

# Administración de Memoria

Sistemas Operativos

2° año Ing. en Sistemas de Información

Universidad Tecnológica Nacional Facultad Regional Villa María



# Administración de Memoria

- „Los programas crecen mas que la memoria“
- SO crea abstracción de la memoria
- Usuario y programas:
  - Memoria rápida, barata, grande, no volatil
- Jerarquía de la memoria
  - Cache
  - Ram
  - Discos
  - Discos removibles



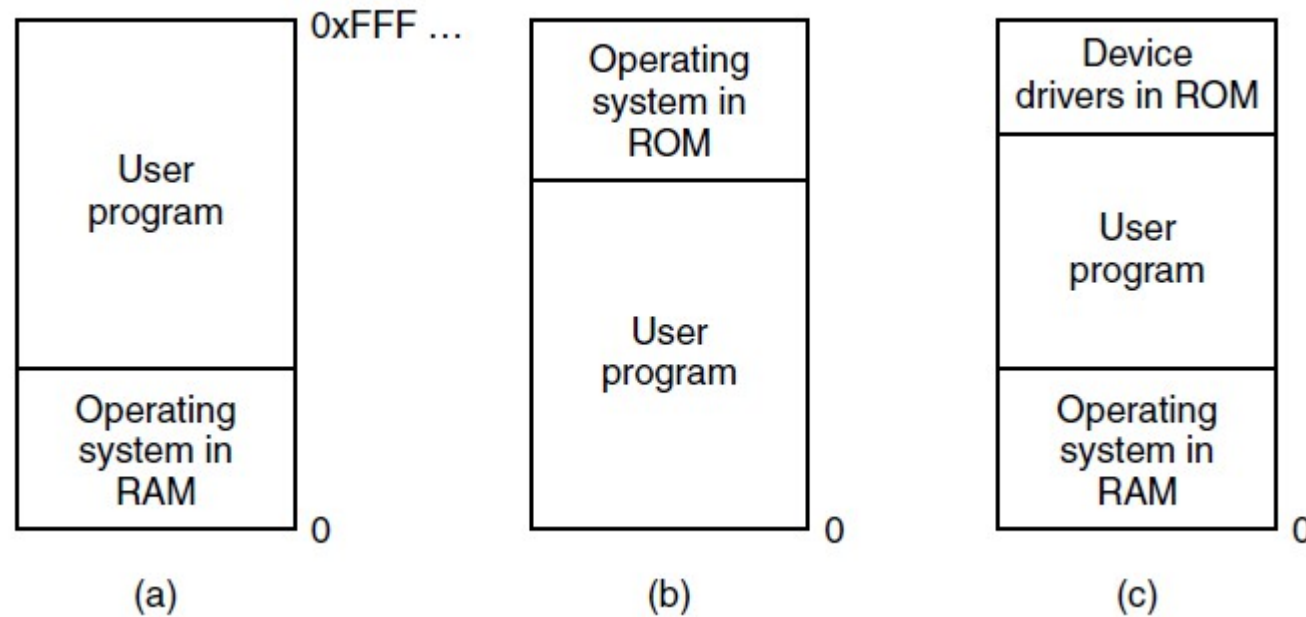
# Administración de Memoria

- Administración de memoria:
  - Administrar jerarquía
  - Asignar programas
  - Desasignar programas
- Sin abstracción de memoria
  - Se ve la memoria física desde el programa
  - No hay multiprogramación



