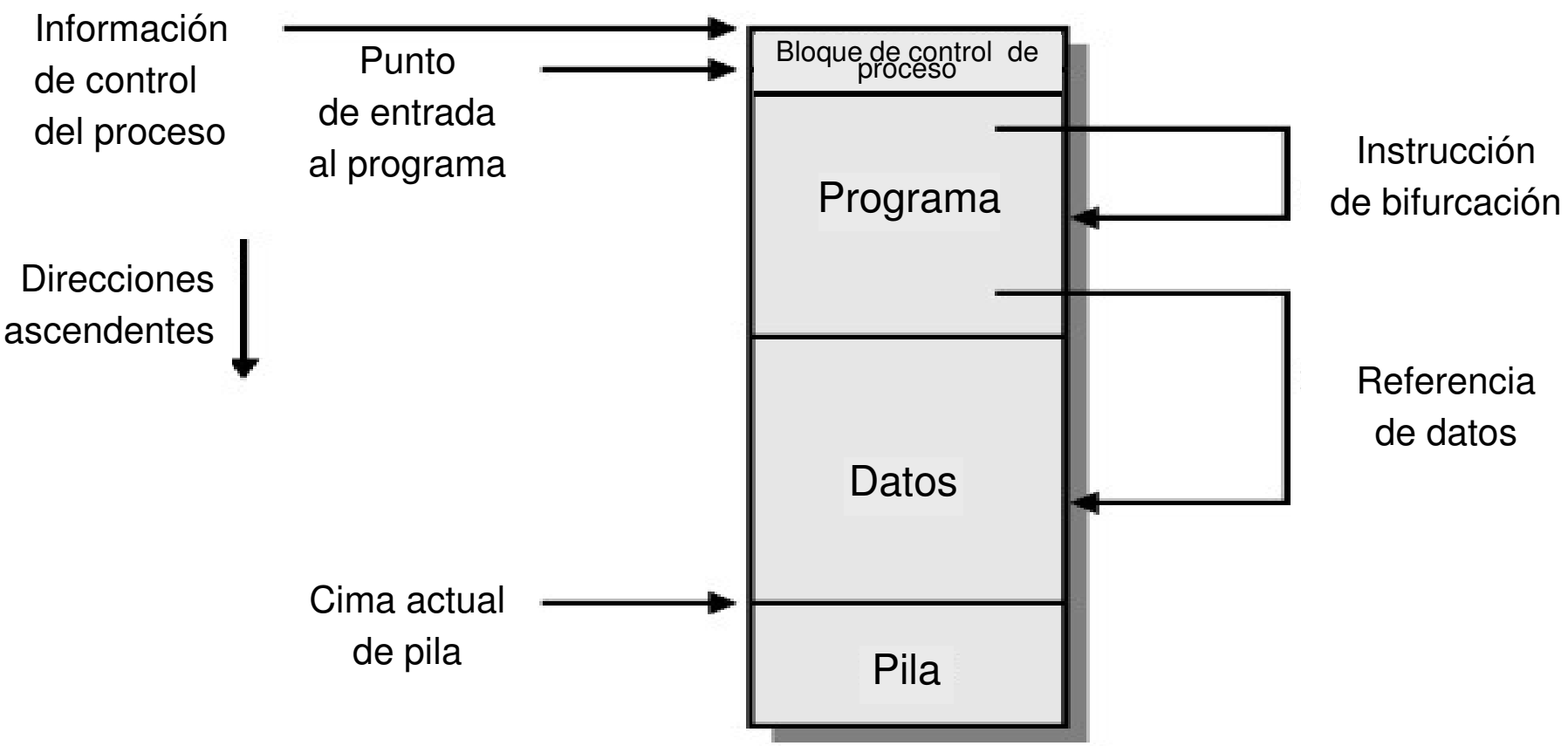
- Es el Administrador más grande de un SOp Monokernel.
- Se busca optimizar el espacio y la velocidad de acceso.
- Subdivisión de la memoria para hacer sitio a varios procesos.
- Hace falta repartir eficientemente la memoria para introducir tantos procesos como sea posible.

