Gestión de Memoria

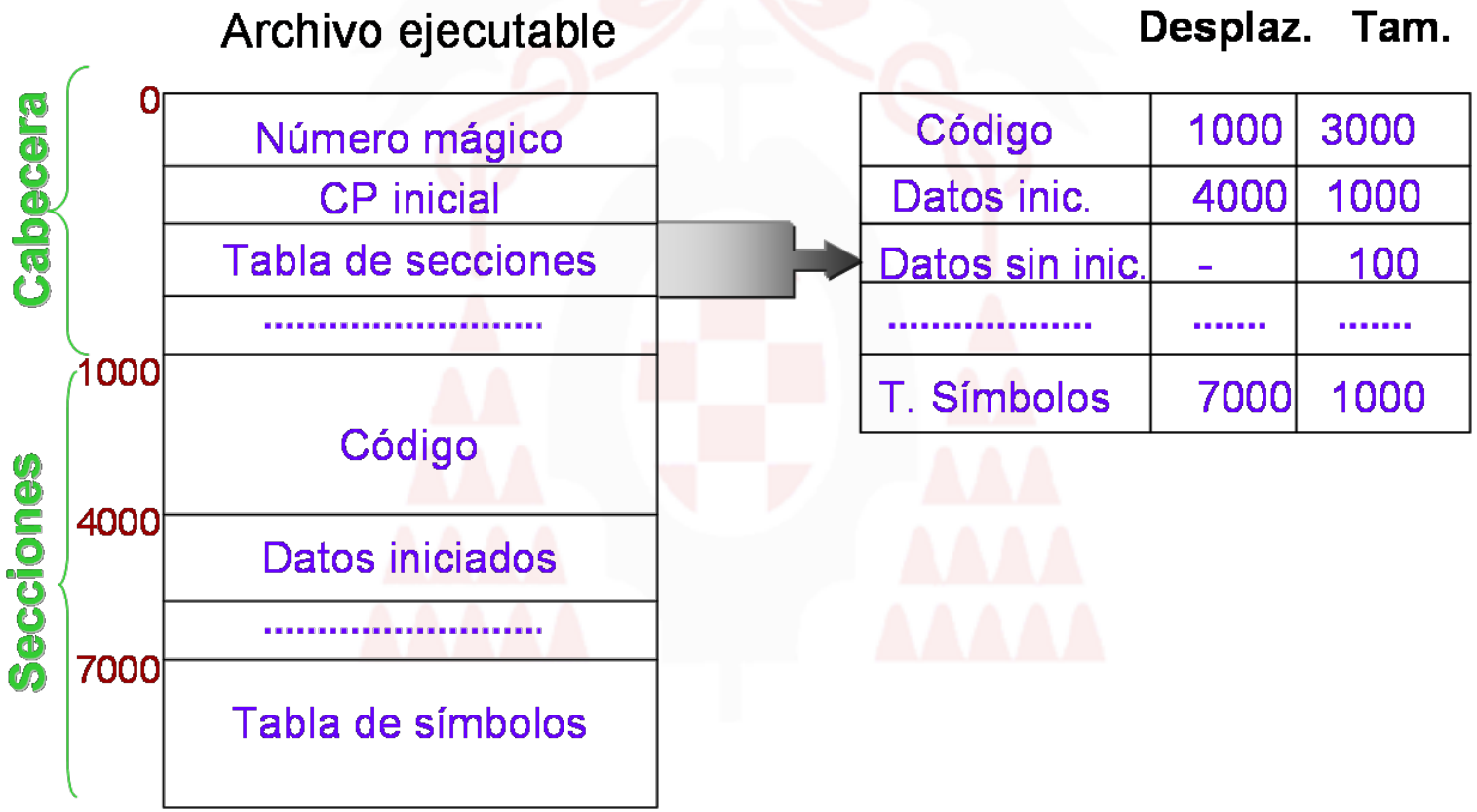
E. Campo M. Knoblauch Ó. López J. Clemente