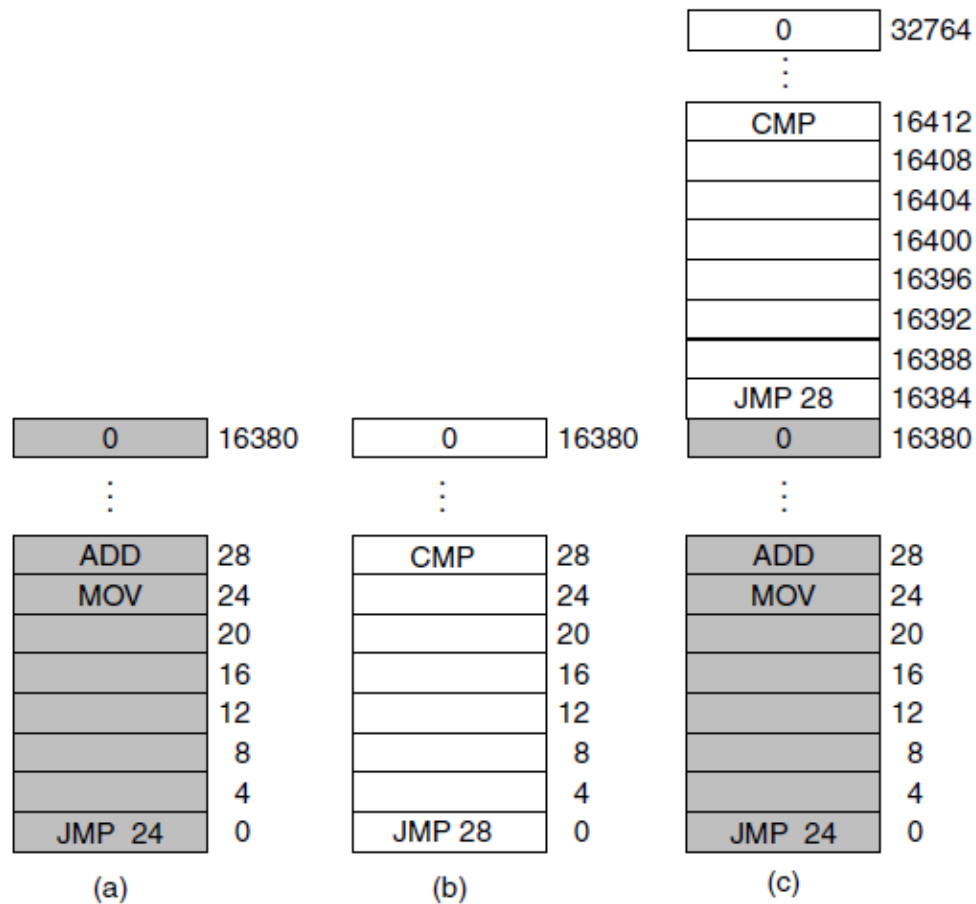
# Administración de Memoria

- Sin abstracción de memoria



# Administración de Memoria

- Múltiples programas sin abstracción de memoria

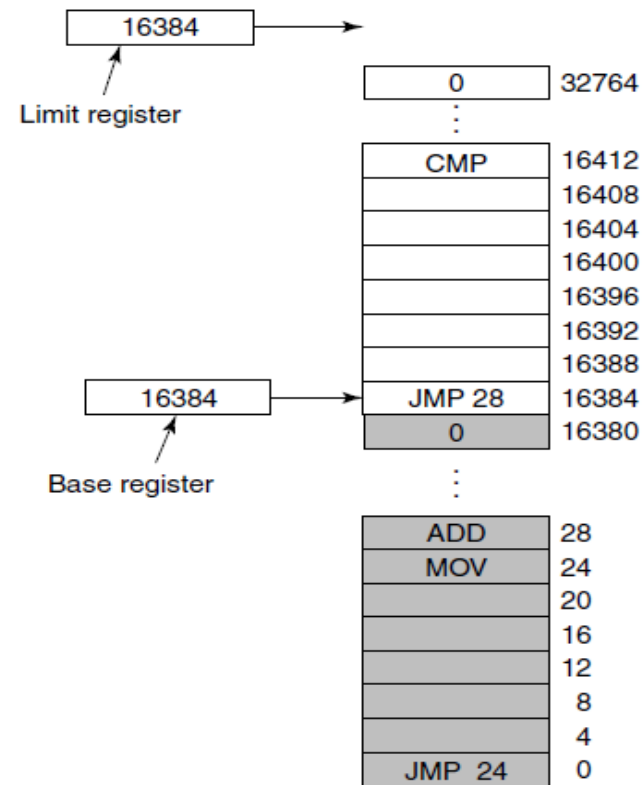


- Alternativa: reubicación estática  
sumar el registro de comienzo a la  
dirección del programa



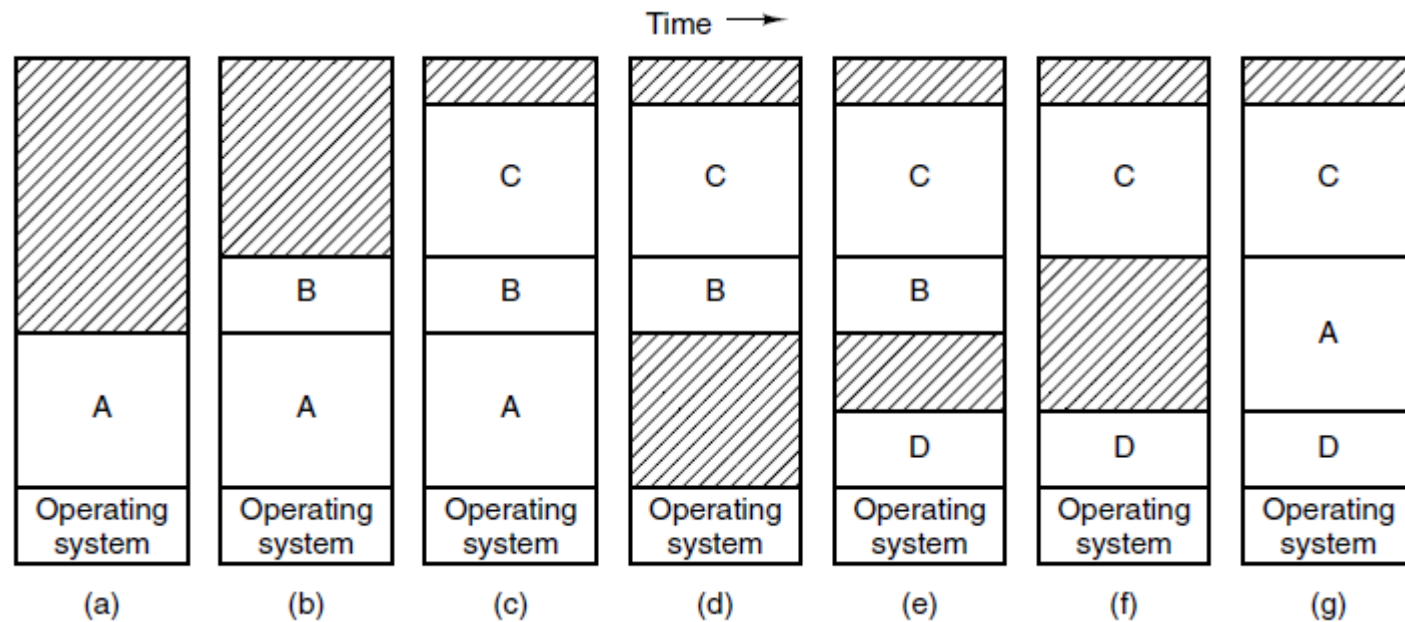
# Administración de Memoria

- Abstracción de memoria
  - Espacio de direcciones.
  - La dirección 28 de un programa es diferente a la 28 de otro.
  - Registro base y límite

