UNIVERSIDAD CATOLICA DEL MAULE

# Gestión de Memoria

Dr. Felipe Tirado

# Objetivos

- Distinguir entre espacios lógicos y físicos de memoria. Entender el proceso de traducción de direcciones.
- Entender el concepto de MMU (Unidad de manejo de memoria).
- Conocer técnicas elementales de intercambio, enlace dinámico, registro base y límite.
- Conocer la gestión de memoria contigua: partición única, particiones múltiples. El problema de la fragmentación y sus posibles soluciones.
- Conocer las técnicas de gestión del espacio libre.

UNIVERSIDAD CATOLICA DEL MAULE

# Objetivos

- Conocer el concepto de Paginación.
- Identificar la conveniencia de usar sistemas paginados de dos o más niveles.
- Conocer las técnicas para compartir memoria o protegerla usando segmentación y paginación.
- Conocer los algoritmos de sustitución. Cómo medir el rendimiento de un algoritmo FIFO, óptimo, LRU, NRU, LFU.

UNIVERSIDAD CATOLICA DEL MAULE

# Propósito Gestión Memoria

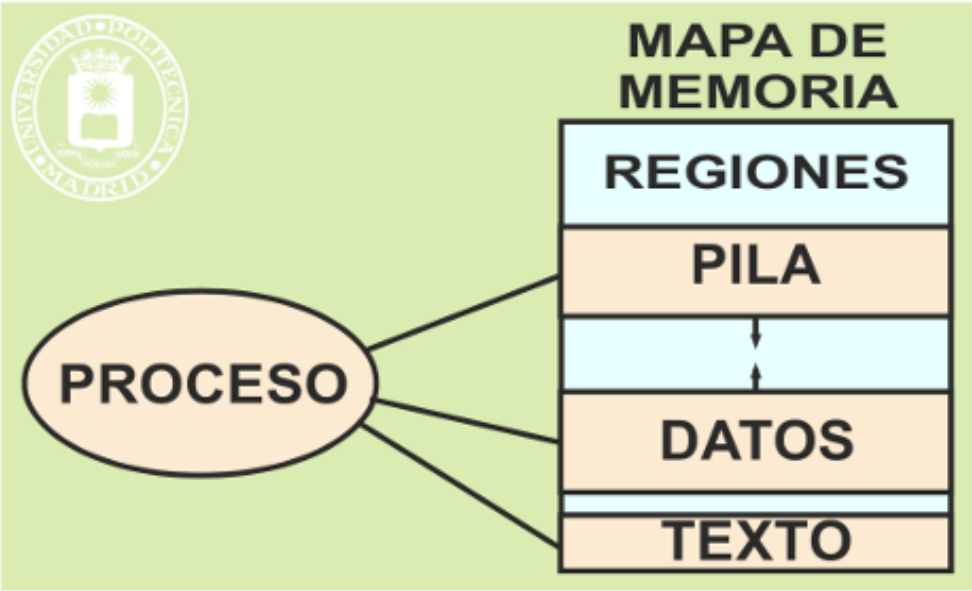
- La memoria es el recurso donde residen el código y los datos de los procesos que se ejecutan en nuestro equipo.

- La memoria es un recurso limitado, y más encima considerado escaso
  - El sistema operativo debe repartirlo entre los diferentes procesos que solicitan ejecutarse, asegurando algún nivel de protección entre las zonas ocupadas por los diferentes programas.

UNIVERSIDAD CATOLICA DEL MAULE

# Misión del sistema operativo con respecto a la memoria

- Todo proceso requiere de una imagen de memoria
  - Código del programa
  - Datos que necesita el programa
  - Pila
- Esta imagen está formada por regiones




UNIVERSIDAD CATOLICA DEL MAULE

# Misión del sistema operativo con respecto a la memoria

- En un entorno multiprogramado, pueden existir varios programas ejecutándose al mismo tiempo.

- Esto obliga a gestionar el espacio disponible de memoria.
  - Como la capacidad de la memoria es infinita, puede ocurrir que en un momento dado no se pueden ejecutar todos los programas que los usuarios pretendar lanzar.

UNIVERSIDAD CATOLICA DEL MAULE

# Misión del sistema operativo con respecto a la memoria

- Hay que conocer cuales son las zonas de memoria ya ocupadas, para no sobreescribirlas de forma accidental.
  - El responsable de esta administración es el sistema operativo.

- Cuando un programa se carga, el sistema operativo debe reservarle una zona de memoria libre.

UNIVERSIDAD CATOLICA DEL MAULE

# Protección de la memoria

Si tenemos varios procesos de forma simultanea en la memoria, algunos de ellos pueden acceder (lectura o escritura) a direcciones de memoria asignadas a otros procesos. Esto puede ser una verdadera fuente de problemas.

UNIVERSIDAD CATOLICA DEL MAULE

# Protección de la memoria

- Proteger al SO de los procesos activos
- Proteger unos procesos frente a otros
- Garantizar que los accesos a memoria son correctos
- Validar cada dirección generada por los procesos
- Hardware genera excepciones de acceso inválido y el SO hará el tratamiento del error

UNIVERSIDAD CATOLICA DEL MAULE

# Resumen: Objetivos de la gestión de memoria

- Repartir la memoria disponible entre los procesos que la solicitan.
- Intentar que este reparto sea lo más equilibrado posible.
- Llevar el control de qué zonas de memoria están libres y cuáles están ocupadas.
- Proteger las zonas de memoria de accesos indebidos.

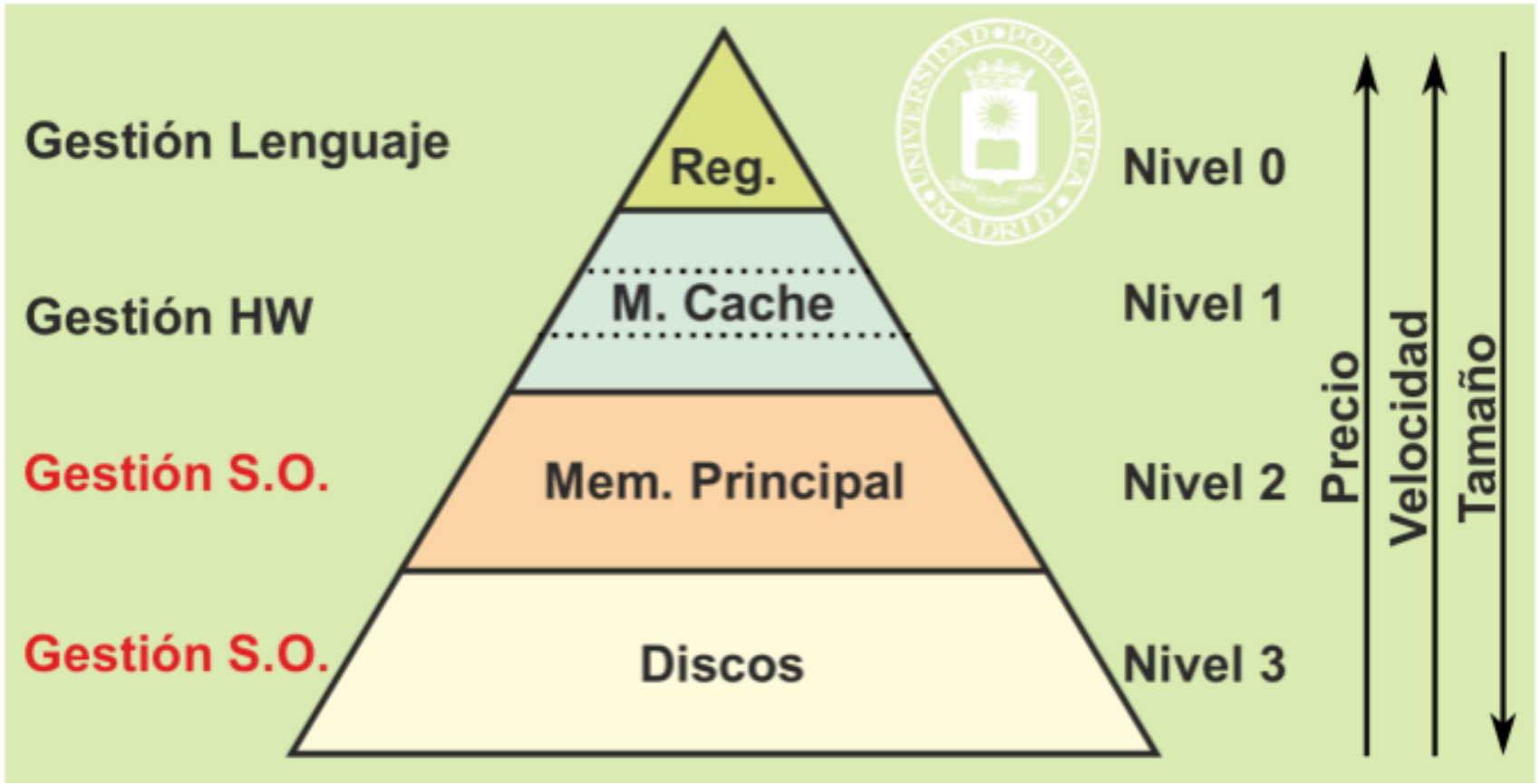
UNIVERSIDAD CATOLICA DEL MAULE

# Resumen: Objetivos de la gestión de memoria

- Lograr que el espacio desaprovechado sea el mínimo posible.
- Aprovechar toda la jerarquía de memorias, incluso los discos (memoria virtual).
- Resolver el “salto semántico” que hay entre la memoria física (tal y cómo está estructurada) y la memoria lógica (tal y como la ve el usuario).