ADM. UMC


1. REUBICACION
2. PROTECCION
REQUISITOS
3. COMPARTICION
4. ORGANIZACION LOGICA
5. ORGANIZACION FISICA

1. REUBICACION

- El programador no conoce qué otros programas residirán en la memoria en el momento de la ejecución.
- Mientras que se está ejecutando el programa, puede que se descargue en el disco y que vuelva a la memoria principal, pero en una ubicación distinta a la anterior (reubicación).
- Se deben traducir las referencias a la memoria encontradas en el código del programa a las direcciones físicas reales.



1. REUBICACION

Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Eduación S.A. Madrid, 2001 ISBN: 84-205-3177-4

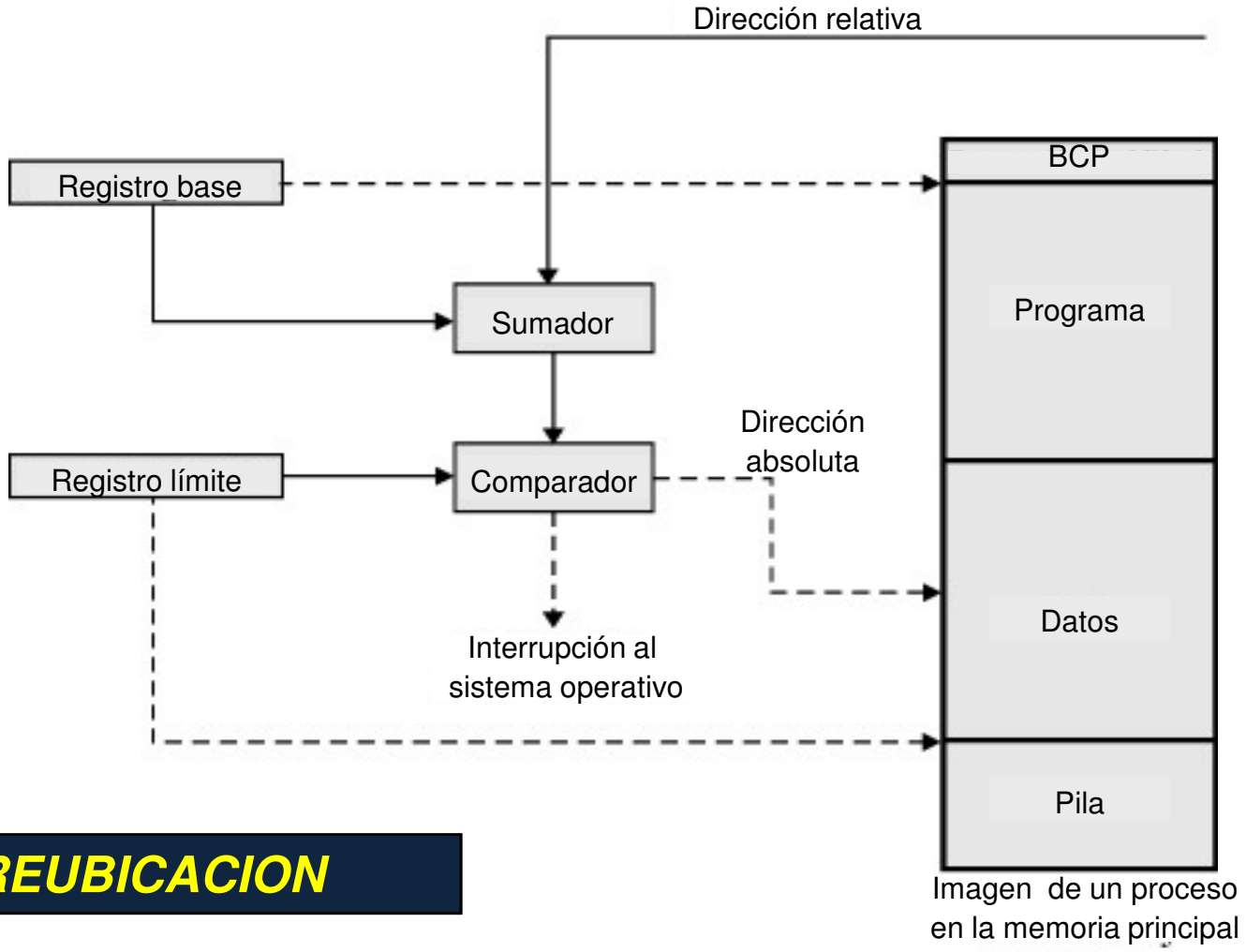
Figura 7.1. Requisitos de dirección para un proceso.