Departamento de Automática
Universidad de Alcalá

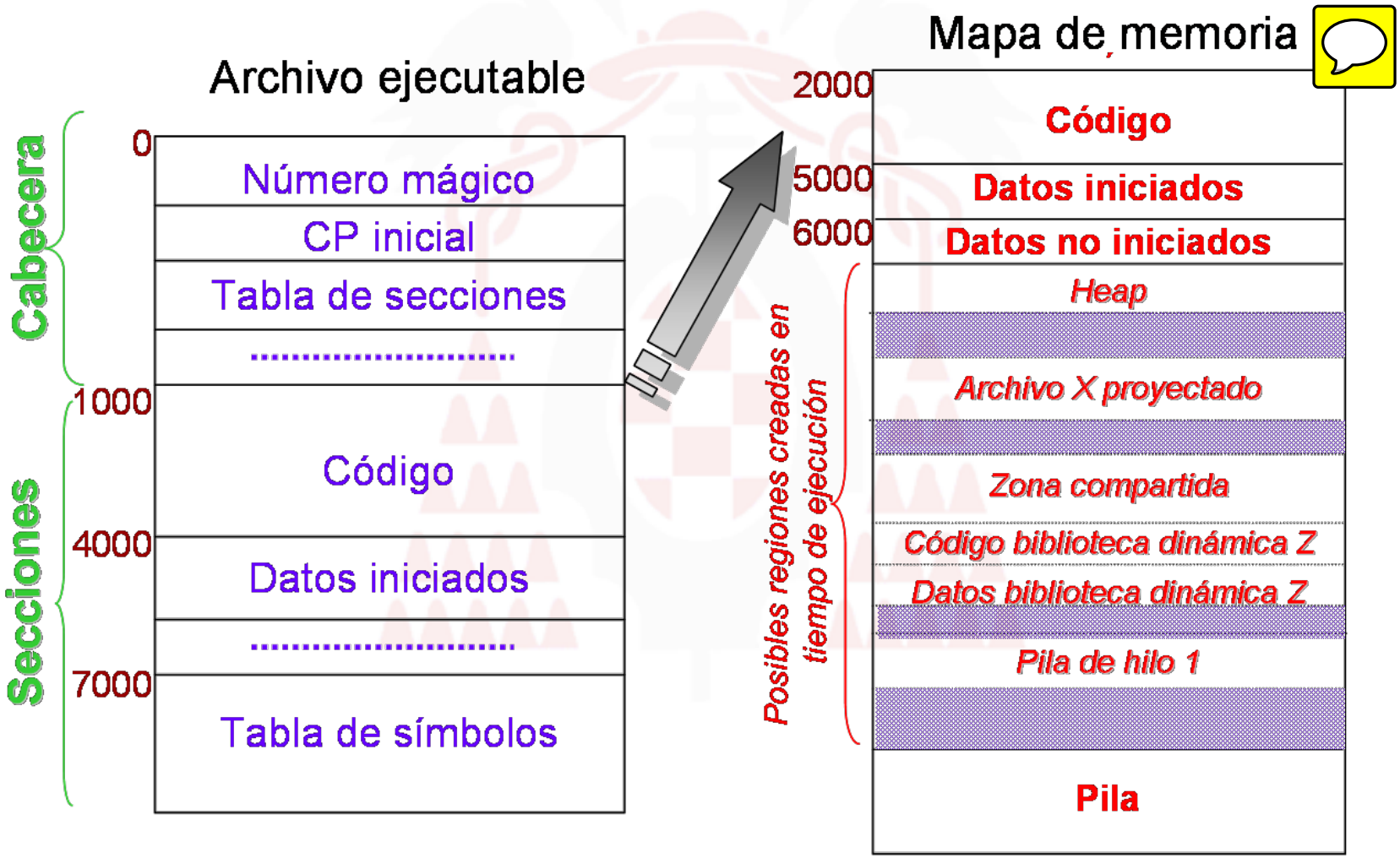
/gso>

Índice

Formato de archivo ejecutable



Mapa de memoria de un proceso



Programa y proceso

```
...
char * nomprog, int cont = 1;

void Func (int x)
{
    int resultado = 0;
    char caracter = 'a';

    if (cont)
        cont = resultado ++;
    ...
    return;
}

main (int argc, char * argv[])
{
    int i;
    char * nomprog;

    Func (cont);
    nomprog = (char *) malloc (1 + strlen(argv[0]));
    ...
    free (nomprog);
    ...
    exit (0);
}
```