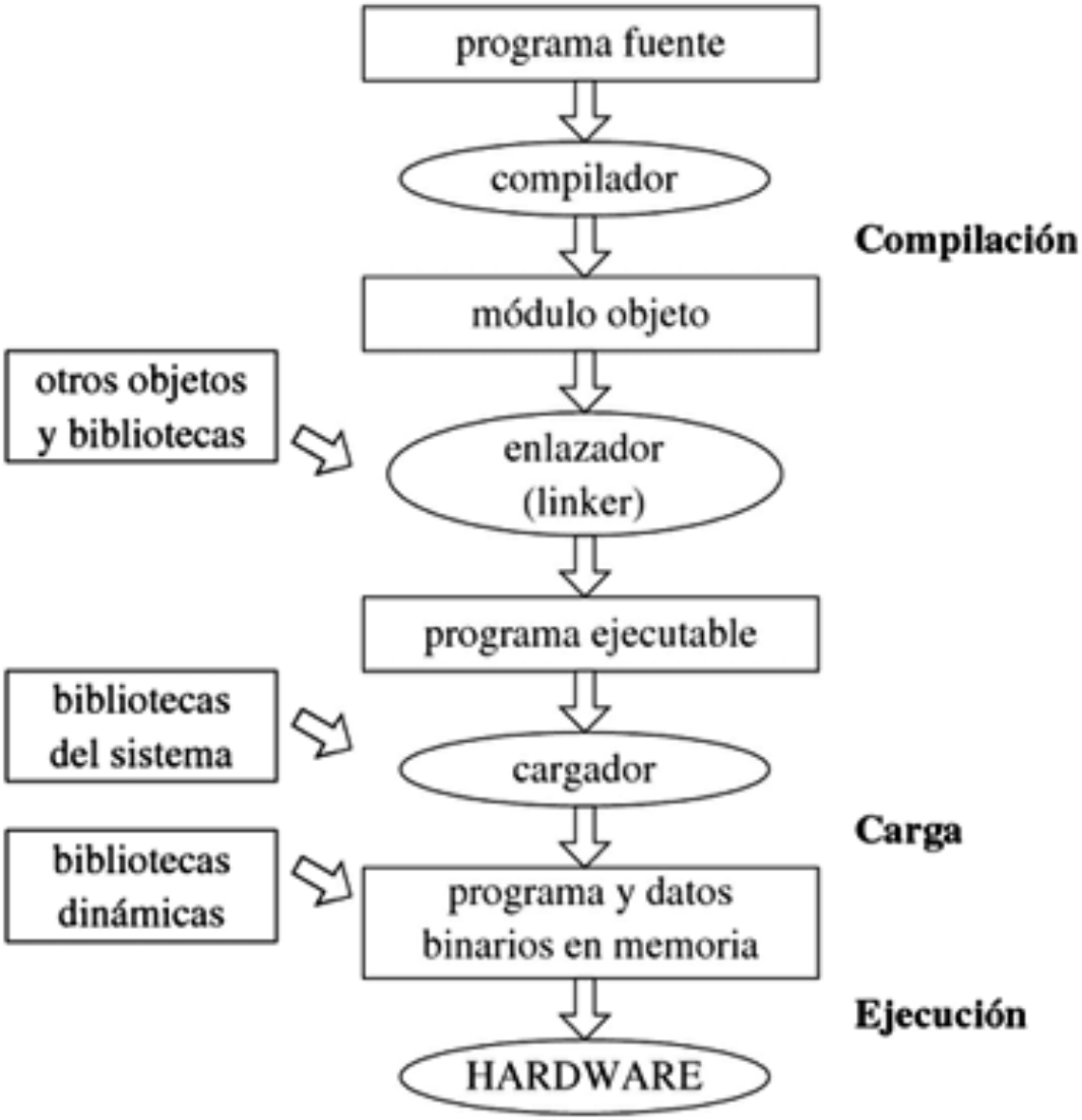
UNIVERSIDAD CATOLICA DEL MAULE

# Jerarquía de memoria




UNIVERSIDAD CATOLICA DEL MAULE

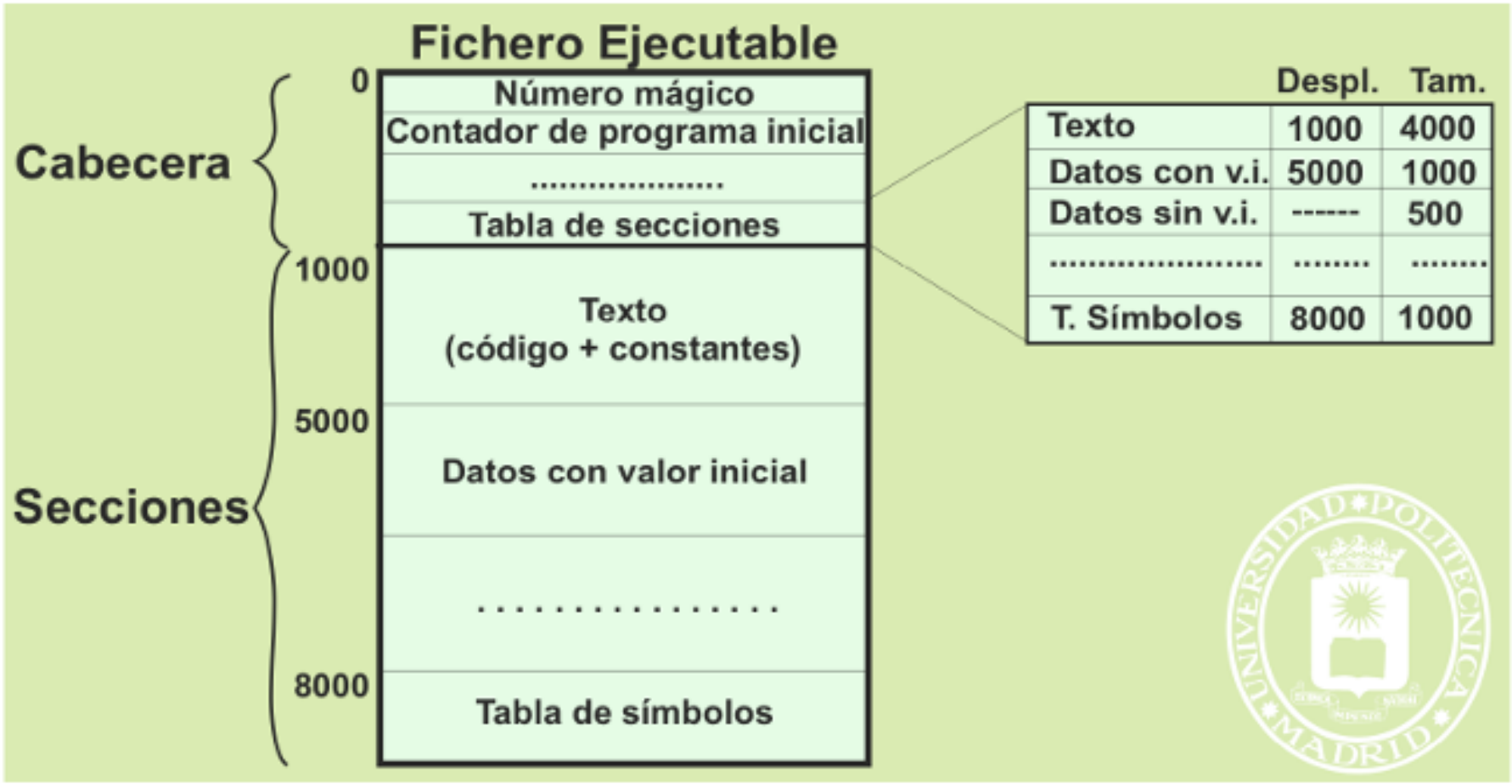
# Ciclo de vida de un programa




- **Compilación:** El compilador traduce las direcciones simbólicas, a direcciones numéricas.
- **Enlace (Link):** Los archivos objeto tienen que combinarse en las bibliotecas del lenguaje.
- **Carga (load):** El sistema operativo debe reservarse un espacio de memoria y copiar el código y datos iniciales, así como configurar y lanzar el nuevo proceso.
- **Ejecución (run):** Una vez que el programa se ejecuta, estará continuamente realizando accesos a memoria.

