



# **Sistemas Operativos 1**

## **Administración De Memoria**



# **SIN ABSTRACCIÓN DE MEMORIA**

# SIN ABSTRACCIÓN DE MEMORIA

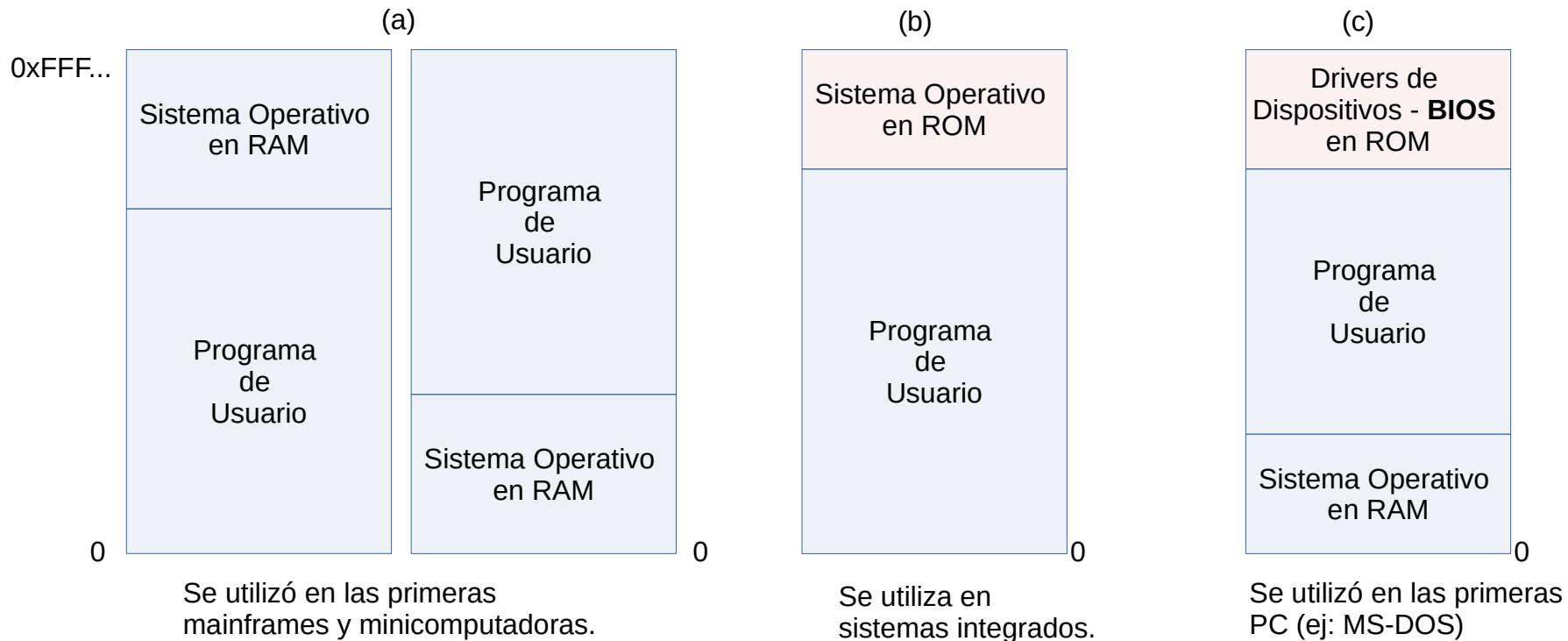
- Las primeras computadoras (mainframe <1960), (minicomputadoras <1970) y (pc <1980) no tenían abstracción de memoria.
- Cada programa veía simplemente la **memoria física**
- **Conjunto de direcciones** desde 0 hasta N (valor máximo)
- Cada dirección corresponde a una **celda** que contenía cierto número de bits (comúnmente ocho)
- No es posible tener **dos programas ejecutándose** en memoria al mismo tiempo.

```
MOV REGISTRO1, [1000]
```

la computadora sólo movía el contenido de la ubicación de memoria física 1000 a REGISTRO1

# SIN ABSTRACCIÓN DE MEMORIA

Tres formas simples de organizar la memoria con un sistema operativo y un proceso de usuario



Los modelos (a) y (c) tienen la desventaja de que un error en el programa de usuario puede borrar/dañar el sistema operativo, posiblemente con resultados desastrosos