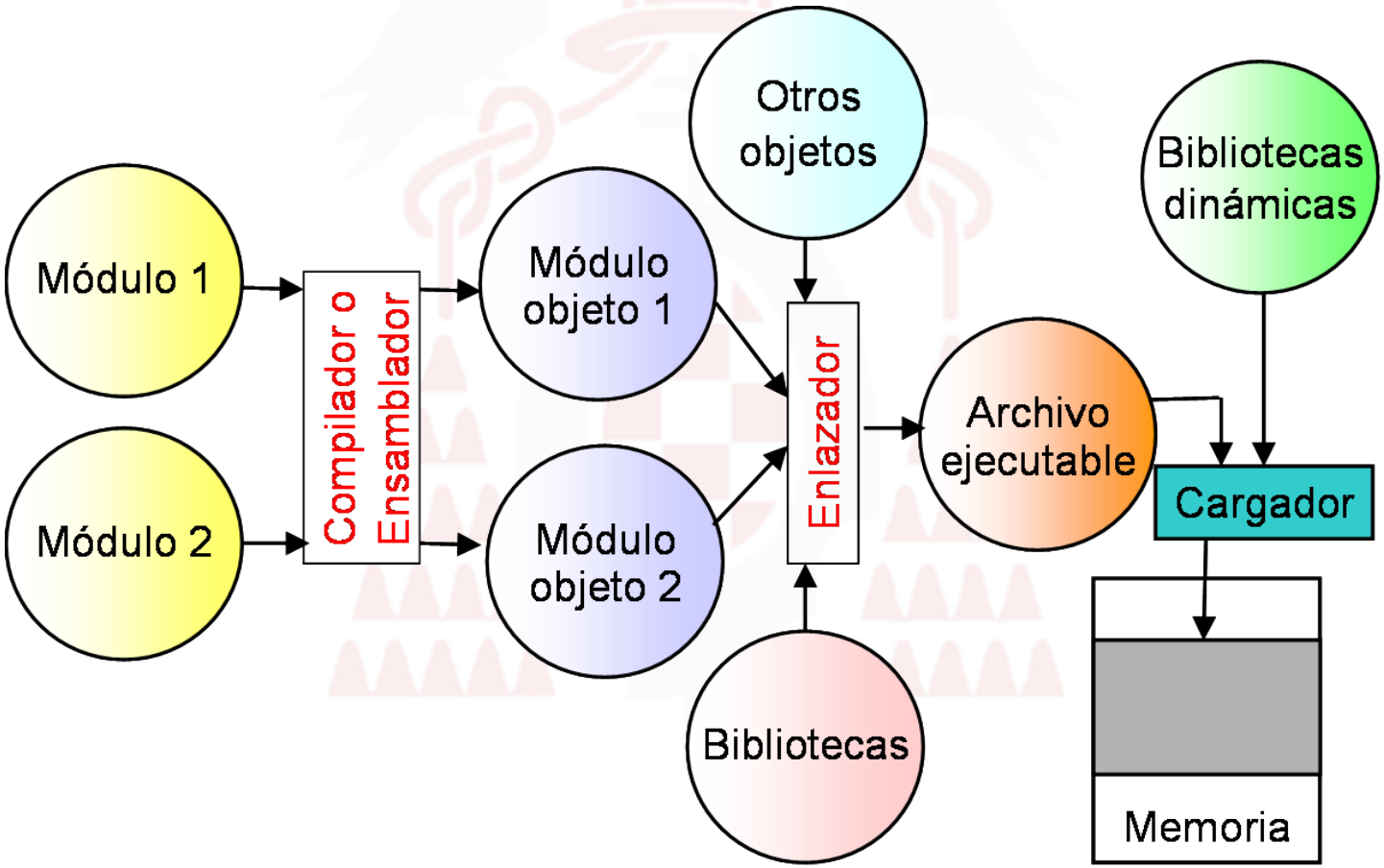
Abstracciones de direccionamiento

Espacio de direccionamiento

Conjunto de direcciones referenciables

- Espacio de direccionamiento virtual $\Rightarrow$ independiente para cada proceso
- Espacio de direccionamiento físico $\Rightarrow$ repartido entre todos los procesos
- Los procesos sólo referencian direcciones virtuales
- Tiene que haber una traducción de dirección virtual a física transparente al proceso