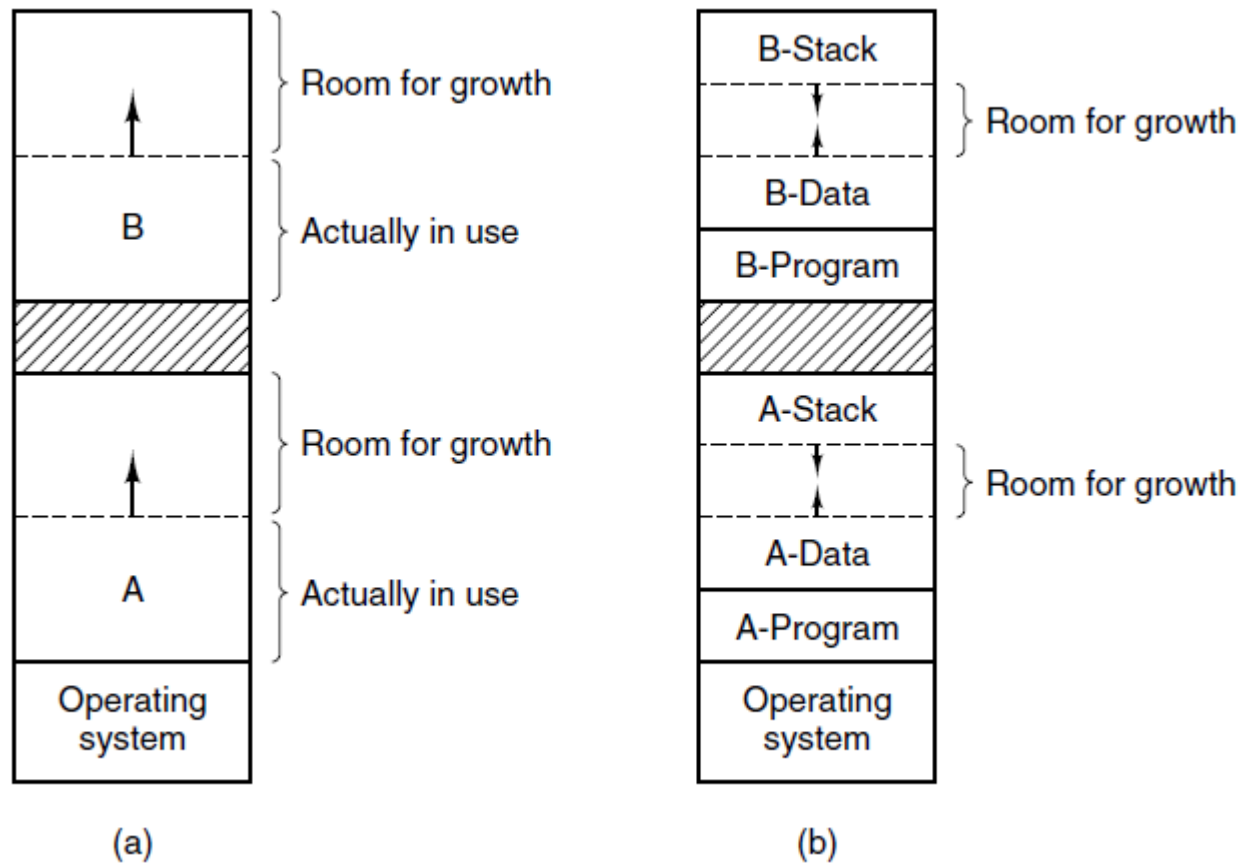


# Intercambio

- Intercambio
- Memoria virtual



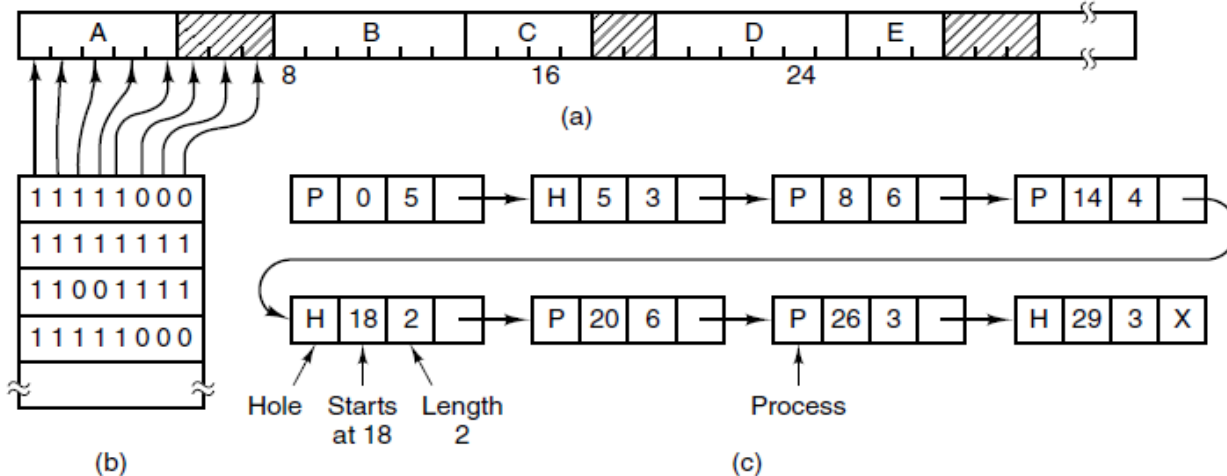
# Intercambio





# Administración de memoria libre

- Mapa de bits
- Listas enlazadas



# Algoritmos listas enlazadas

- Primer ajuste (desde el principio)
- Siguiente ajuste (continua del punto anterior)
- Mejor ajuste: (el hueco que mejor se ajusta)
- Peor ajuste: (el hueco mas grande posible)
- Alternativa: tener dos listas separas. Una para los huecos y otra procesos



# Memoria Virtual

- Agrandamiento del software
- Capacidades limitadas
- Alternativa: overlays (sobrepuestos)
- Memoria virtual: dividir en trozos llamados páginas. No hace falta que todas estén en la memoria.



# Memoria Virtual

Todas las referencias a la memoria se traducirán dinámicamente a direcciones físicas durante la ejecución:

Un proceso puede cargarse y descargarse de la memoria principal de tal forma que ocupe regiones diferentes.

Un proceso puede dividirse en varias partes y no es necesario que estas partes se encuentren contiguas en la memoria principal durante la ejecución:

No será necesario que todas las páginas o todos los segmentos de un proceso estén en la memoria durante la ejecución.