# PSW Program Status Word

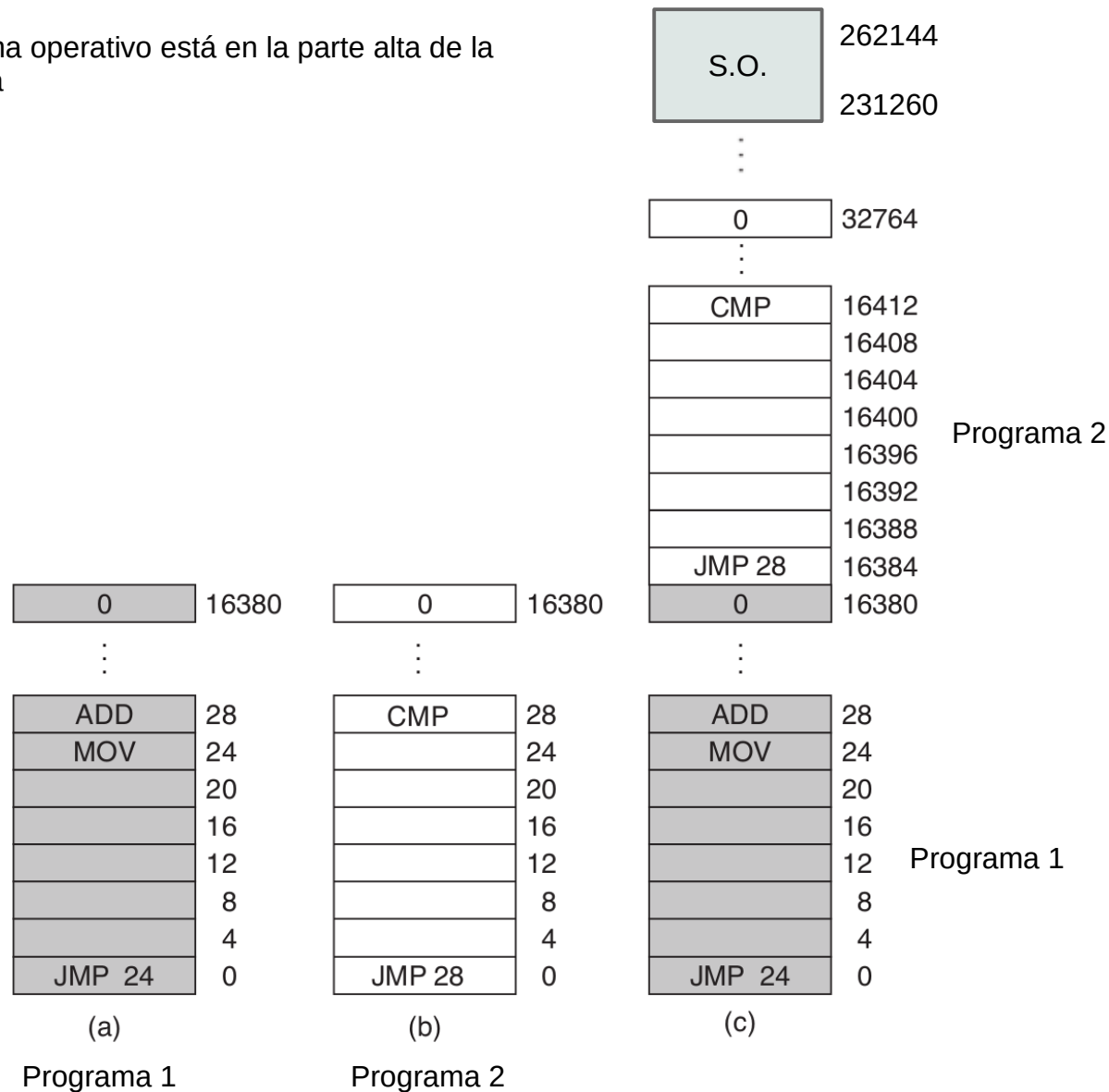
- La IBM 360 (1965-1978), resolvió el problema **incorporando** cierto **hardware especial (PSW - Program Status Word)** que controlaba cualquier intento por parte de un proceso en ejecución de acceder a la memoria.