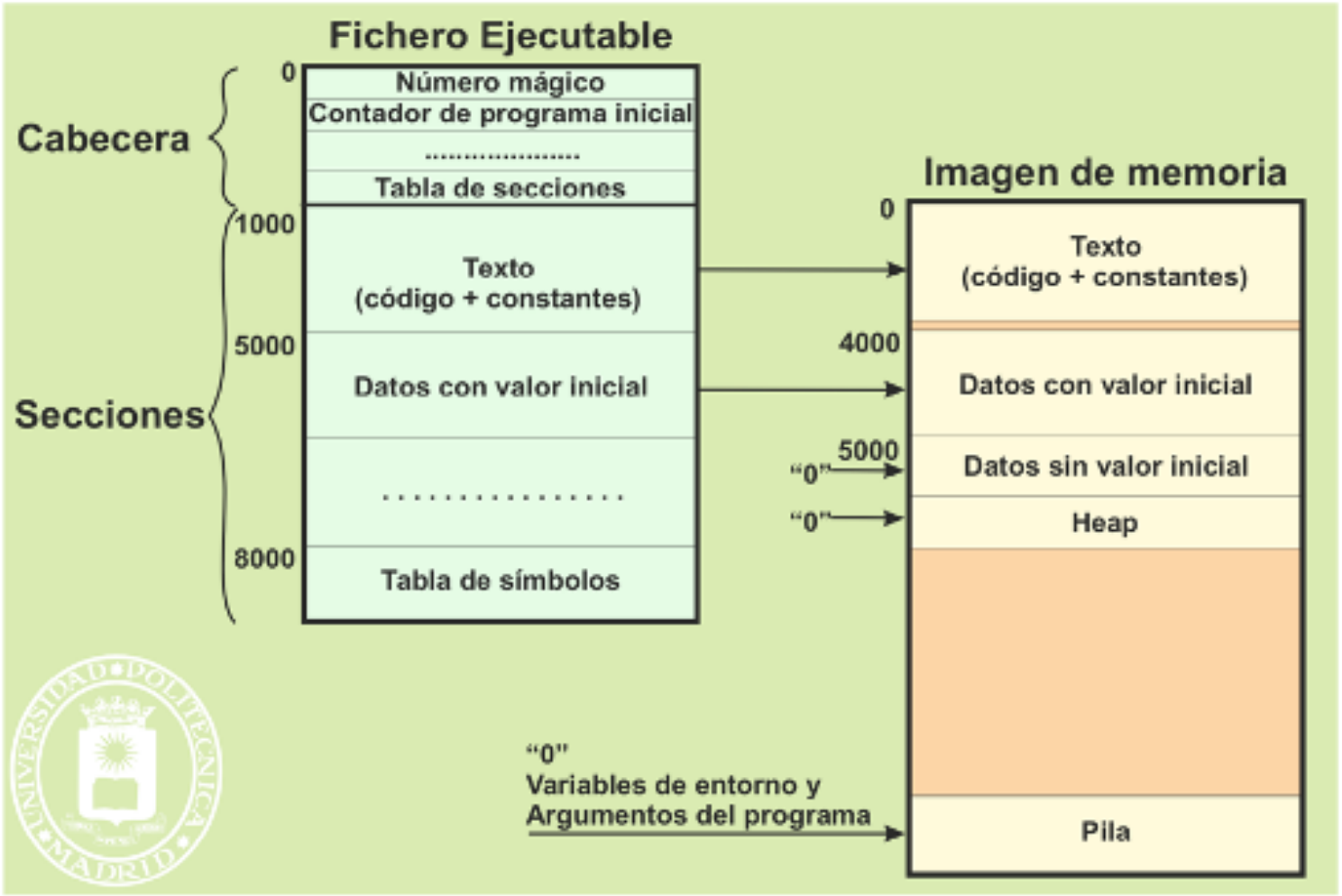
UNIVERSIDAD CATOLICA DEL MAULE

# Formato del archivo ejecutable




UNIVERSIDAD CATOLICA DEL MAULE

# Creación del mapa del ejecutable




UNIVERSIDAD CATOLICA DEL MAULE

# Traducción de direcciones: Espacios lógicos y físicos

- **Las direcciones lógicas** son aquellas a las que accede el programa, tal y como se generó el archivo ejecutable. En muchos cosas, se trata de direcciones contiguas numeradas de cero en adelante.

- **Las direcciones físicas** son aquellas donde se ha ubicado realmente el programa en la memoria principal. A cada dirección le corresponderá una dirección física, que no tienen por qué coincidir.

UNIVERSIDAD CATOLICA DEL MAULE

# Traducción de direcciones: Espacios lógicos y físicos

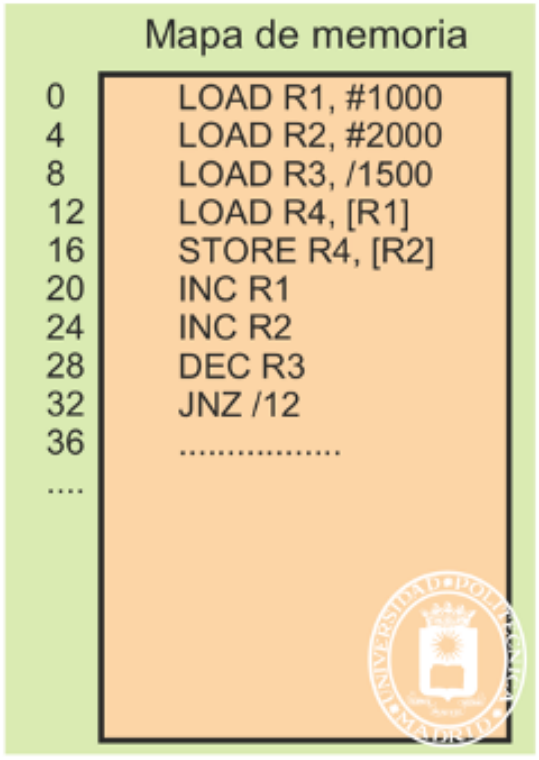
```
LOAD (120), R1
ADD R1,1
STORE R1, (200)
JSR 240
```

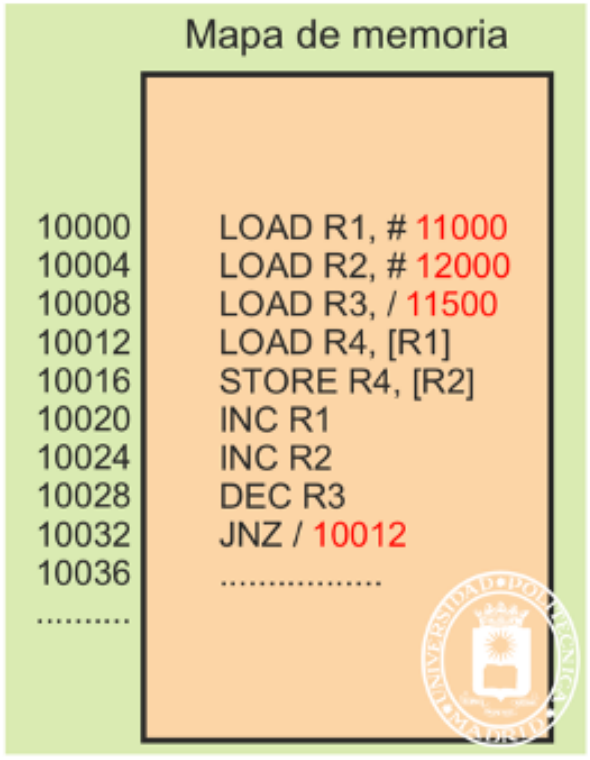
Si el programa se ha cargado a partir de la celda 550 de memoria principal. La dirección lógica 120 corresponderá con la dirección física 670.

UNIVERSIDAD CATOLICA DEL MAULE

# Reubicación estática y dinámica

**Reubicación estática,** la asignación de las direcciones de memoria físicas se realiza antes de la ejecución del programa. Esta estrategia asume que el programa ocupará una dirección de memoria específica durante toda su ejecución, y el código se genera con direcciones absolutas en la fase de compilación o al momento de cargar el programa.

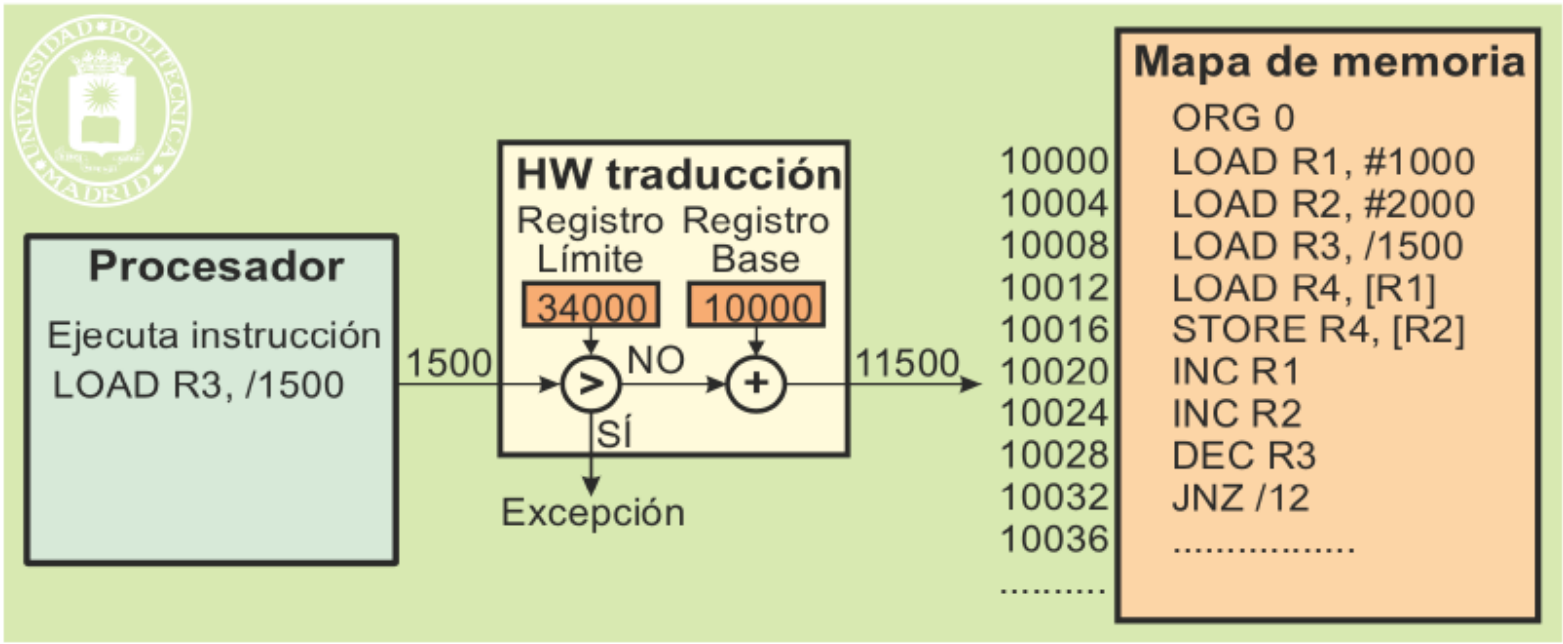



UNIVERSIDAD CATOLICA DEL MAULE

# Reubicación estática y dinámica

La **reubicación dinámica** es más flexible que la estática. En este enfoque, las direcciones que el programa utiliza (direcciones lógicas o virtuales) son convertidas dinámicamente a direcciones físicas durante la ejecución del programa. Esto lo hace el sistema operativo utilizando hardware específico, como la **Unidad de Gestión de Memoria** (**MMU**, Memory Management Unit).




UNIVERSIDAD CATOLICA DEL MAULE

# Gestión de memoria contigua

UNIVERSIDAD CATOLICA DEL MAULE

# Intercambio(Swapping)

- Los primeros sistemas multiprogramados tenían muy poca memoria y además carecían de la MMU más básicas.
  - Por ese motivo no había opción de mantener varios procesos en memoria principal al mismo tiempo.

- *¿Como se puede construir un sistema multiprogramado en el que solo hay un proceso en memoria principal?*
  - Invento técnica llamada intercambio.