# Memoria Virtual

El sistema operativo comienza trayendo sólo unos pocos fragmentos del programa.

El conjunto residente es la parte de un proceso que está realmente en la memoria principal.

Si el procesador encuentra una dirección lógica que no está en la memoria principal, genera una interrupción que indica un fallo de acceso a la memoria.

El sistema operativo pone al proceso interrumpido en estado Bloqueado.

El sistema operativo necesita traer a la memoria principal el fragmento del proceso que contiene la dirección lógica que provocó el fallo de acceso:

El sistema operativo emite una solicitud de Lectura de E/S al disco.

El sistema operativo puede expedir otro proceso para que se ejecute mientras realiza la operación de E/S.

Una vez que el fragmento deseado se ha traído a la memoria principal y se ha emitido la interrupción de E/S, se devuelve el control al sistema operativo, que coloca el proceso afectado en el estado de Listo.



# Memoria Virtual

Se pueden mantener más procesos en la memoria principal:

Se cargan sólo algunos fragmentos de un proceso particular.

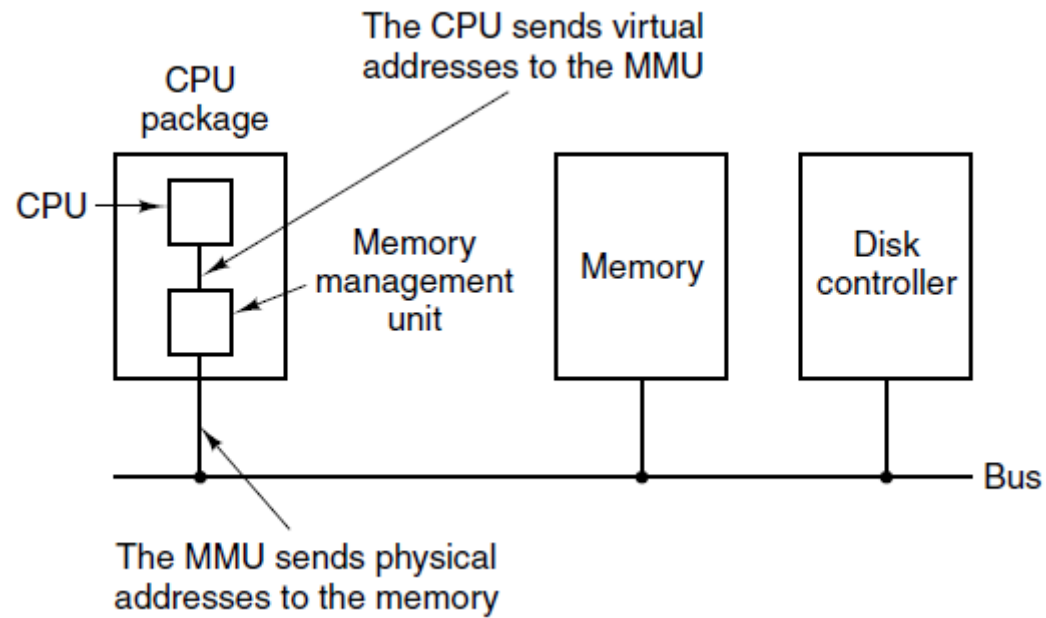
Con tantos procesos en la memoria principal es muy probable que uno de los procesos esté en estado Listo en un instante determinado.

Es posible que un proceso sea más grande que toda la memoria principal.

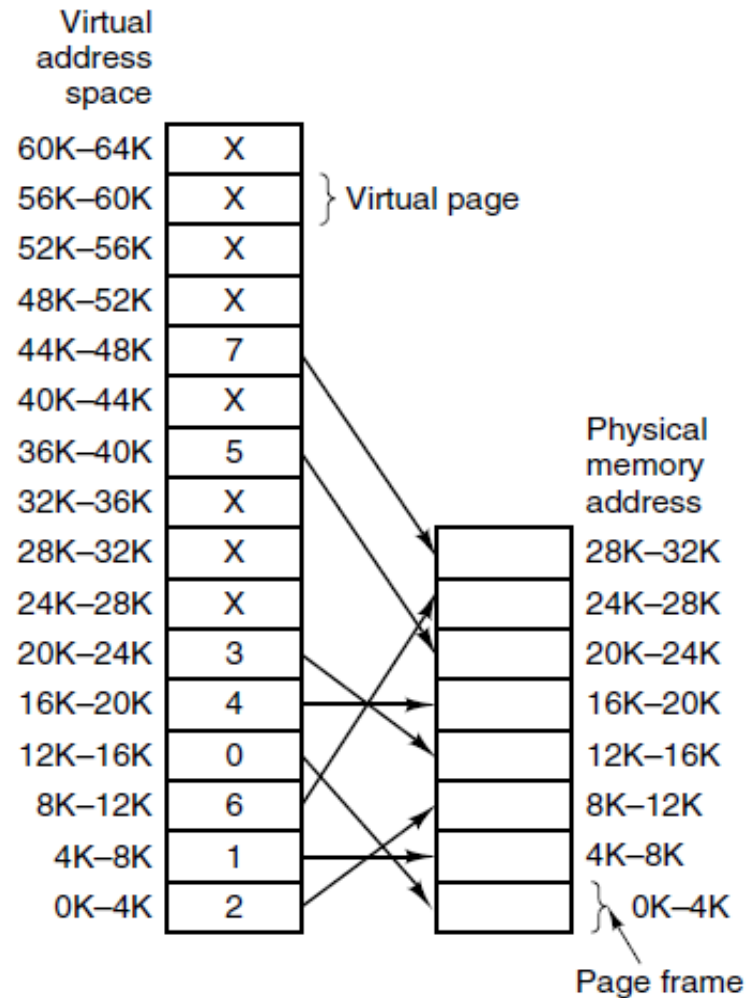
- **Hiperpaginación:**  
El sistema operativo expulsa un fragmento de un proceso justo antes de ser usado.  
El procesador consume más tiempo intercambiando fragmentos que ejecutando instrucciones de usuario.