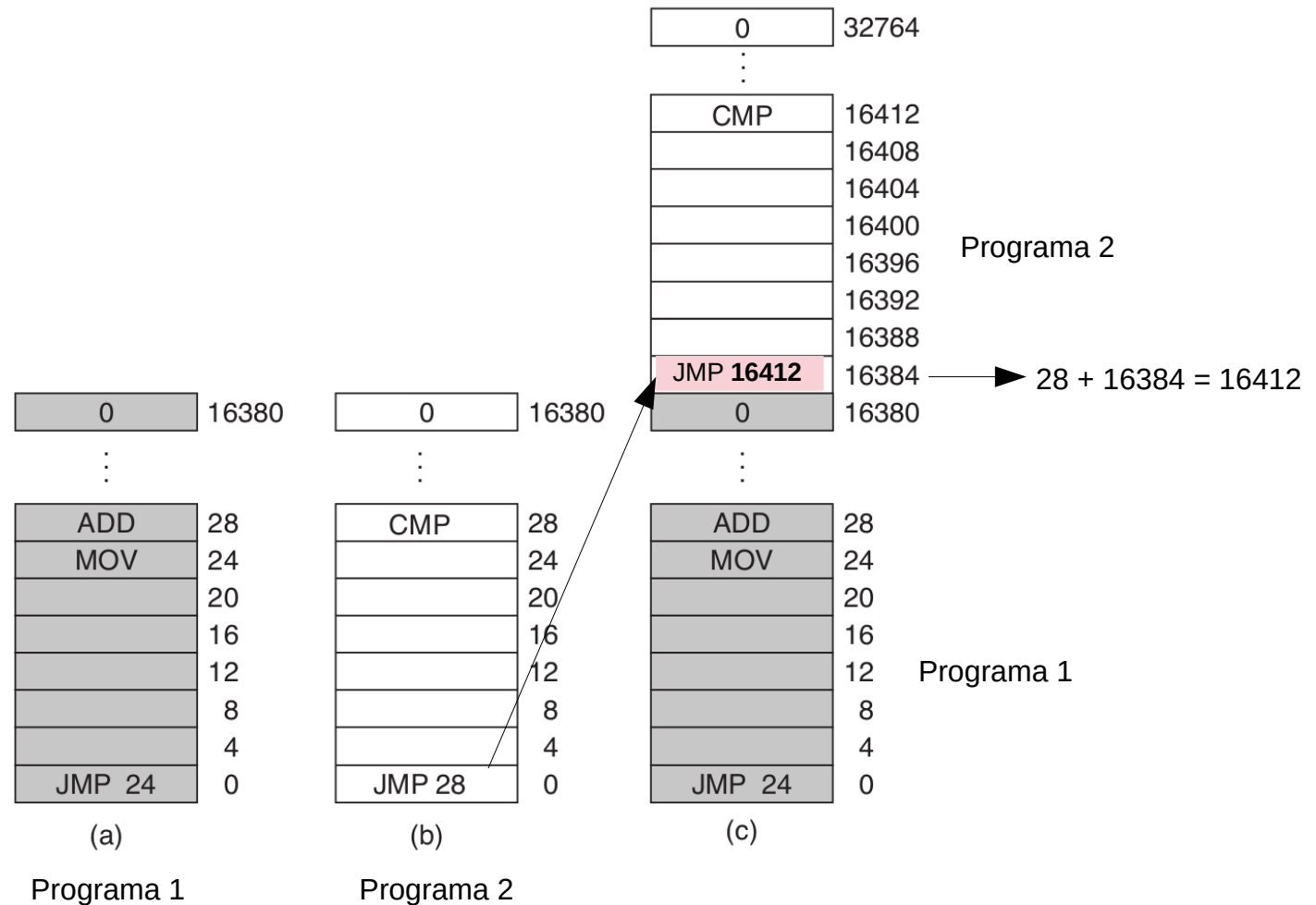
El hardware de la **IBM 360** controlaba mediante un trap cualquier intento por parte de un proceso en ejecución de acceder a la memoria con un código de protección distinto del de la llave del **PSW**

# Ejecución de múltiple programas sin una abstracción de memoria

el sistema operativo está en la parte alta de la memoria



# Reubicación Estática



La IBM 360 modificaba el segundo programa al instante, a medida que se cargaba en la memoria, usando una técnica conocida como **reubicación estática**.

# Reubicación Estática

- **Funciona** si se lleva a cabo en la forma correcta
- Reduce la **velocidad de la carga**
- Requiere **información adicional** en todos los programas ejecutables para indicar cuáles palabras contienen direcciones (reubicables) y cuáles no.
- Ej:

```
MOV REGISTRO1, [1000]
```

```
MOV REGISTRO1, 28
```

- El cargador necesita saber qué es una **dirección** y qué es una **constante**



# Exposición de Memoria Física

+ Muy Simple

- Los programas de usuario pueden **estropear el sistema** operativo con facilidad (intencional o accidentalmente)

- Es difícil tener varios programas en ejecución a la vez .  
problemas: **protección** y **reubicación**

+ PSW (Program Status Word - llaves)