1. REUBICACION

- Cuando el proceso se carga en la memoria, se determina la ubicación real (absoluta) de la memoria.
- Un proceso puede ocupar diferentes particiones, lo que significa diferentes posiciones absolutas de la memoria durante su ejecución (a partir de la carga).
- La compactación también hará que un programa ocupe una partición distinta, lo que significa que las ubicaciones absolutas de la memoria cambien.

1. REUBICACION

- Dirección lógica:
  - Es una referencia a una posición de memoria independiente de la asignación actual de datos a la memoria.
  - Se debe hacer una traducción a una dirección física.
- Dirección relativa:
  - La dirección se expresa como una posición relativa a algún punto conocido.
- Dirección física:
  - La dirección absoluta o la posición real en la memoria principal.



1. REUBICACION

Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Eduación S.A. Madrid, 2001 ISBN: 84-205-3177-4

Figura 7.8. Soporte de hardware para la reubicación.

1. REUBICACION

- Registro base:
  - Se carga con la dirección en la memoria principal del proceso.
- Registro límite:
  - Indica la posición final del programa.
- Estos valores deben asignarse cuando se carga el programa y cuando se carga el proceso.

1. REUBICACION

- Se añade el valor del registro base a la dirección relativa para obtener una dirección absoluta.
- La dirección obtenida se compara con el valor del registro límite.
- Si la dirección no está dentro de los límites, se generará una interrupción en el sistema operativo.

2. PROTECCION

- El código de un proceso no puede hacer referencia a posiciones de memoria de otros procesos sin permiso.
- Es imposible comprobar las direcciones absolutas de los programas, puesto que se desconoce la ubicación de un programa en la memoria principal.
- Debe comprobarse durante la ejecución:
  - El sistema operativo no puede anticiparse a todas las referencias a la memoria que hará un programa.

3. COMPARTICION

- Permite el acceso de varios procesos a la misma zona de la memoria principal.