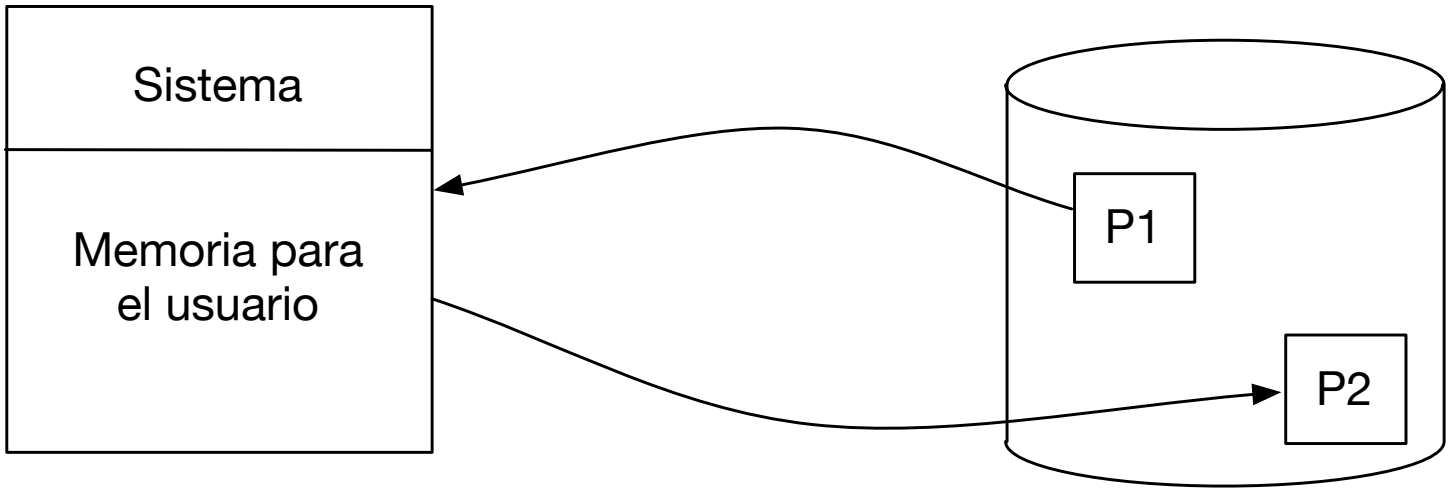
UNIVERSIDAD CATOLICA DEL MAULE

# Intercambio

- Cuando el proceso actualmente en ejecución se queda bloqueado, se entrega el control a otro proceso preparado para ejecutarse.
- Para no perder la información del proceso bloqueado, su código y sus datos se guardan en el disco (*swap out*).
- El código y los datos del segundo proceso se copian en la memoria principal desde el disco (*swap in*).
- El intercambio consiste en utilizar el disco como almacén secundario del estado de los procesos que se encuentran en espera.

UNIVERSIDAD CATOLICA DEL MAULE

# Memoria con intercambio o swapping




UNIVERSIDAD
CATOLICA
DEL MAULE

# Intercambio (problema)

- El principal problema de esta técnica es que realiza mucha transferencia entre el disco y la memoria lo que ralentiza la ejecución de los procesos.

- **Mejora:** Utilizar un doble buffer que permita cargar/ descargar un proceso mientras se esta ejecutando otro.

UNIVERSIDAD CATOLICA DEL MAULE

# Particiones múltiples

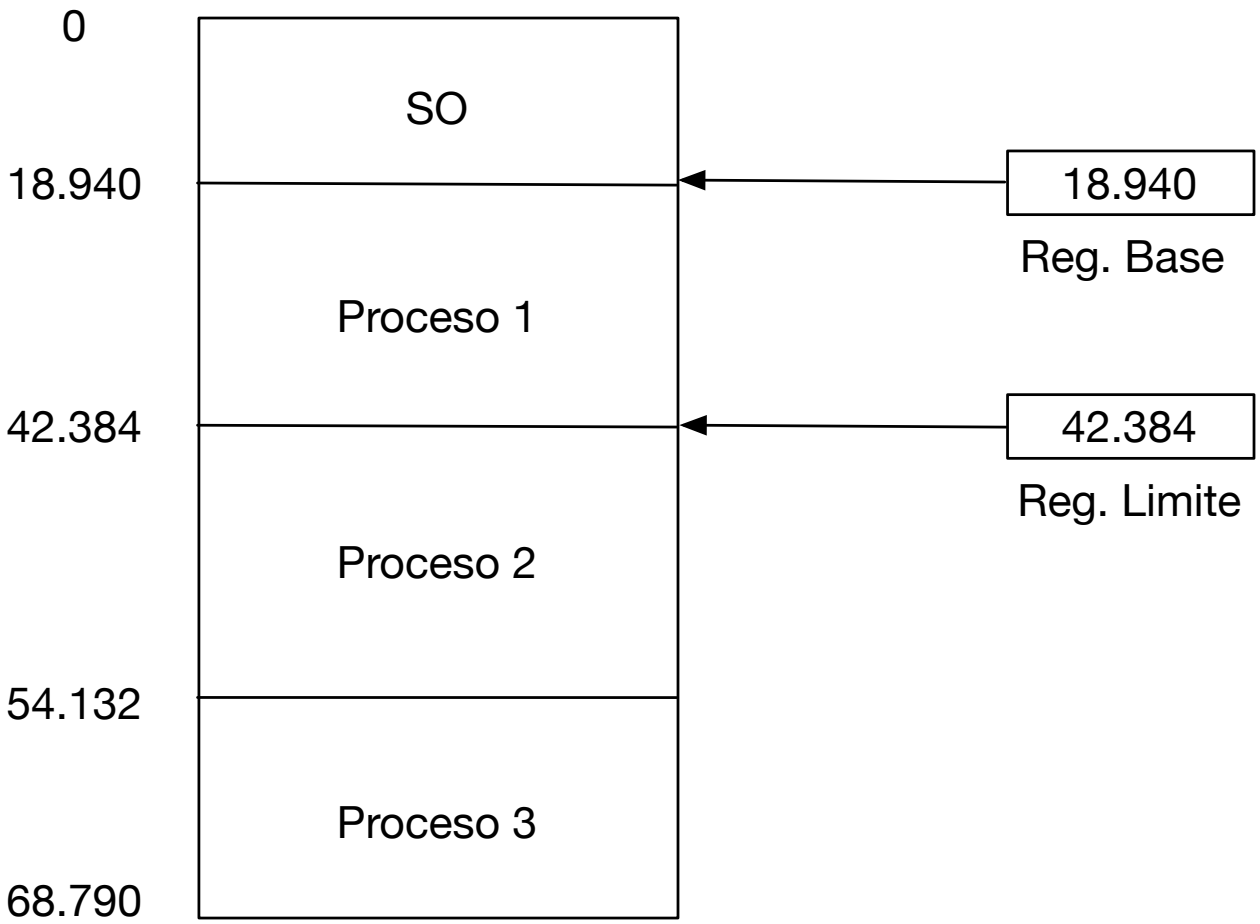
Una solución mas avanzada consiste en permitir que residan varios programas en la memoria principal.

- SO concederá a cada proceso una zona de memoria contigua (partición) en la que reside su código y sus datos.

UNIVERSIDAD CATOLICA DEL MAULE

# Particiones múltiples

Memoria con múltiples particiones




UNIVERSIDAD CATOLICA DEL MAULE

# Particiones fijas y variables

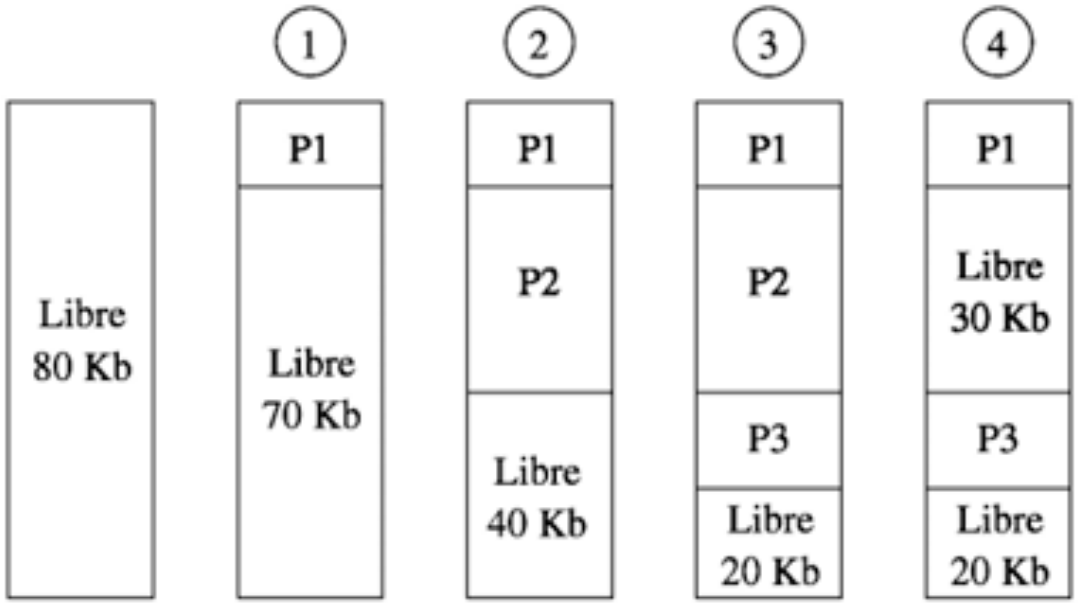
- En la antigüedad, las zonas de memoria tenían un tamaño predefinido (particiones fijas).
  - Gestionarla era más sencilla.
  - Desperdiciaba espacio de memoria

- Posteriormente se paso a particiones variables, en los que las zonas ocupada por un programa podía ser de cualquier tamaño.
  - Gestionarla mas compleja.
  - Aprovechamiento mejor de memoria
  - Generan huecos libres de memoria entre programas.

UNIVERSIDAD CATOLICA DEL MAULE

# Fragmentación

- **Ejemplo**
  - Dispone de 80 Kb de memoria libre
    - Llega un programa P1 que necesita 10 Kb.
    - Llega un programa P2 que necesita 30 Kb.
    - Llega un programa P3 que necesita 20 Kb.
    - Termina P2
    - Llega un programa P4 que necesita 40 Kb.