- Reubicación estática: **LENTA** y **COMPLICADA**

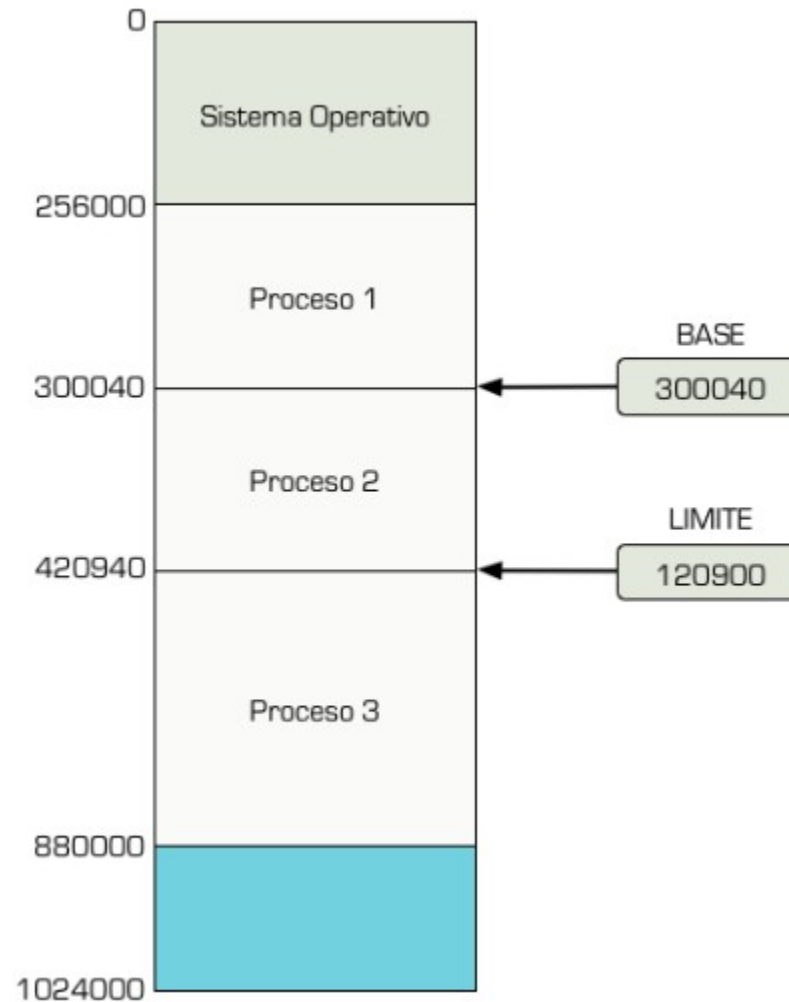


# ESPACIO DE DIRECCIONES LÓGICO VS. FÍSICO

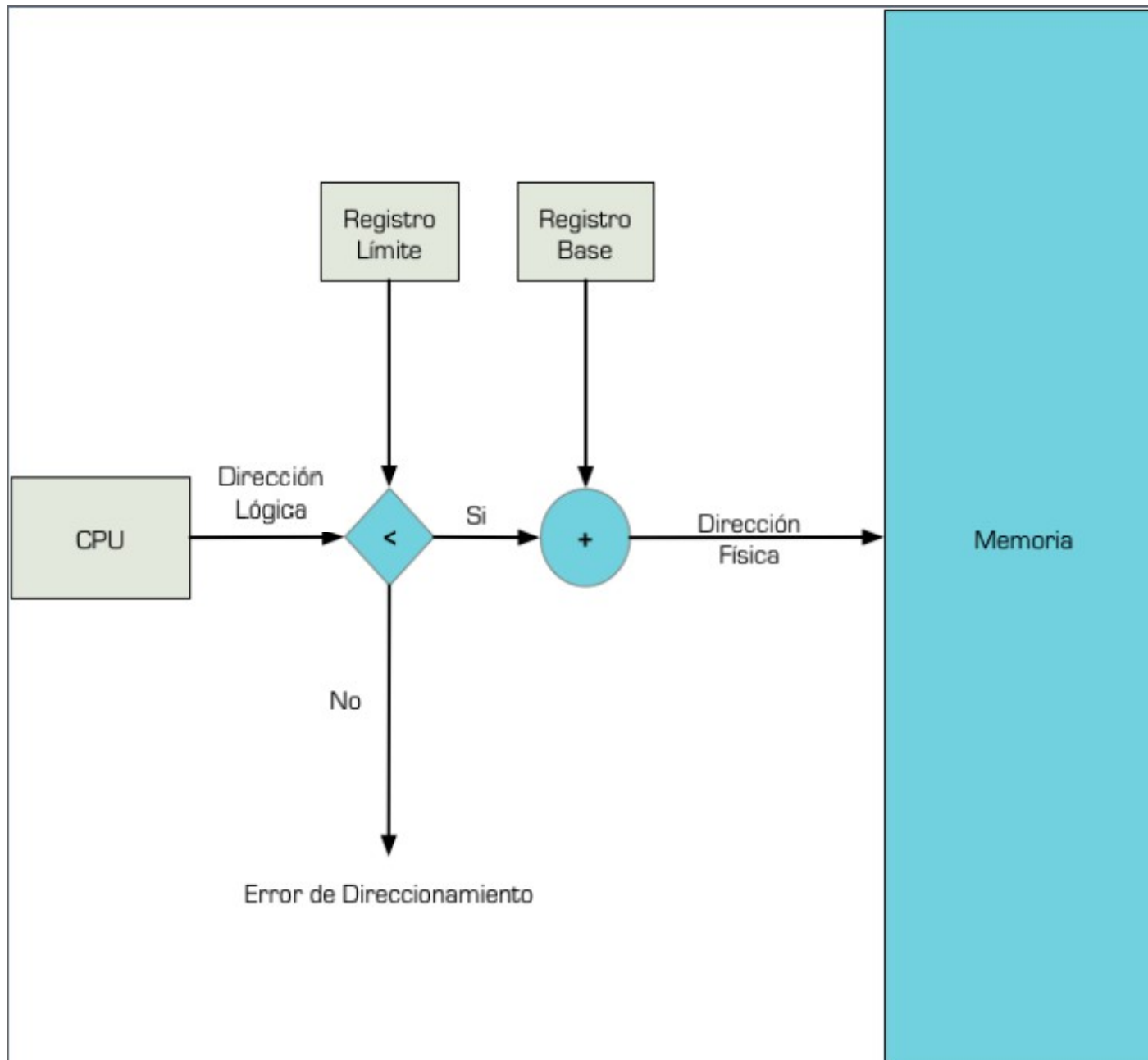
- El concepto de espacio de direcciones lógico que está separado de un espacio físico es central en la administración de memoria.
  - **Dirección lógica** – generada por la CPU; también conocida como **dirección virtual**
  - **Dirección Física** – dirección que ve la unidad de memoria
- Las direcciones lógicas y físicas son las mismas en la vinculación (binding) de tiempo de compilación y carga; y difieren en la vinculación (binding) de tiempo de ejecución