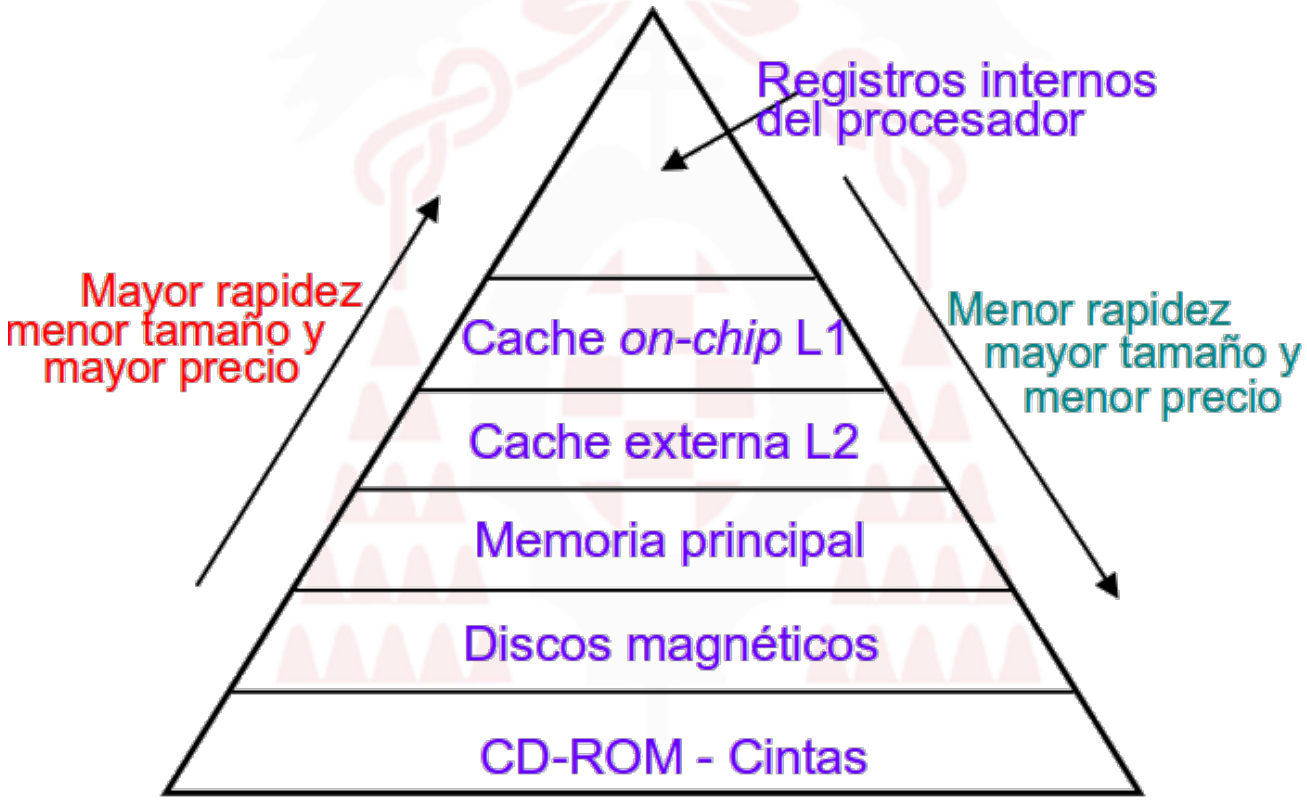
Abstracciones de direccionamiento



Jerarquía de memoria en un computador

- La jerarquización de la memoria es un intento de aumentar el rendimiento de los computadores
- Para ello se aprovechan los avances tecnológicos en el diseño de memorias y la localidad de los programas
- Memorias rápidas: tienen un coste elevado y una capacidad pequeña
- Memorias lentas: son baratas y tienen una capacidad alta

Esquema de jerarquía de memoria



Principio de localidad

- Los procesos tienden a concentrar sus referencias en un intervalo de tiempo en un subconjunto de su espacio de direcciones

Donald Knuth [1971]:

Los programas, normalmente, tienen un perfil muy desigual, con unos pocos picos agudos. [..] También encontramos que menos del 4 por 100 de un programa, generalmente, representa más de la mitad de su tiempo de ejecución

- Es una propiedad empírica
- La localidad puede ser de dos tipos:
  - Localidad espacial
  - Localidad temporal

Localidad espacial

- Una vez hecha una referencia a una posición de memoria, es muy probable que las localidades cercanas sean también referenciadas.
- En apoyo a esta observación encontramos:
  - Ejecución secuencial del código
  - Tendencia de los programadores a colocar próximas entre sí las variables relacionadas
  - Acceso a estructuras de datos de tipo matriz o pila

Localidad temporal

- Una vez hecha una referencia a una posición de memoria en un determinado instante t, es muy probable que esa misma posición vuelva a ser accedida en un instante $t + \Delta t$
- Justificada por:
  - Formación de ciclos
  - Subrutinas
  - Pilas

Fragmentación

Fragmentación

Desaprovechamiento de la memoria libre disponible debido al mecanismo de gestión utilizado

- Puede ser de dos tipos, interna y externa
- Fragmentación interna
  - Se debe a la diferencia de tamaño entre la partición de memoria y el objeto residente dentro de ella
- Fragmentación externa
  - Se debe al desaprovechamiento de memoria entre particiones