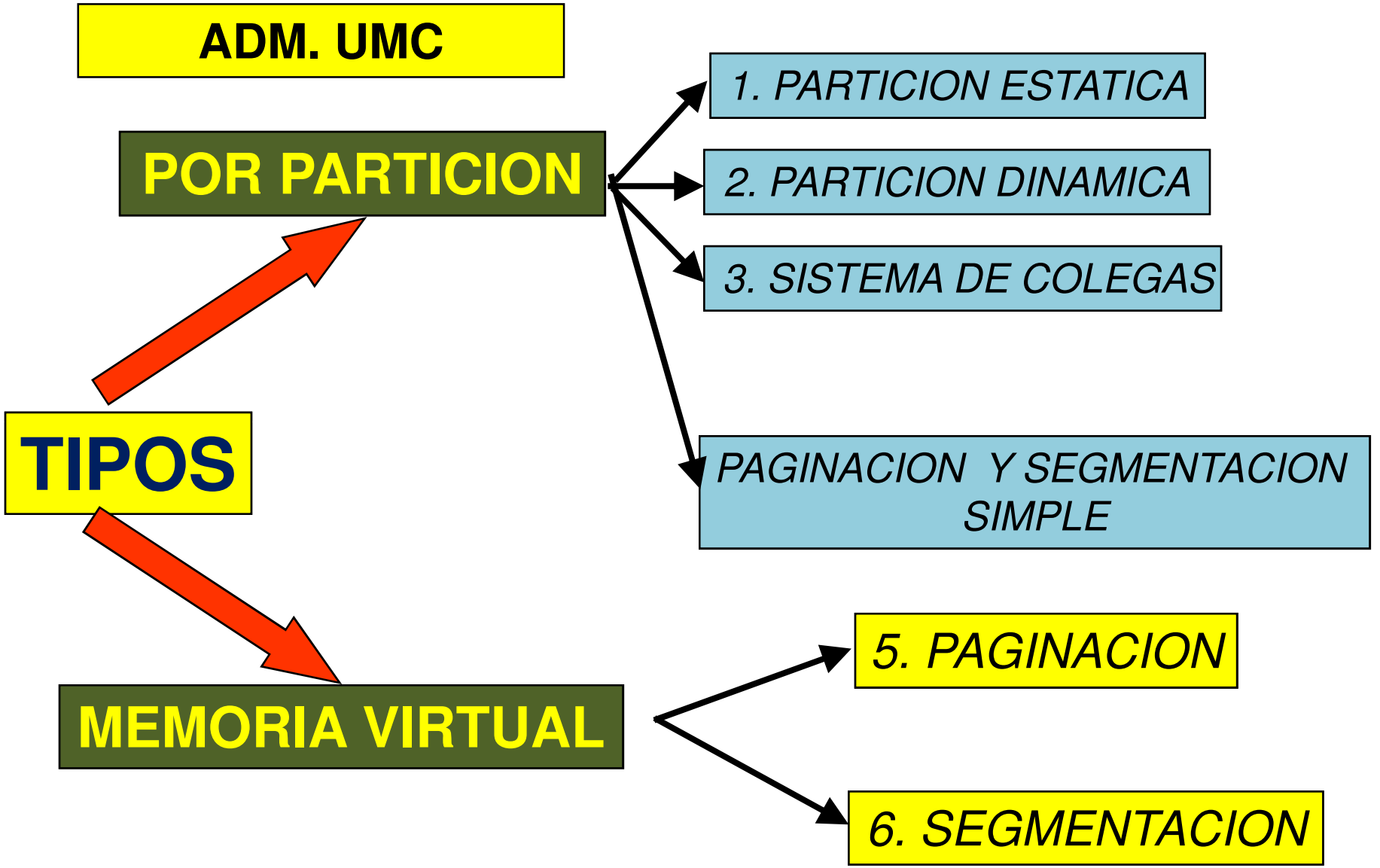
- Es mejor permitir a cada proceso (persona) que acceda a la misma copia del programa, en lugar de tener cada uno su propia copia aparte.

4. ORGANIZACION LOGICA

- La mayoría de los programas se organizan en módulos.
- Los módulos pueden escribirse y compilarse independientemente.
- Pueden otorgarse distintos grados de protección (sólo lectura, sólo ejecución) a los módulos.
- Compartir módulos.

5. ORGANIZACION FISICA

- La memoria disponible para un programa y sus datos puede ser insuficiente:
  - La superposición permite que varios módulos sean asignados a la misma región de memoria.
- El programador no conoce cuánto espacio habrá disponible.


ADM. UMC
POR PARTICION
1. PARTICION ESTATICA
2. PARTICION DINAMICA
3. SISTEMA DE COLEGAS
TIPOS
PAGINACION Y SEGMENTACION SIMPLE
MEMORIA VIRTUAL
5. PAGINACION
6. SEGMENTACION

2. A.M. POR PARTICIONES DINAMICAS

- Las particiones son variables en número y longitud.
- Al proceso se le asigna exactamente tanta memoria como necesite.
- Finalmente, hay varios huecos en la memoria. Este fenómeno se denomina fragmentación externa.
- Se debe usar la compactación para desplazar los procesos que estén contiguos, de forma que toda la memoria libre quede junta en un bloque.