# REGISTROS BASE Y LÍMITE

- Un par de registros **base** y **límite** definen el espacio lógico de direcciones

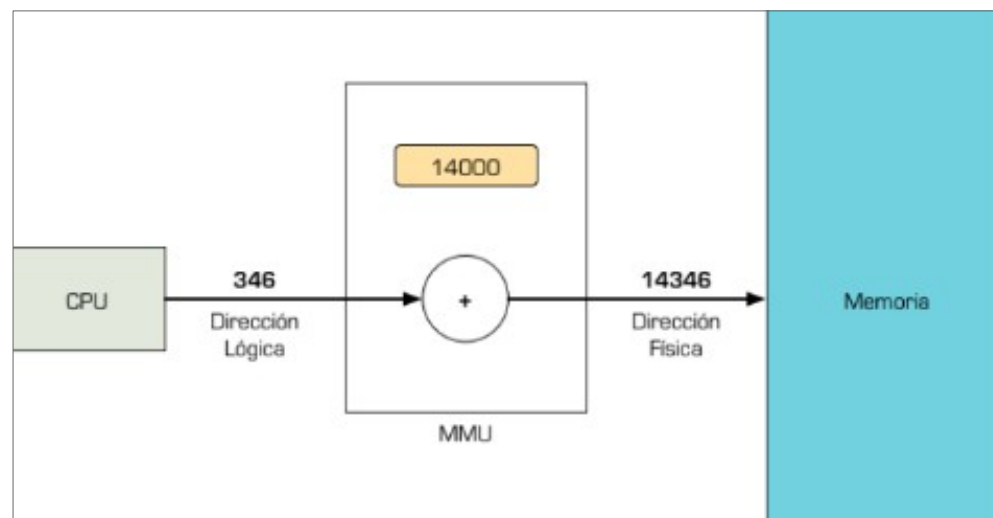


# SOPORTE DE HARDWARE



# UNIDAD DE ADMINISTRACIÓN DE MEMORIA (MMU)

- Dispositivo de hardware que mapea la memoria virtual o lógica con la memoria física
- En un esquema con MMU, el valor del registro de reubicación es **sumado** a cada dirección generada por un proceso del usuario cuando se envía a memoria
- Los programas del usuario trabajan con direcciones **lógicas**; nunca ven las direcciones **físicas** reales





# **ASIGNACIÓN CONTINUA**

# ASIGNACIÓN CONTINUA

- La memoria se divide generalmente en 2 particiones:
  - SO residente, generalmente en la parte baja de memoria junto al vector de interrupciones
  - Procesos del usuario
- Se utilizan registros de re-ubicación para proteger los procesos entre si, y al sistema operativo
  - **Registro base** – contiene el valor de la menor dirección física
  - **Registro límite** – contiene el rango de direcciones lógicas; las direcciones lógicas deben ser menor que este valor
- La MMU mapea direcciones lógicas en forma dinámica

# ASIGNACIÓN CONTINUA

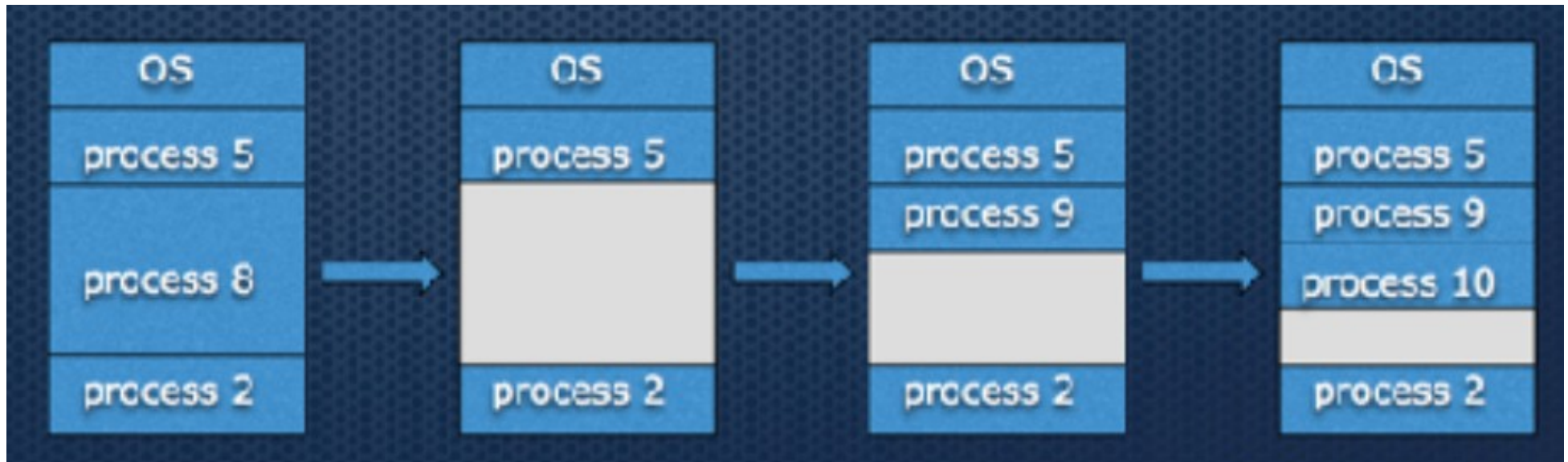
- Asignación con múltiples particiones
- **Hueco – bloque de memoria disponible;** de varios tamaños desparramados por memoria
- Cuando un proceso llega, se le asigna un bloque lo suficientemente grande.
- El SO mantiene información de:
  - a) particiones asignadas
  - b) particiones libres (huecos)