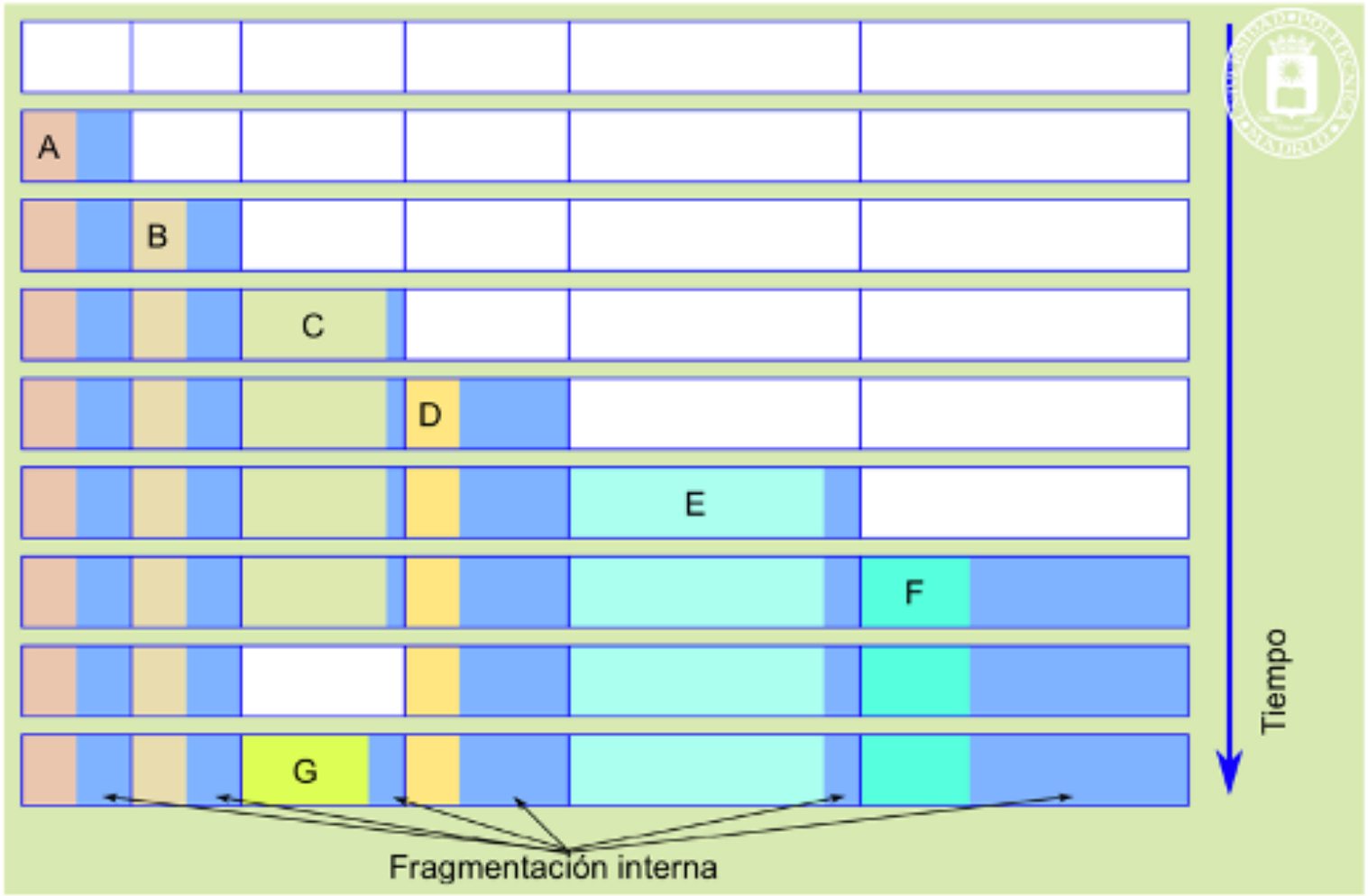

UNIVERSIDAD CATOLICA DEL MAULE

# Fragmentación

- Tipos
  - **Fragmentación externa.**
    - Zonas no ocupadas están marcada como libre por el sistema operativo.
  - **Fragmentación interna.**
    - Zonas no aprovechadas, pero están marcada como ocupadas.
    - Ocurre en particiones fijas de 10 K para acomodar una demanda real de 8 K, queda 2 K inutilizables.

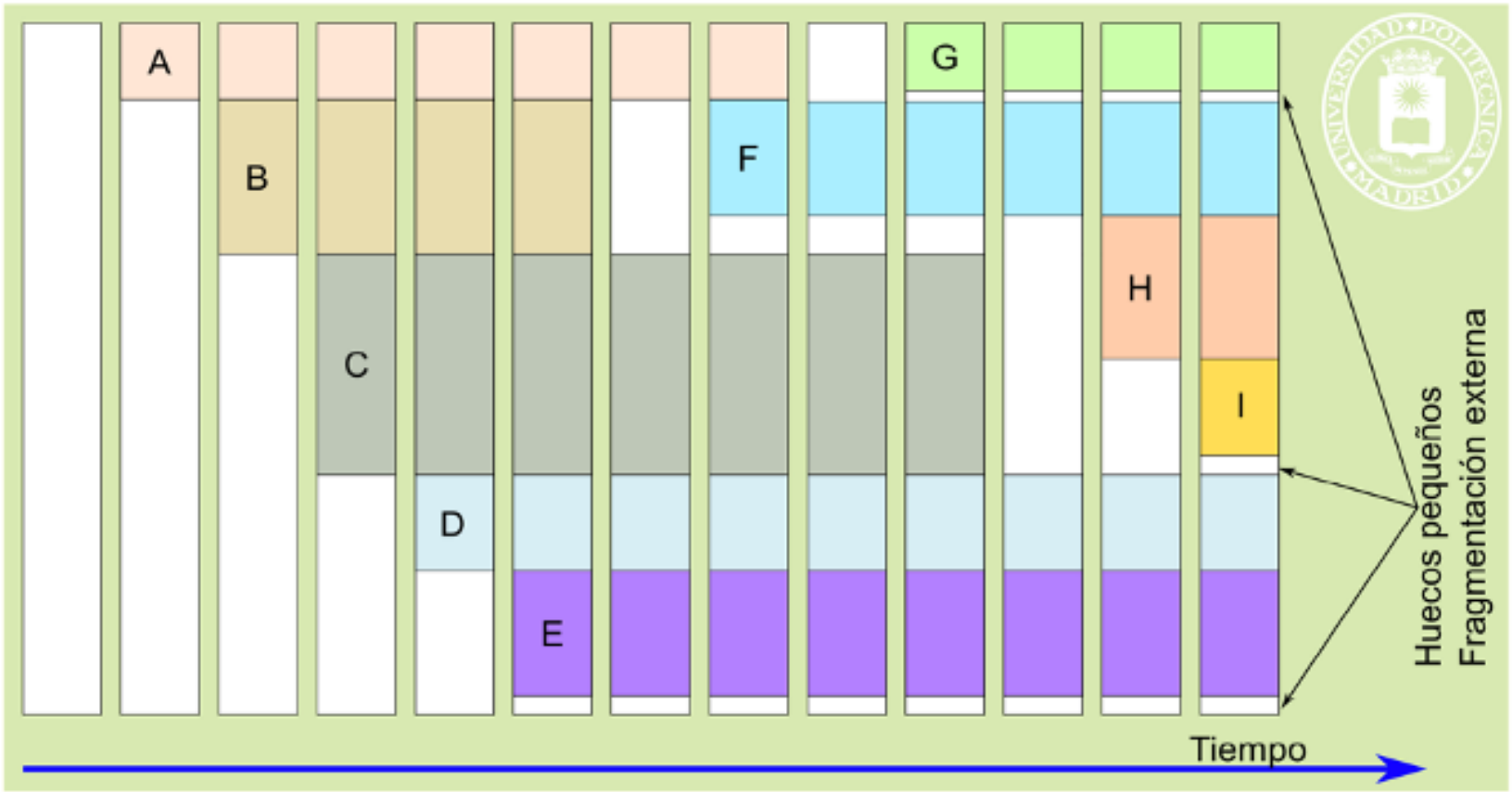
UNIVERSIDAD CATOLICA DEL MAULE

# Particiones fijas y fragmentación interna




UNIVERSIDAD CATOLICA DEL MAULE

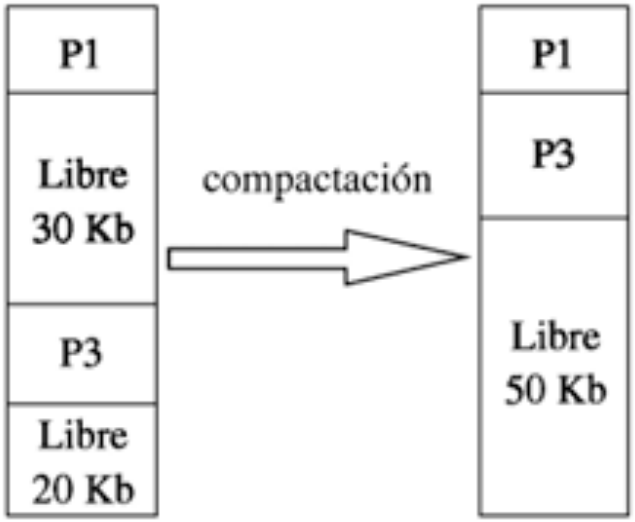
# Particiones variables y fragmentación externa




UNIVERSIDAD CATOLICA DEL MAULE

# Fragmentación

- **Una solución**
  - Compactación de bloques.
    - Que tenga como resultado final de un único bloque libre.
    - **Problema**
      - Consume bastante tiempo.
      - Puede ocurrir que el programa que ocupa ese bloque ya no puede funcionar.
        - » Direcciones que esta trabajando a cambiado de lugar.




UNIVERSIDAD CATOLICA DEL MAULE

# Gestión del espacio libre

*¿ Como sabe el sistema operativo que zona de memoria están libres y ocupadas ?.*

- Necesita una estructura en la que sea anote que proceso se encuentra en cada partición, donde comienza y termina esta, etc.