# Paginación



# Paginación

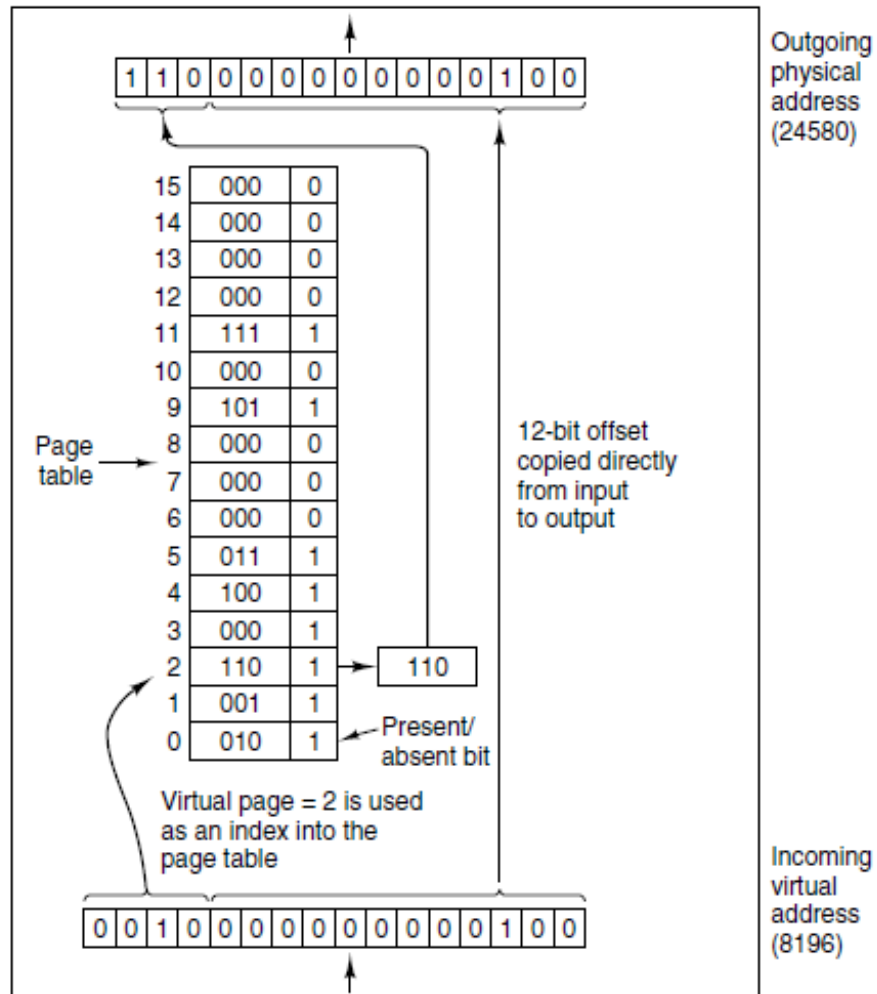


- 64KB virtuales
- 32KB físico
- 16 páginas virtuales
- 8 marcos de páginas
- Tamaño del marco 4KB

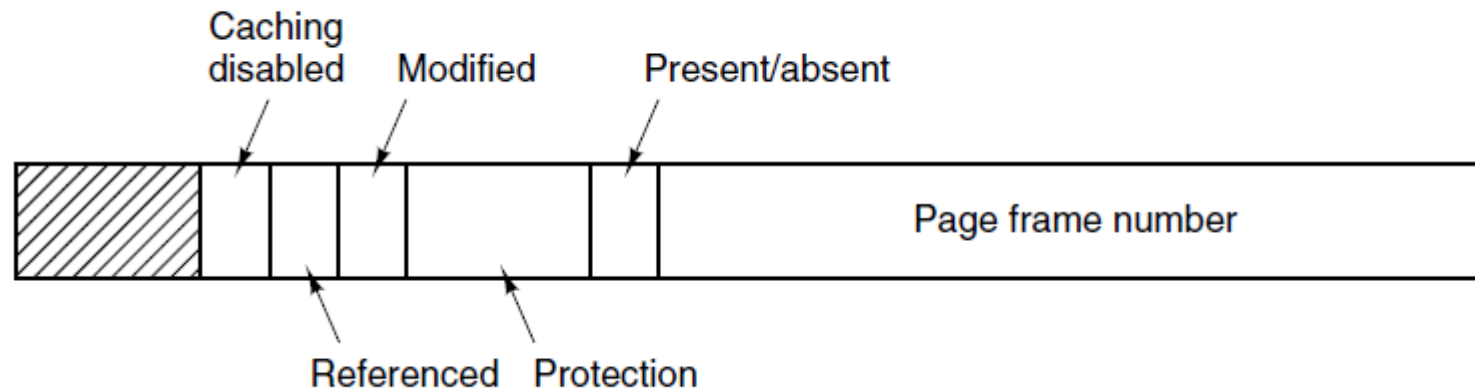




# Paginación



# Entrada en la tabla de páginas



# Aceleración de la paginación

1. La asociación de una dirección virtual a una dirección física debe ser rápida
2. Si el espacio de direcciones virtuales es grande, la tabla de páginas será grande.