## The block list method



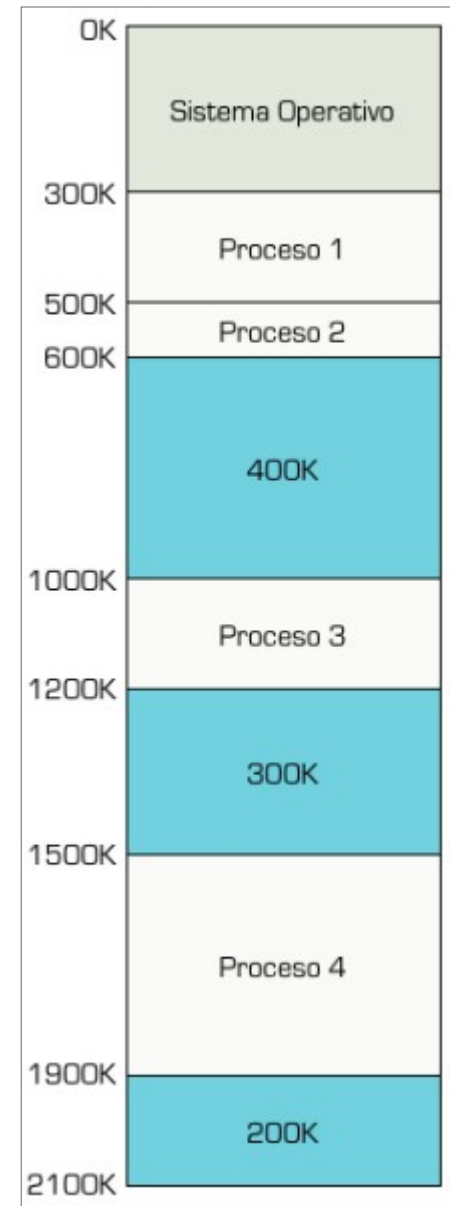


# ASIGNACIÓN CONTINUA



# PROBLEMA DE ASIGNACIÓN DINÁMICA

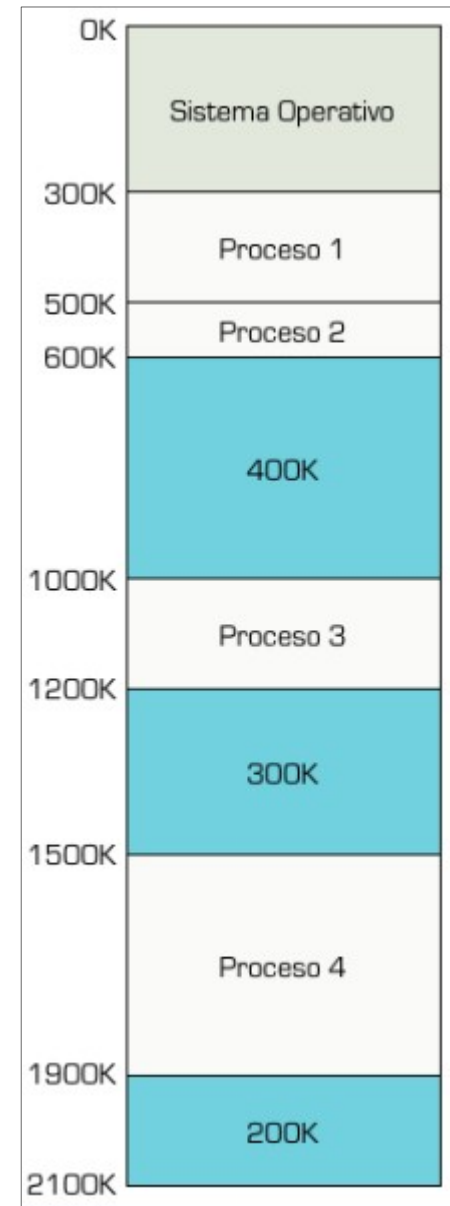
**Cómo satisfacer un pedido de tamaño  $n$  con una lista de huecos? (ej: 100K)**



# PROBLEMA DE ASIGNACIÓN DINÁMICA

Cómo satisfacer un pedido de tamaño  $n$  con una lista de huecos? (ej: 100K)

- **Primer ajuste (First-fit):** asigna el primer bloque libre que es suficientemente grande
  - Produce el desperdicio más chico por hueco
- **Mejor ajuste (Best-fit):** asigna el menor bloque que es suficientemente grande para contener al proceso; debe buscar en toda la lista, salvo que este ordenada
  - Produce el desperdicio más chico por hueco
- **Peor ajuste (Worst-fit):** asigna el bloque más grande; debe recorrer la lista completa
  - Deja el desperdicio más grande