- Para gestionar el espacio libre se utiliza una lista de *huecos libre* implementada por lista enlazadas.

UNIVERSIDAD CATOLICA DEL MAULE

# Gestión del espacio libre

*Ante una solicitud de espacio de memoria, ¿qué algoritmo se usan para escoger el mejor hueco?*

- **Primer ajuste (first fit)**, devuelve el primer hueco que se encuentre que sea mayor o igual que el tamaño solicitado.

- **Mejor ajuste (best fit)**, devuelve el hueco libre más pequeño que satisfaga la petición.

- **Peor ajuste (worst fit)**, devuelve siempre el hueco libre mas grande para no generar huecos pequeños.

UNIVERSIDAD CATOLICA DEL MAULE

# Biblioteca de enlace dinámico

- Supongamos que un sistema multiprogramado cada una de las aplicaciones ejecutadas consume un espacio en la memoria principal.

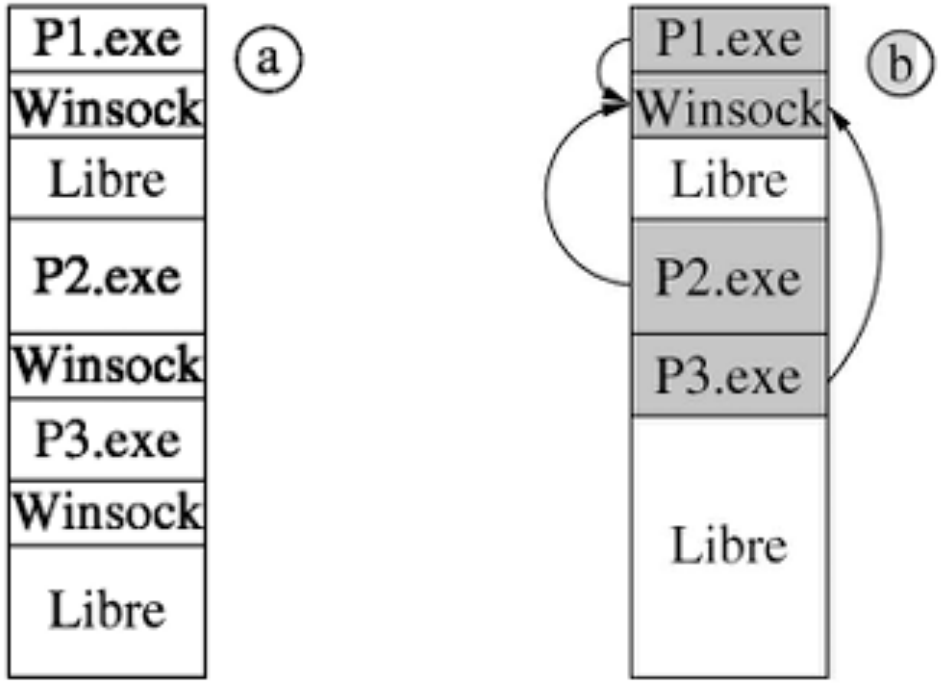
- Podemos observar que muchos de estos programas usan el mismo código.
  - Todos los programas que trabajan en la red usan unas subrutinas llamadas "W*insock*".
  - Muchos juegos usan una biblioteca llamada "**DirectX**".
  - Etc.

- Ocurrirá que este módulo estará copiado en memoria varias veces.
  - Desperdicio de memoria

UNIVERSIDAD CATOLICA DEL MAULE

# Biblioteca de enlace dinámico

Memoria sin DLL(a)  Memoria usando DLL (b)




UNIVERSIDAD CATOLICA DEL MAULE

# Biblioteca de enlace dinámico

- La gestión de DLL complica la gestión del espacio de memoria.
  - Programa ya no consiste en un único bloque.

- La segmentación es una técnica que facilita la implementación de DLL.

- Además el SO debe encargarse de retirar de memoria las DLL que nadie usa.

UNIVERSIDAD CATOLICA DEL MAULE

# Segmento

- Un segmento de memoria es una unidad lógica que agrupa las diferentes secciones de un programa en áreas de memoria separadas según su funcionalidad. Cada segmento tiene un propósito específico y puede ser asignado en una parte diferente de la memoria física.

- Propósito lógico: Cada segmento corresponde a una parte lógica del programa. Por ejemplo:
  - **Segmento de código:** Contiene las instrucciones ejecutables del programa.
  - **Segmento de datos:** Contiene las variables globales y los datos con valores iniciales.
  - **Segmento de pila:** Se utiliza para almacenar las variables locales y las direcciones de retorno durante las llamadas a funciones.
  - **Segmento de heap:** Almacena la memoria asignada dinámicamente.

UNIVERSIDAD CATOLICA DEL MAULE

# Segmentación

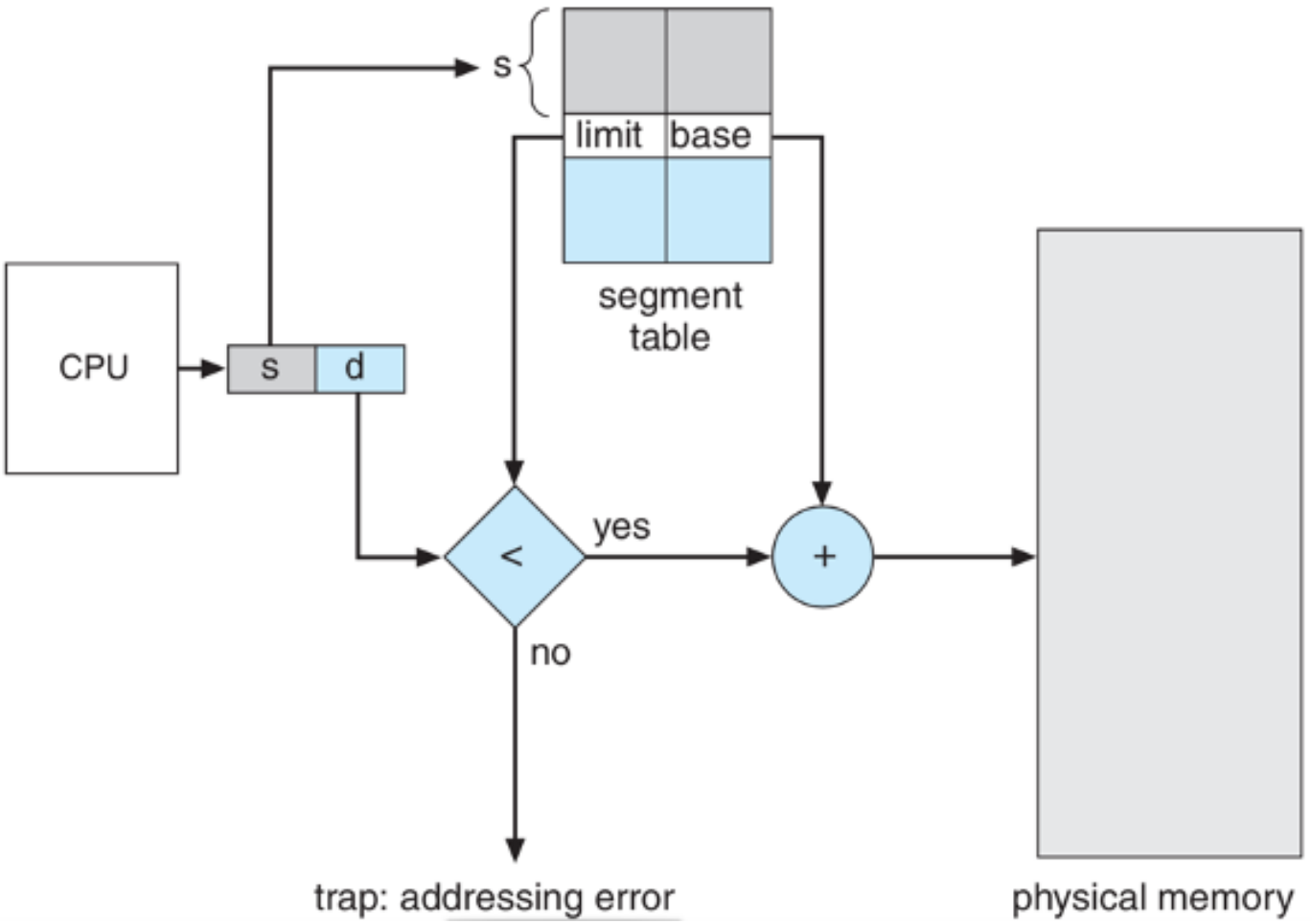
- Mapeo a dirección física se realiza con una **tabla de segmentos**

- Cada segmento tiene
  - Base del segmento: dirección inicial del segmento en memoria física
  - Límite: Tamaño del segmento

- Dirección lógica:
  - Número de segmento
  - Desplazamiento dentro del segmento (entre 0 y el límite)