Reubicación

Reubicación

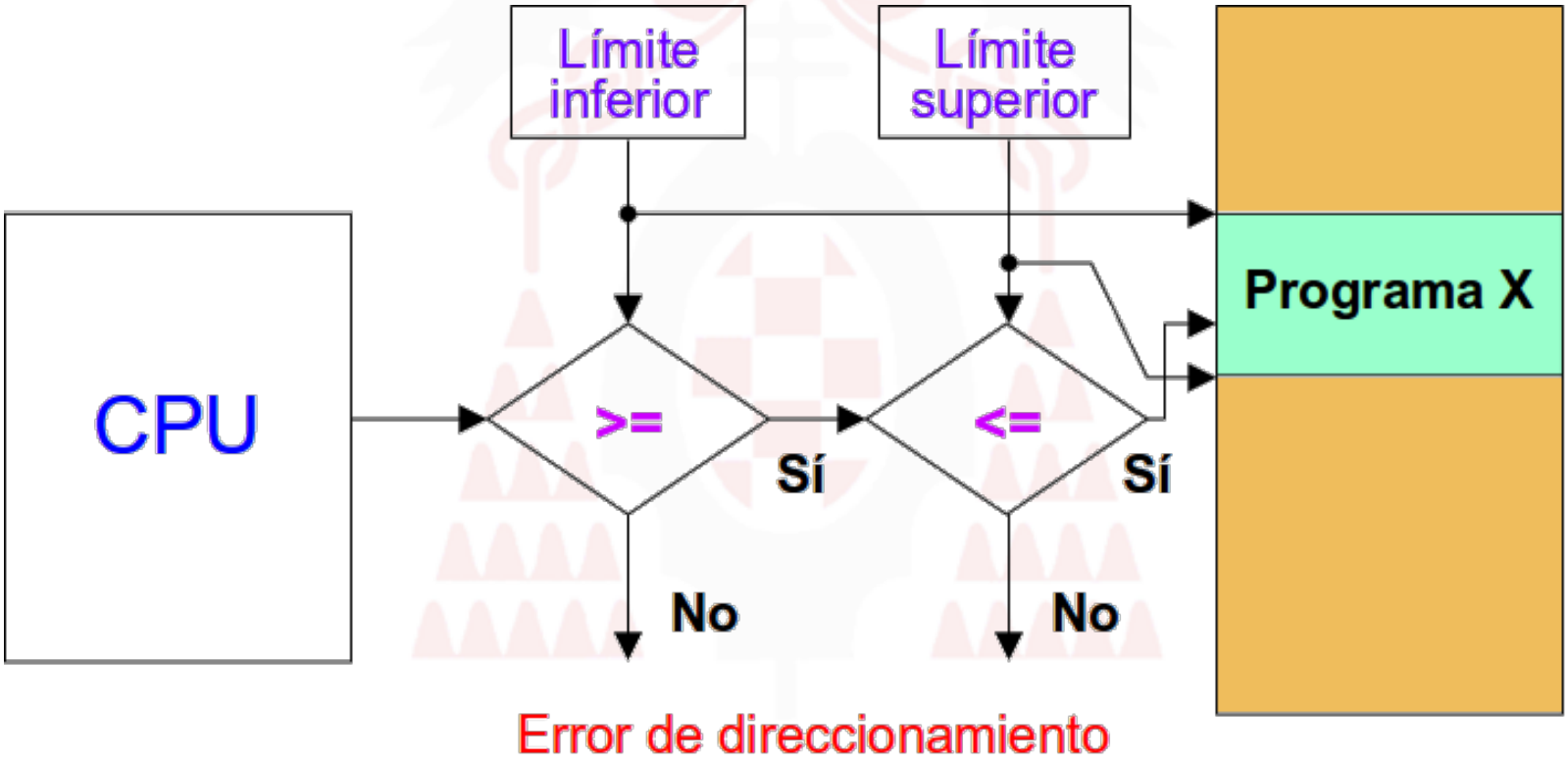
Proceso de asignar direcciones a las diferentes partes de un programa (código, datos, pila...)

- En función de cuándo se conozca la ubicación definitiva, se puede producir en la etapa de compilación, en la etapa de carga o en la etapa de ejecución
- Reubicación estática
  - Se realiza antes o durante la carga del programa
  - Los programas no pueden ser movidos una vez iniciados
- Reubicación dinámica
  - El paso de dirección virtual a dirección real, se realiza en tiempo de ejecución
  - Necesita hardware adicional (MMU)
  - Los programas pueden moverse en tiempo de ejecución