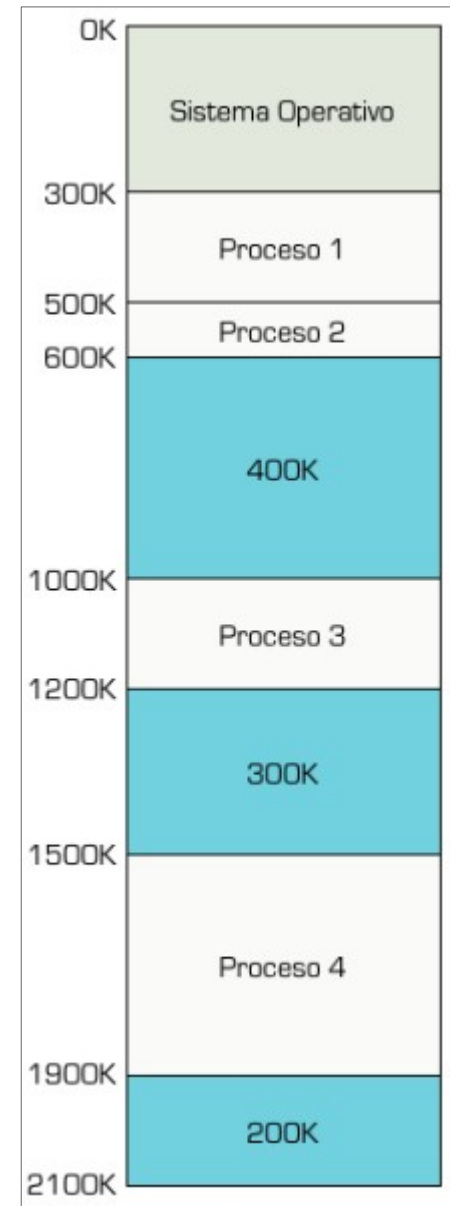
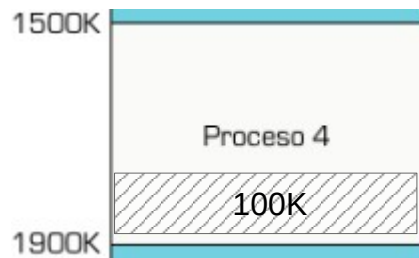
# FRAGMENTACIÓN

- **Fragmentación Externa**

existe suficiente para satisfacer un pedido, pero no en forma continua

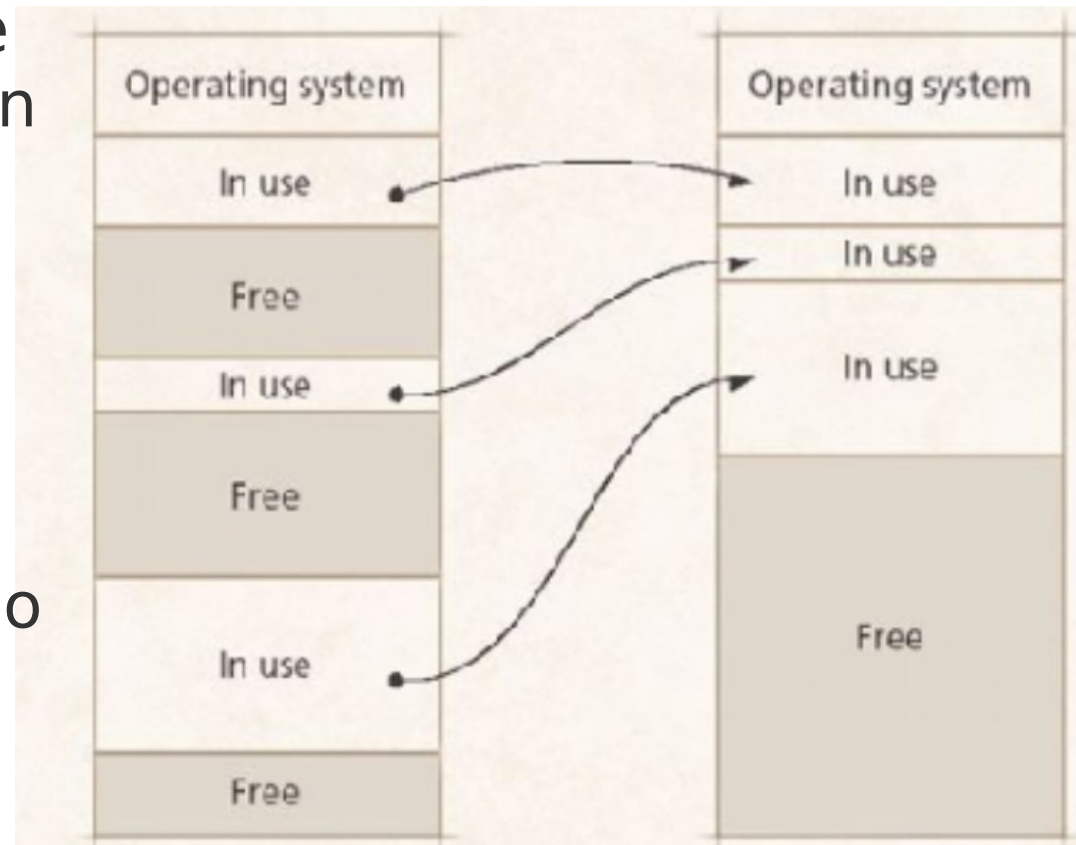
- **Fragmentación Interna**

la memoria asignada puede ser algo mayor que la requerida; esta diferencia es interna a una partición pero no es usada