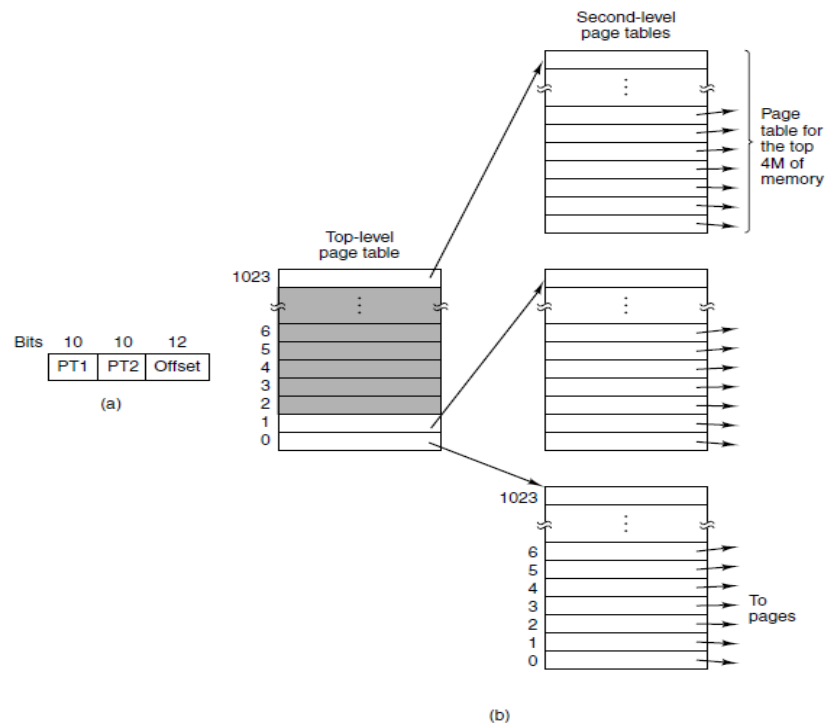
Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

- Buffer de traducción adelantada



# Tablas de páginas para memorias extensas

- Tablas de páginas multinivel
- Tablas de páginas invertidas



# Algoritmos de reemplazo de páginas

- Algoritmo de reemplazo de páginas: Óptimo
  - Eliminar la página que no se referenciará por mucho tiempo.
  - No se puede implementar porque no se puede conocer a priori cuándo se va a referenciar



# Algoritmos de reemplazo de páginas

- Algoritmo de reemplazo de páginas: No Usadas Recientemente (NRU)

Clase 0: no ha sido referenciada, no ha sido modificada

Clase 1: no ha sido referenciada, ha sido modificada

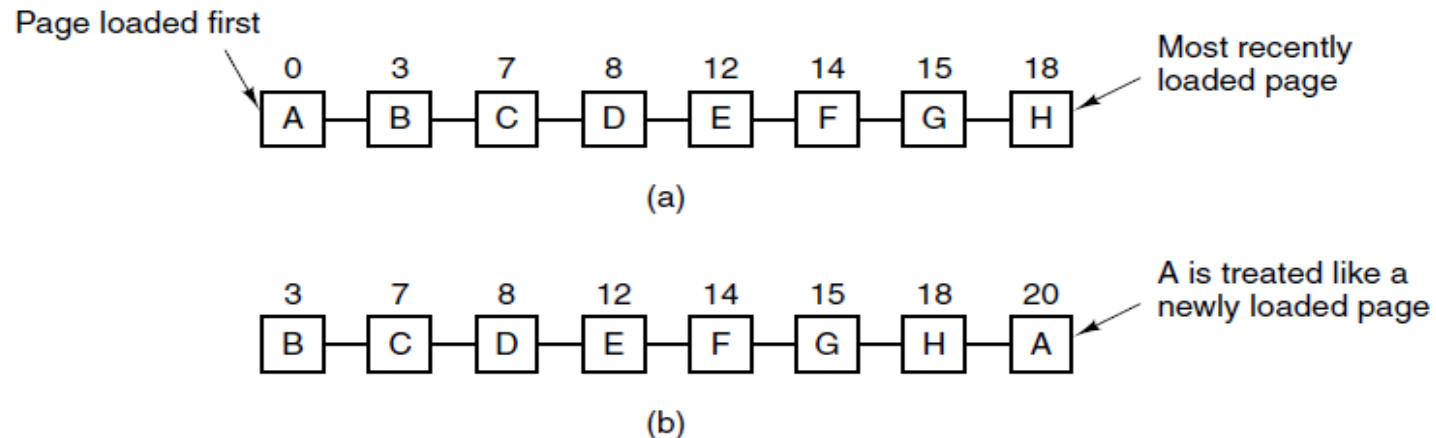
Clase 2: ha sido referenciada, no ha sido modificada

Clase 3: ha sido referenciada, ha sido modificada



# Algoritmos de reemplazo de páginas

- Algoritmo de reemplazo de páginas: Segunda Oportunidad
  - Se elimina la página mas vieja que no ha sido referenciada. Variación de FIFO.

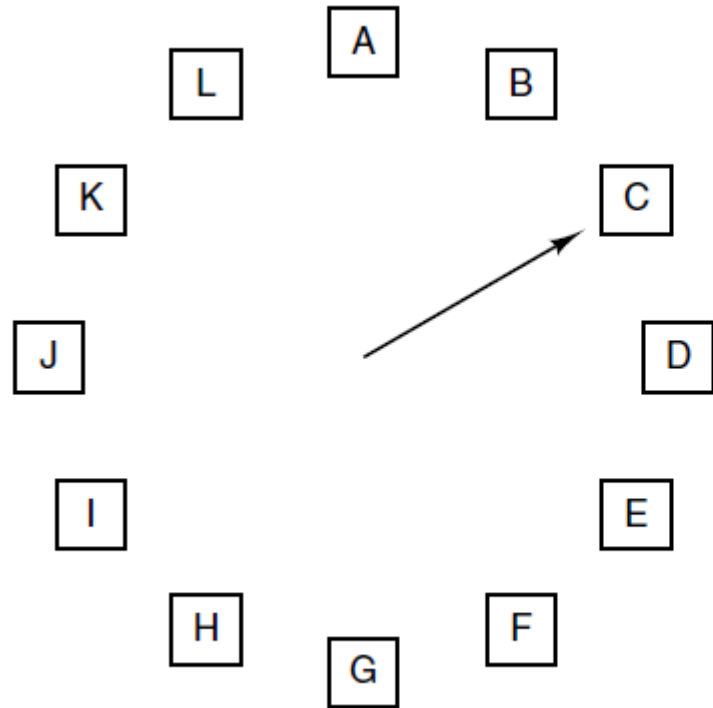


A ha sido referenciada, se pone al final y se desactiva R



# Algoritmos de reemplazo de páginas

- Algoritmo de reemplazo de páginas: Reloj



When a page fault occurs,  
the page the hand is  
pointing to is inspected.  
The action taken depends  
on the R bit:

R = 0: Evict the page

R = 1: Clear R and advance hand





# Algoritmos de reemplazo de páginas

- Algoritmo de reemplazo de páginas: Menos Utilizadas Recientemente
  - Descartar la página que menos se utiliza en el tiempo
  - Contador de referencia a la página. Se elimina la menor.