2. A.M. POR PARTICIONES DINAMICAS

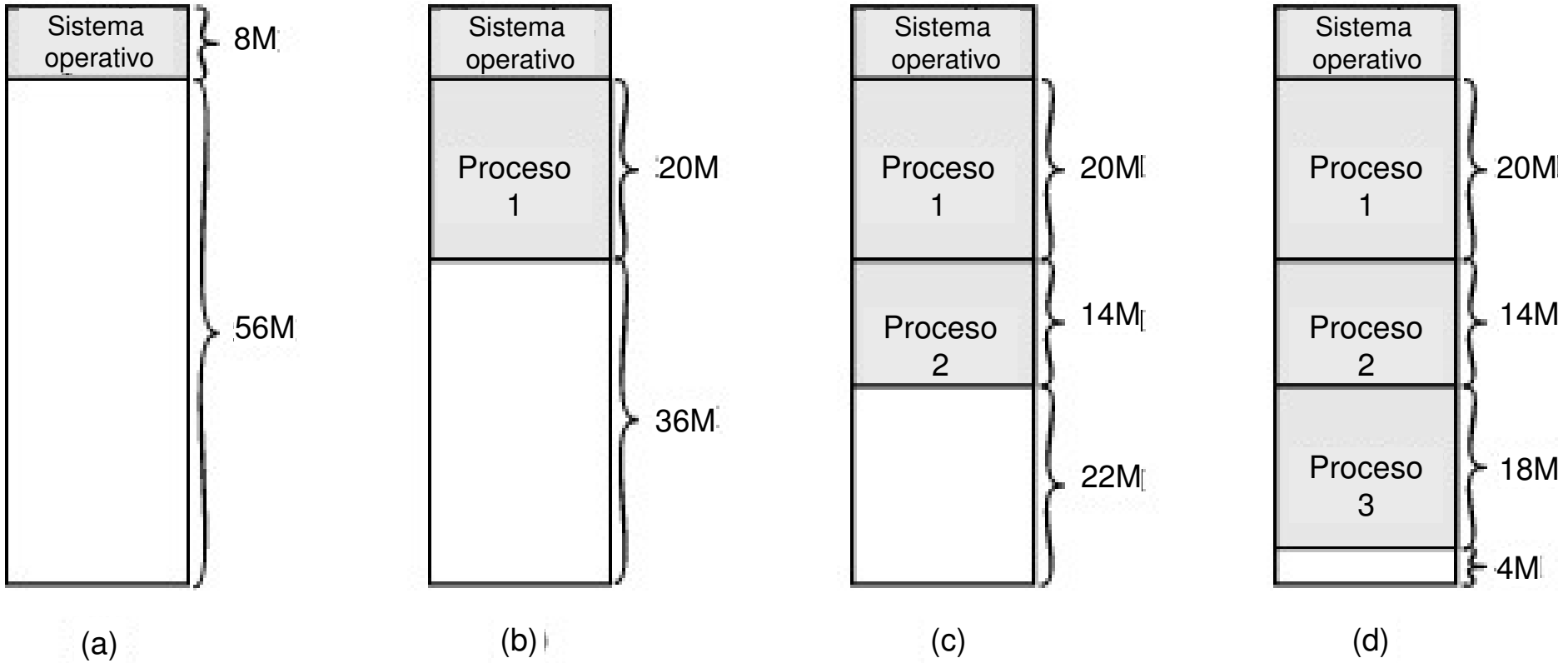


Figura 7.4. Efectos de la partición dinámica.