UNIVERSIDAD CATOLICA DEL MAULE

# Segmentación




UNIVERSIDAD CATOLICA DEL MAULE

# Traducción de direcciones

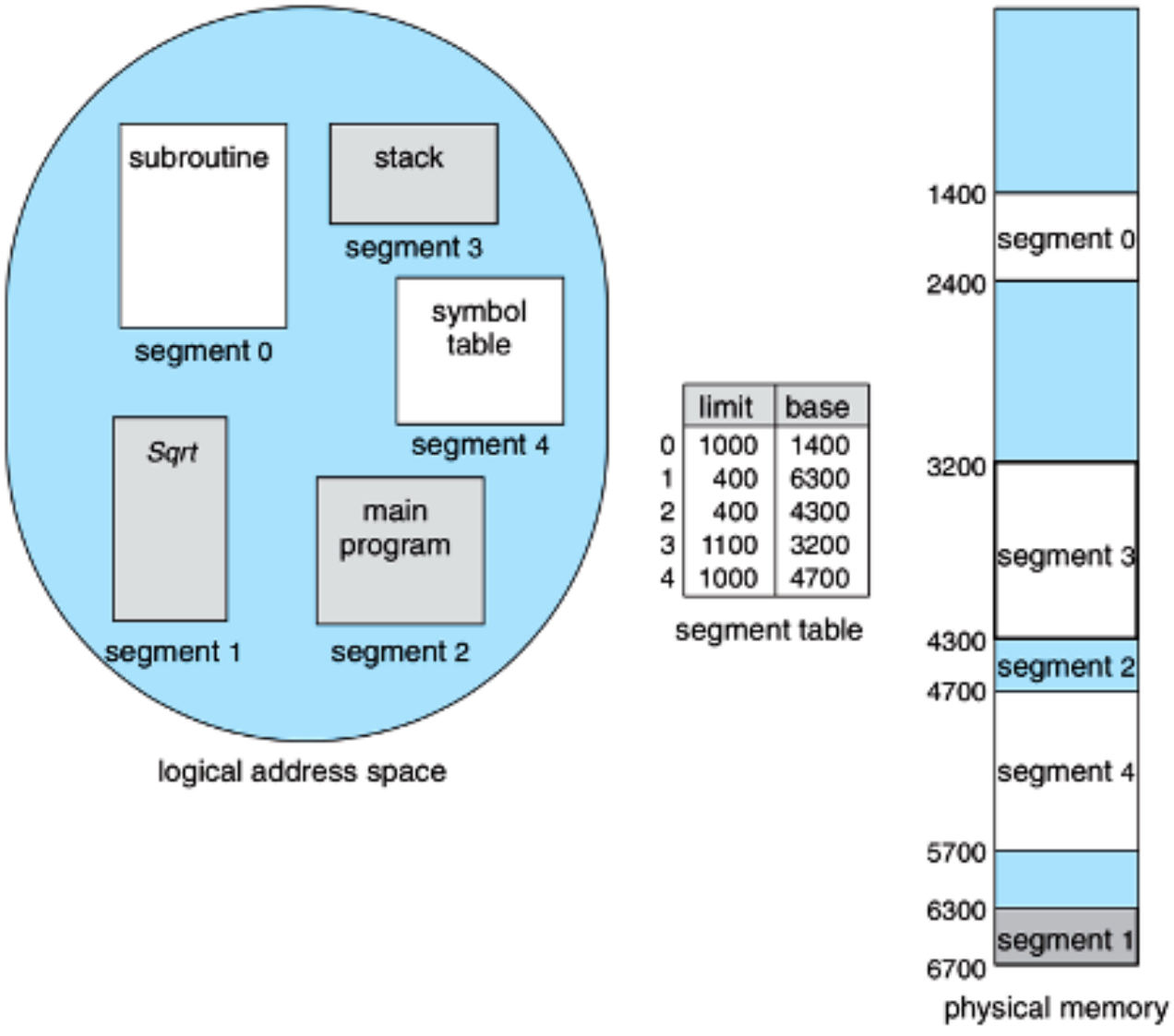
**Ejemplo**

Se genera la dirección (3:180), segmento 3 y desplazamiento 180.

- Buscar en la tabla de segmento la fila 3
- Tomar dirección base (500) y la dirección limite (250)
- Comprobar que el desplazamiento es menor que la dirección limite.
- Sumas la dirección base con el desplazamiento 500+180 =680
- Accede a la dirección física 680

UNIVERSIDAD CATOLICA DEL MAULE

# Ejemplo segmentación




UNIVERSIDAD CATOLICA DEL MAULE

# Paginación

- Técnica de paginación para librarse del problema de fragmentación.
- Un esquema de paginación implica que el mapa de memoria de cada proceso se considera dividido en páginas.
- A su vez, la MP del sistema se considera dividida en zonas del mismo tamaño que se denominan marcos de pagina.
- Un marco de pagina contendrá en un determinado instante una página de memoria de un proceso.

UNIVERSIDAD CATOLICA DEL MAULE

# Paginación

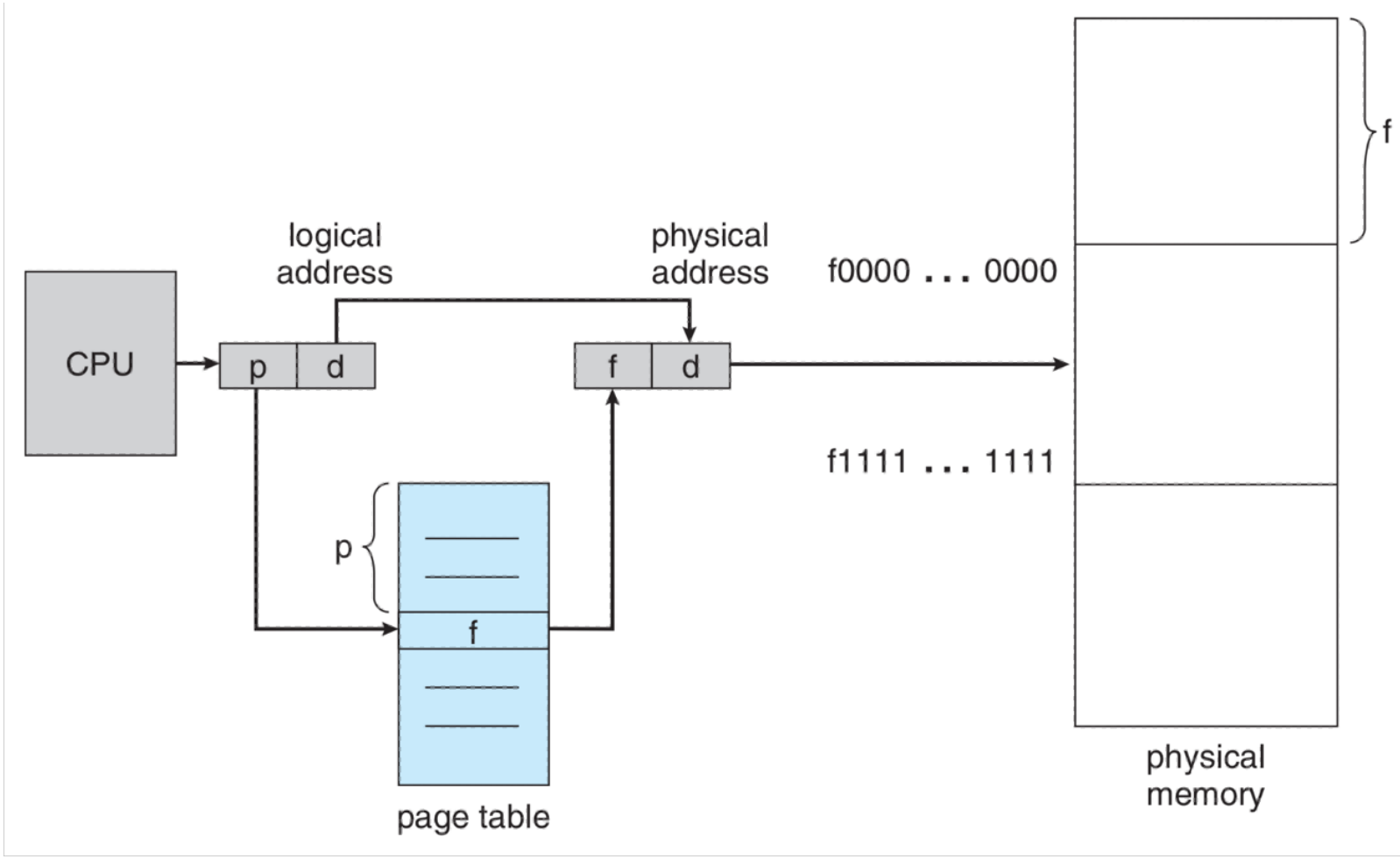
- El tamaño de las paginas varían según el procesador empleado.
  - Entre 1Kb y 32Kb, Intel usa tamaños de 4Kb

- La estructura de datos que relaciona cada página con el marco que está almacenada es la tabla de páginas.

- El hardware de gestión de memoria (MMU, Memory Management Unit) usa esta tabla para traducir todas las direcciones que genera un programa.