# Cuestiones de Diseño para los sistemas de Paginación

- Política de Asignación Local vs Global

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

(a)

A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

(b)

Local

A0
A1
A2
A3
A4
A5
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

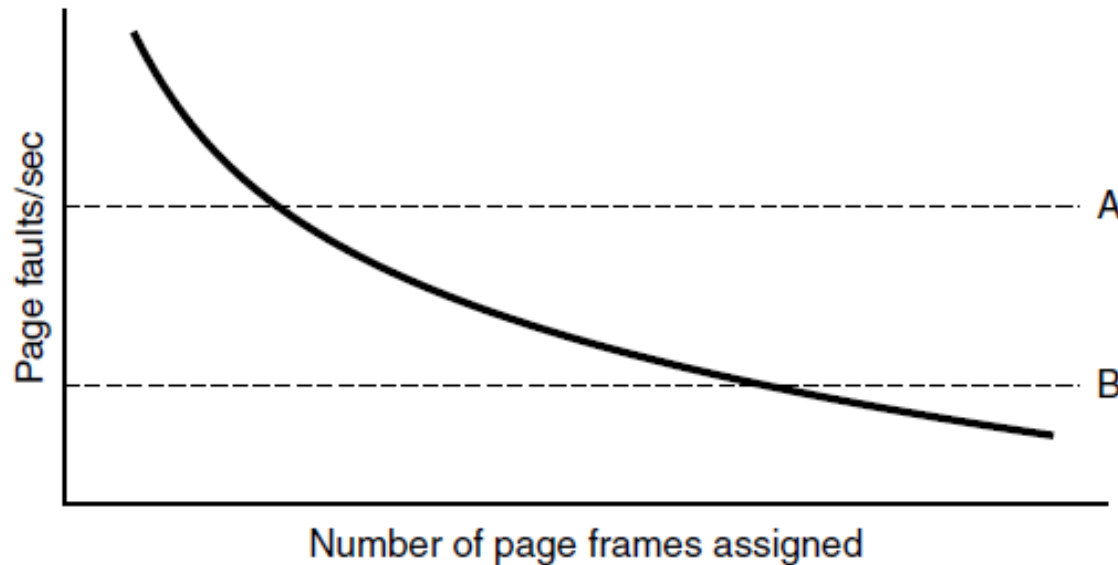
(c)

Global



# Cuestiones de Diseño para los sistemas de Paginación

- Proporción de fallos de página como una función de marcos de página de asignación



# Cuestiones de Diseño para los sistemas de Paginación

## Control de Carga

- Grado de multiprogramación
- Sobrepaginación: continuos fallos de página del proceso
- Alternativa: Intercambio

## Tamaño de la página

Páginas compartidas (mas de una instancia del programa..)

Bibliotecas compartidas

Archivos compartidos

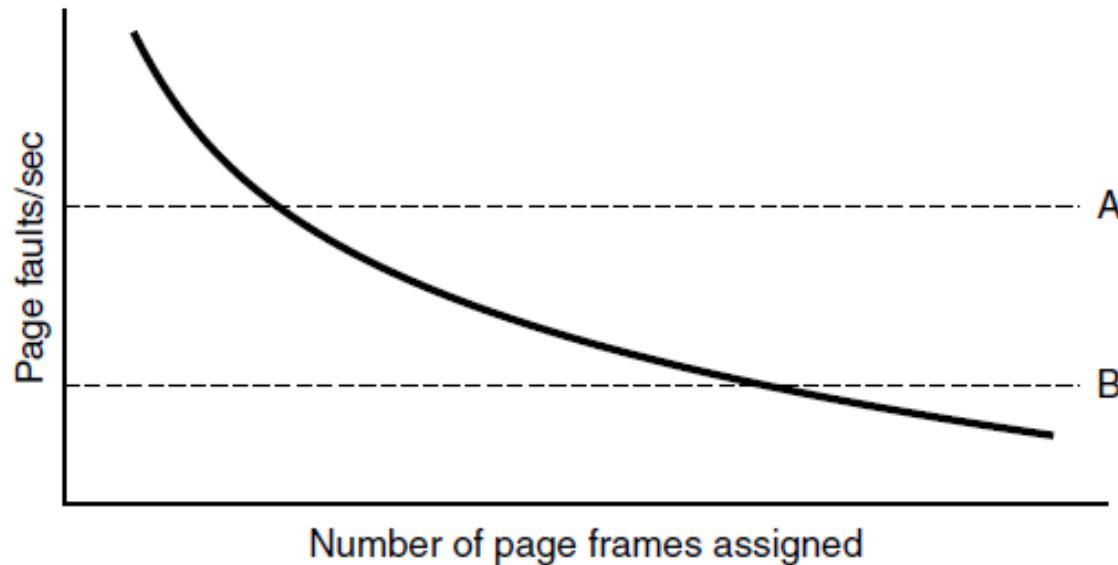
Politica de limiyeza (demonio de paginación)

Interfaz de memoria virtual (el programador „podria a llegar a tener cierto grado de control“)