2. A.M. POR PARTICIONES DINAMICAS

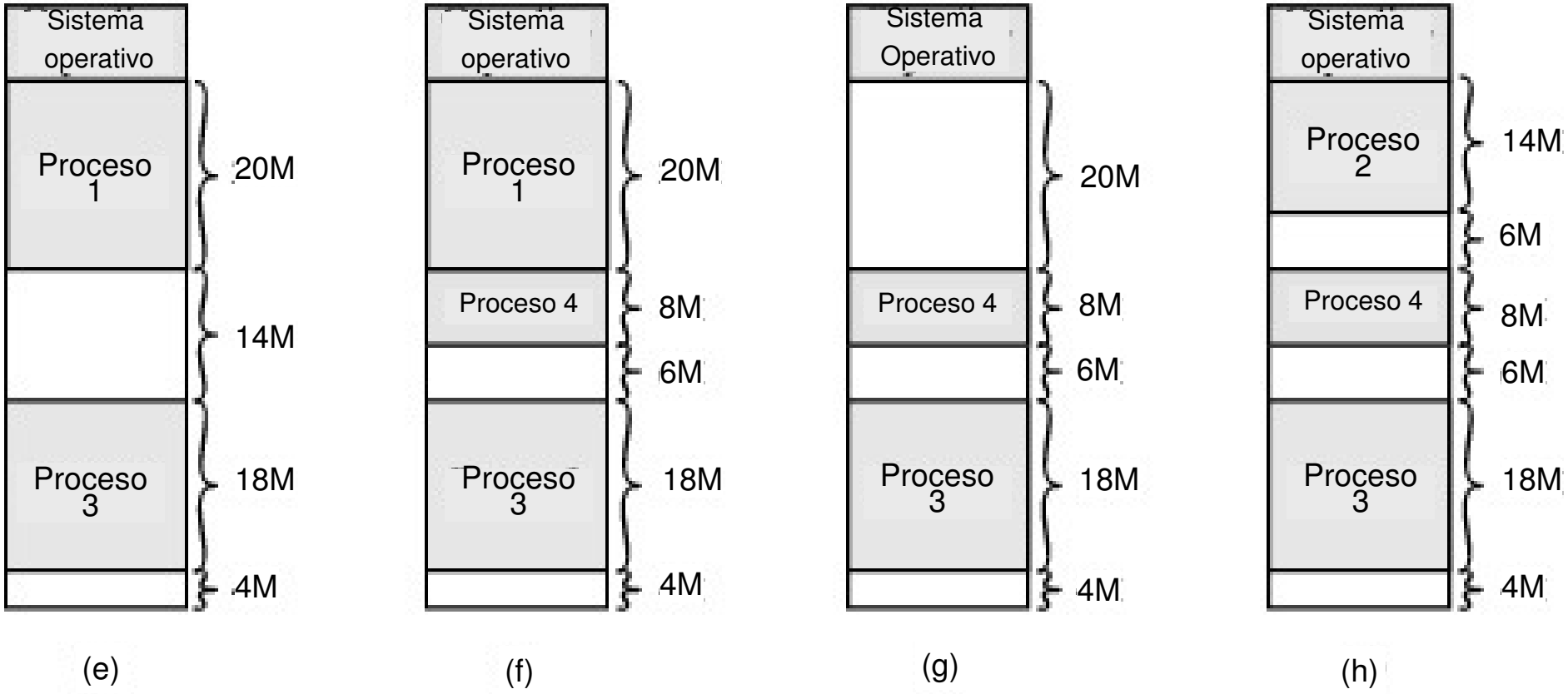
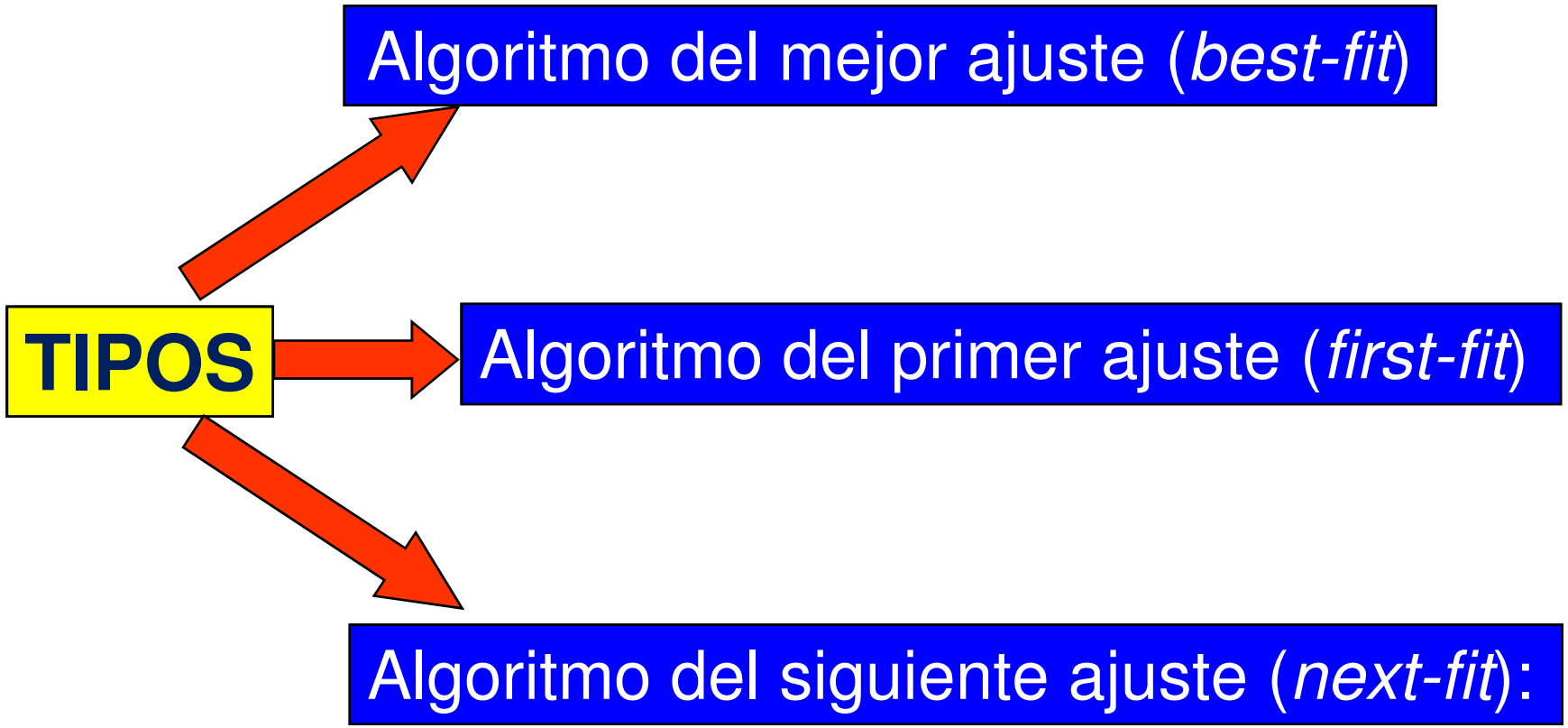