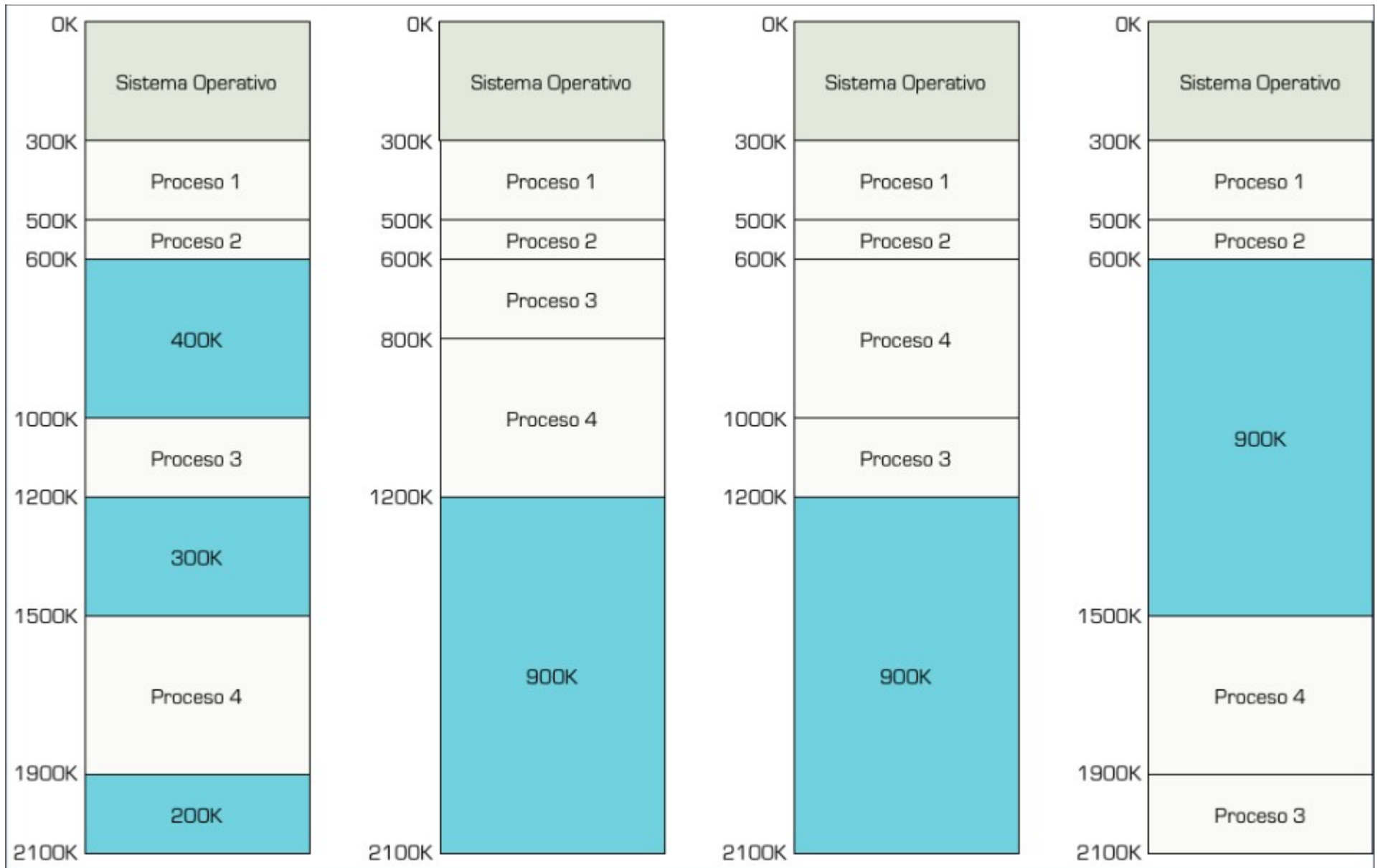


# COMPACTACIÓN

- Reduce la **fragmentación externa** por medio de la compactación
- Mueve el contenido de memoria para ubicar toda la memoria libre junta
- La compactación es posible sólo si tenemos re-ubicación dinámica, y el binding es hecho en tiempo de ejecución
- Problemas de I/O
  - Anclar el proceso en memoria si está haciendo E/S
  - Hacer E/S sólo a buffers del SO



# COMPACTACIÓN



# Asignación Continua

- + Soporta **múltiples programas** ejecutando a la vez
- + Los programas de usuario tienen un **espacio de memoria** definido (protección)
- Requiere traducción **Dirección Lógica → Dirección Física**
  - + MMU Hardware
- Genera **Fragmentación Externa**
- **Compactación**
  - + Reduce **fragmentación externa**
  - **Excesivamente costosa**