# Cuestiones de Diseño para los sistemas de Paginación

- Proporción de fallos de página como una función de marcos de página de asignación

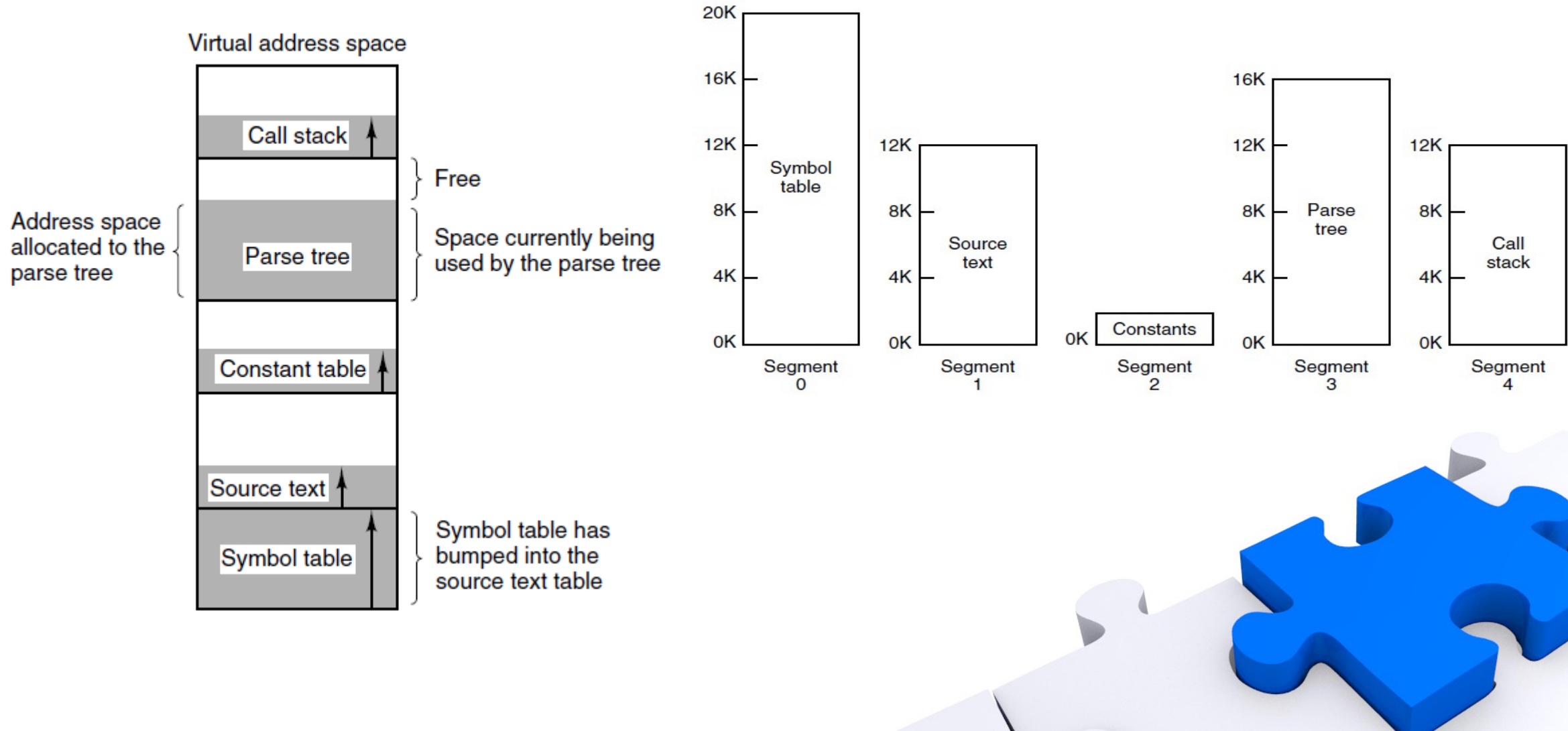


# Segmentación

- No es necesario que todos los segmentos de todos los programas tengan la misma longitud.
- Existe una longitud máxima de segmento.
- Una dirección lógica segmentada consta de dos partes, un número de segmento y un desplazamiento.
- Como consecuencia del empleo de segmentos de distinto tamaño, la segmentación resulta similar a la partición dinámica.



# Segmentación





# Segmentación

Consideración	Paginación	Segmentación
¿Necesita el programador estar consciente de que se está utilizando esta técnica?	No	Sí
¿Cuántos espacios de direcciones lineales hay?	1	Muchos
¿Puede el espacio de direcciones total exceder al tamaño de la memoria física?	Sí	Sí
¿Pueden los procedimientos y los datos diferenciarse y protegerse por separado?	No	Sí
¿Pueden las tablas cuyo tamaño fluctúa acomodarse con facilidad?	No	Sí
¿Se facilita la compartición de procedimientos entre usuarios?	No	Sí
¿Por qué se inventó esta técnica?	Para obtener un gran espacio de direcciones lineal sin tener que comprar más memoria física	Para permitir a los programas y datos dividirse en espacios de direcciones lógicamente independientes, ayudando a la compartición y la protección





# Segmentación

Segment 4 (7K)
Segment 3 (8K)
Segment 2 (5K)
Segment 1 (8K)
Segment 0 (4K)

(a)

Segment 4 (7K)
Segment 3 (8K)
Segment 2 (5K)
(3K)
Segment 7 (5K)
Segment 0 (4K)

(b)

(3K)
Segment 5 (4K)
Segment 3 (8K)
Segment 2 (5K)
(3K)
Segment 7 (5K)
Segment 0 (4K)

(c)

(3K)
Segment 5 (4K)
(4K)
Segment 6 (4K)
Segment 2 (5K)
(3K)
Segment 7 (5K)
Segment 0 (4K)

(d)

(10K)
Segment 5 (4K)
Segment 6 (4K)
Segment 2 (5K)
Segment 7 (5K)
Segment 0 (4K)

(e)