UNIVERSIDAD CATOLICA DEL MAULE

# Esquema de la paginación



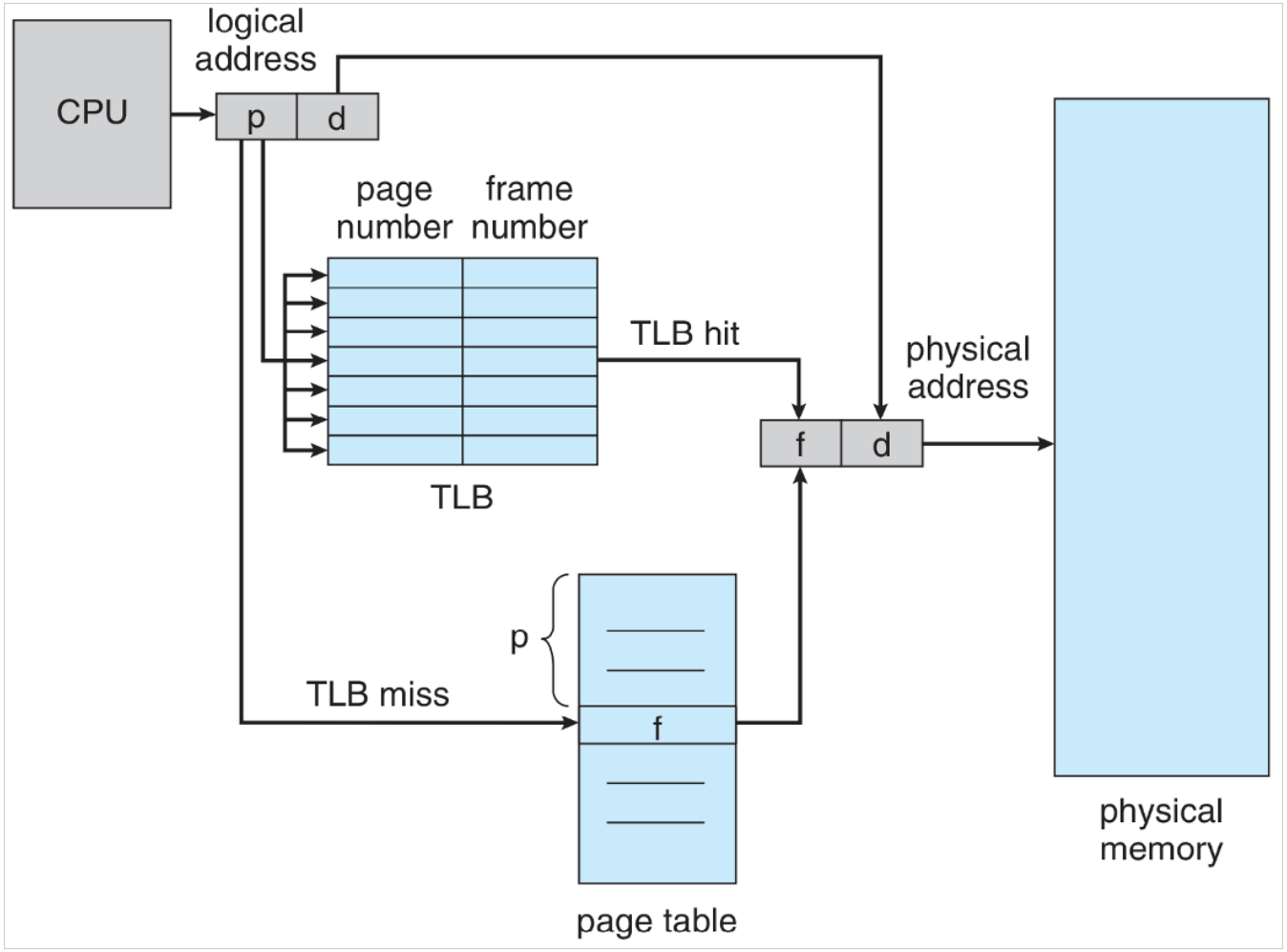

UNIVERSIDAD CATOLICA DEL MAULE

# Translation lookaside buffer

- Debido que se hace acceso a la memoria lógica se realizan dos accesos a memoria fisica
  - Tabla de de pagina.
  - Celda que se requiere.
- La MMU guarda un subconjunto de entrada a tablas de pagínas.
  - En memoria TLB (Translation lookaside buffer).
    - TLB hit, con pocos registros (32 o 64) 98%
    - TLB miss

UNIVERSIDAD CATOLICA DEL MAULE

# Uso de TLB para acceder a páginas




UNIVERSIDAD CATOLICA DEL MAULE

# TLB

- Ejemplo Práctico
  - Supongamos que el tiempo de acceso a memoria principal es de 40nseg, que el tiempo de acceso a la TLB es de 2 nseg y que la tasa de acierto de la TLB es de 98%
  - Tiempo medio de acceso a memoria será.

# TLB

=98% * (2 nseg +40nseg) + 2% * (2 nseg +40 nseg+40 nseg)

=42.8 nseg

Gracias a TLB, el hecho de usar memoria paginada solo provoca un aumento medio de:

=2.8 / 40

=7%

En el tiempo de acceso a la memoria.

UNIVERSIDAD CATOLICA DEL MAULE

# Papel sistema operativo

- Básicamente en asignar y liberar los marcos de página, así como rellenar correctamente las tablas de páginas.

- Calcular los marcos necesarios para cargar un nuevo proceso, los reserva en la MP, crea una tabla de página para el proceso y rellena dicha tabla con los numero de marco que ha escogido.

UNIVERSIDAD CATOLICA DEL MAULE

# Protección de páginas

- La paginación garantiza que un solo proceso pueda acceder a su espacio de memoria.

- Riego de protección, debido a que un proceso intente acceder a una pagina lógica mas allá de su espacio lógico.
  - Registro Limite, cuantas paginas lógicas tiene un proceso

UNIVERSIDAD CATOLICA DEL MAULE

# Paginación de varios niveles

- Programa es muy grande, su tabla de pagina puede tener un tamaño difícil de manejar.

- Ejemplo
  - Tamaño de pagina es de 1Kb, un programa cuyo tamaño es de 3Mb necesita 3.072 páginas, una tabla de pagina de 3.072 entradas, suponemos que cada entrada en la tabla ocupa 4 byte, tabla de paginación 12.288 byte, que han de estar contiguo en la MF para que la MMU pueda realizar correctamente la traducción de direcciones.
    - Fragmentación, problema que trataba de evitar la paginación

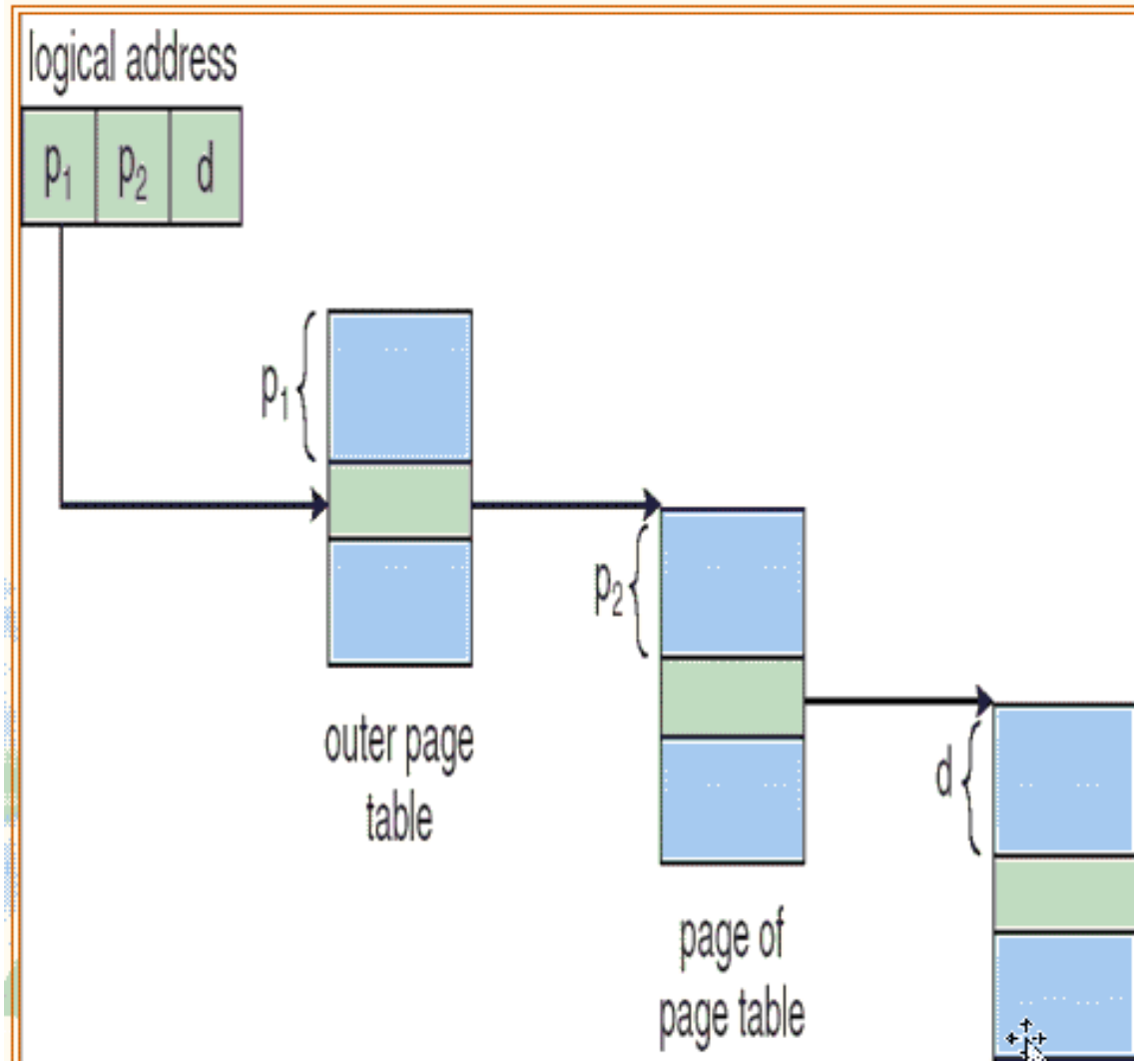
UNIVERSIDAD CATOLICA DEL MAULE

# Paginación varios niveles

- Para evitar esta fragmentación.
  - Paginan las tablas de paginas
    - La tabla de pagina está partida en trozo, cada uno del tamaño de una página, que se reparte arbitrariamente por la memoria física.
    - Para saber donde esta los trozos de la tabla de paginas, se utiliza otra tabla de pagina que Intel denomina directorio.
  - Tiempo de traducción de direcciones aumenta
    - TLB para reducir tiempo acceso

UNIVERSIDAD CATOLICA DEL MAULE

# Ejemplo de paginación de dos niveles




UNIVERSIDAD CATOLICA DEL MAULE

# Problema

Considere un computador con 64 Mb de memoria principal, que utiliza una memoria segmentada y cuya dirección lógica tiene esta estructura: 10 bits para el campo de segmento y 22 bits para el campo desplazamiento. En un momento dado la memoria esta organizada de la siguiente forma:

- SO en los primeros 128 Kbytes de memoria física.
- Hay un único proceso P con cuatro segmentos asignados: los segmentos 0,1 y 2 tiene longitud respectivamente 10Kbyte, 25Kbyte y 2 Kbyte y se hallan físicamente en las posiciones de memoria 300.000, 200.000 y 700.000 respectivamente. El segmento 3 permite acceder a toda el área del SO.

UNIVERSIDAD CATOLICA DEL MAULE

# Preguntas

A. ¿Cuál es el tamaño máximo del espacio direccionable por la maquina?

B. ¿Cuál es el máximo tamaño posible del segmento?

C. Represente la tabla de segmentos del proceso P.

D. ¿Qué dirección física se corresponde con la dirección lógica con segmento 2 y desplazamiento 1.000?

E. SI el sistema utiliza una política de ubicación del mejor ajuste (best-fit) y se solicita espacio para un segmento de 400Kb ¿ que zona de memoria se asignaría?

UNIVERSIDAD CATOLICA DEL MAULE