Figura 7.4. Efectos de la partición dinámica.

2. A.M. POR PARTICIONES DINAMICAS: ALGORITMOS DE ASIGNACION



2. A.M. POR PARTICIONES DINAMICAS: ALGORITMOS DE ASIGNACION

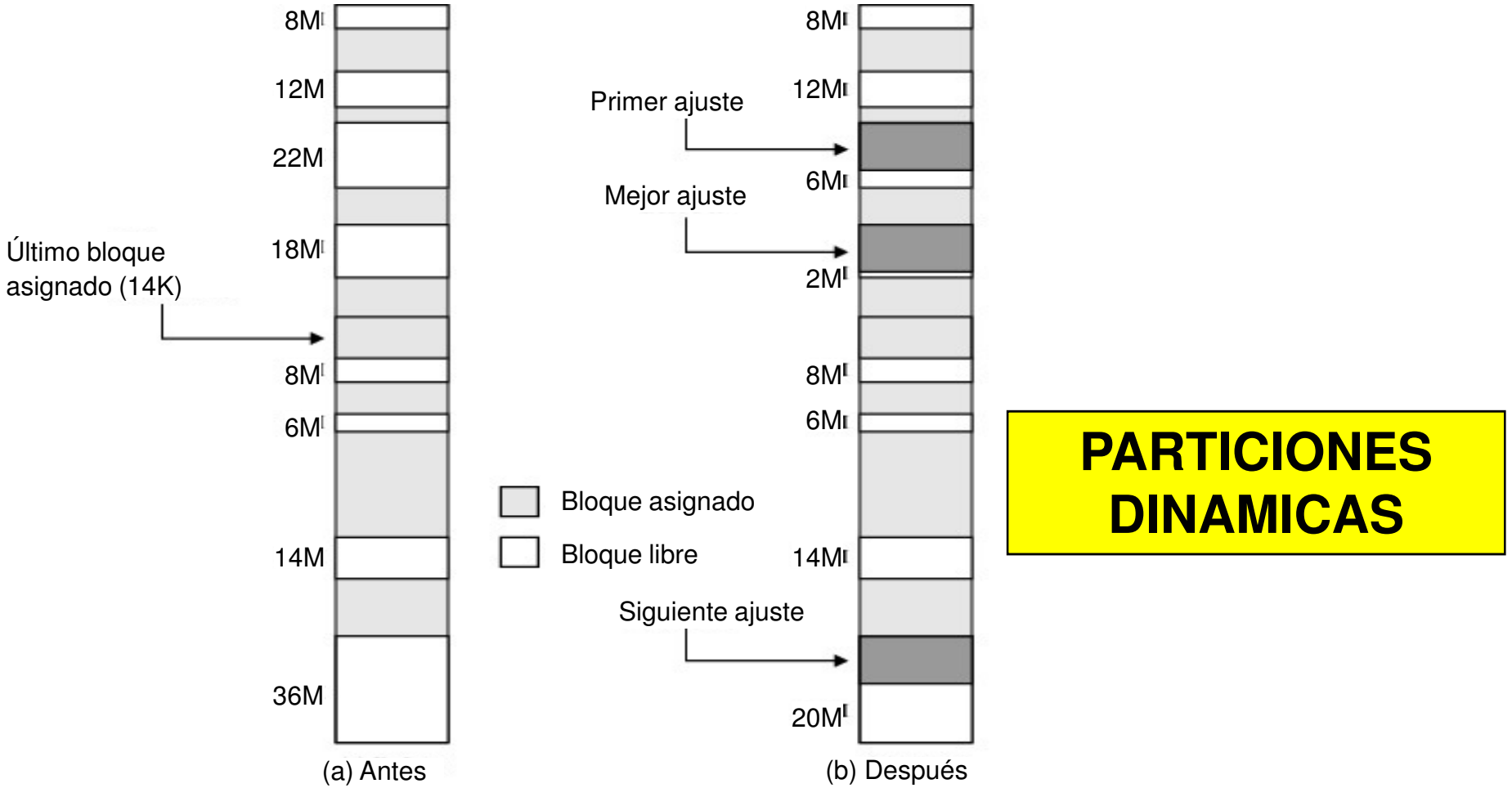
- **Algoritmo del mejor ajuste (*best-fit*):**
  - Elige el bloque de tamaño más próximo al solicitado.
  - Proporciona en general los peores resultados.
  - Puesto que este algoritmo busca el hueco más pequeño para el proceso, garantiza que el fragmento que se deja es lo más pequeño posible y, por lo tanto, se debe compactar más frecuentemente.

2. A.M. POR PARTICIONES DINAMICAS: ALGORITMOS DE ASIGNACION

- **Algoritmo del primer ajuste (*first-fit*):**
  - Es más rápido.
  - Puede tener varios procesos cargados en el extremo inicial de la memoria que es necesario recorrer cuando se intente encontrar un bloque libre.

2. A.M. POR PARTICIONES DINAMICAS: ALGORITMOS DE ASIGNACION

- **Algoritmo del siguiente ajuste (*next-fit*):**
  - Lleva frecuentemente a la asignación de un bloque de memoria de la última ubicación, donde se encuentra el bloque más grande.
  - El bloque de memoria más grande se divide en fragmentos pequeños.
  - Hará falta la compactación para obtener un bloque de memoria grande al final del espacio de memoria.

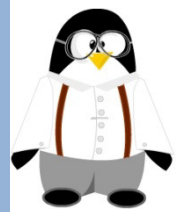


PARTICIONES DINAMICAS

Figura 7.5. Ejemplo de una configuración de memoria antes y después de asignar un bloque de 16 Mbytes.

BIBLIOGRAFIA DE REFERENCIA

1. Programación en Linux, con ejemplos. Kurt Wall. QUE, Prentice Hall. Madrid. 2000.
2. Sistemas Operativos. 5ta Ed. William Stalling. Pearson Prentice Hall. Madrid. 2006
3. Sistemas Operativos. 7ma Ed. William Stalling. Pearson Prentice Hall. Madrid. 2012
4. Sistemas Operativos Modernos. Andrew. S. Tanenbaum. Prentice-Hall. Interamericana S.A. Madrid, 2009.
5. Unix, Sistema V Versión 4. Rosen,Rozinsky y Farber.McGraw Hill. NY 2000.
6. Lunix, Edición especial. Jack Tackett, David Guntery Lance Brown. Ed. Prentice Hall. 1998.
7. El Libro de Linux. Syed M. Sarwar, Robert Koretsky y Syed. A. Sarwar. Ed. Addison Wesley. 2007. España.



Lic. en Sistemas de Información

FIN UNIDAD 6 (Parte A)

GESTION DE MEMORIA