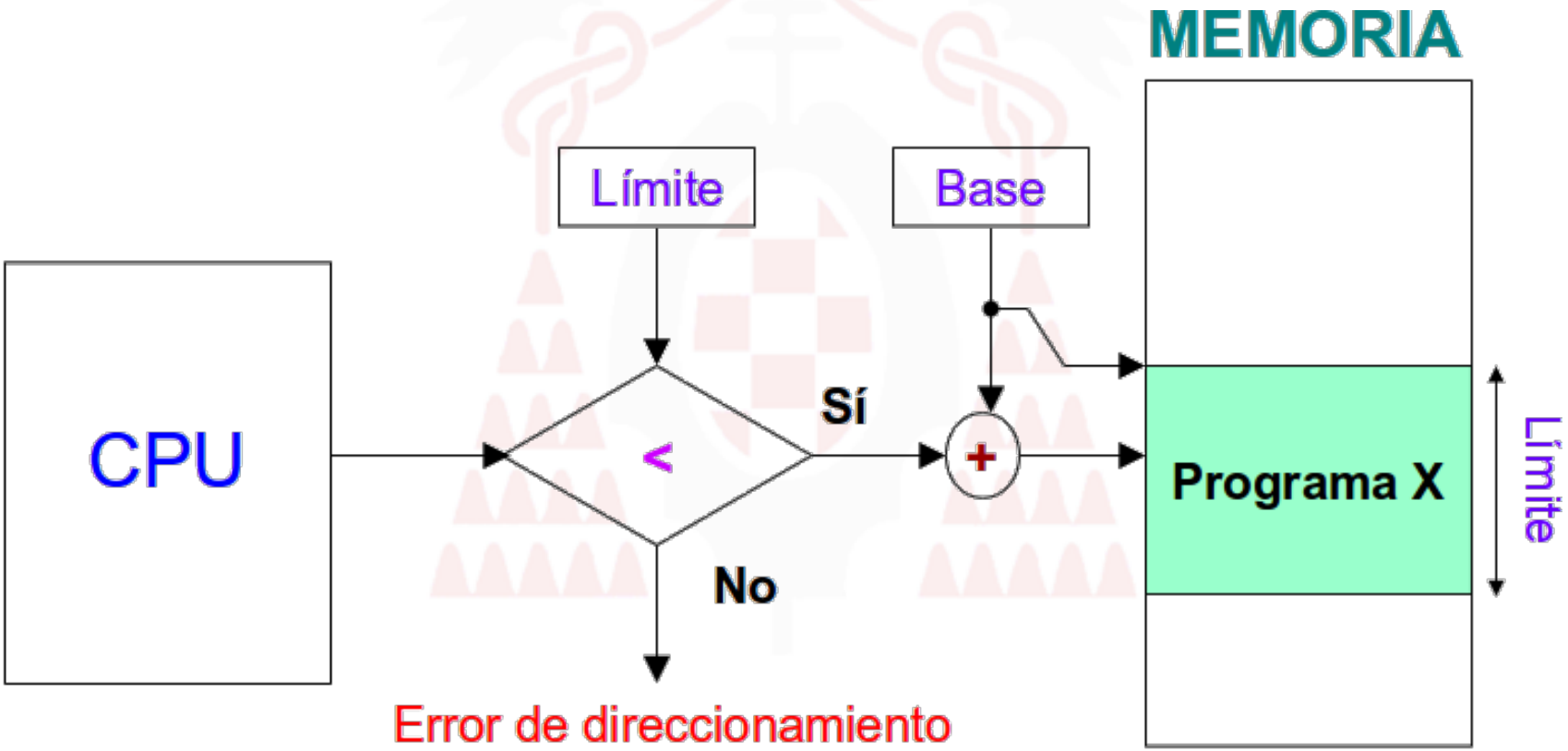
Protección y uso compartido

- Necesario delimitar acceso a memoria:
  - Sistema operativo
  - Procesos de usuario
- Metodos de protección
  - Registros límite
  - Registros base y límite
  - Bits de protección en memoria
  - Derechos de acceso en tablas de traducción
  - ¿Dónde se almacenan?
- ¿Cómo compartir memoria entre procesos?

Registros límite



Registros base y límite



Evolución histórica

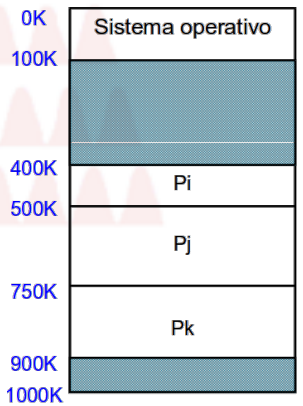
- Máquina desnuda
  - El sistema no proporciona ningún servicio
- Monitor monolítico
  - Además del sistema operativo, hay un único proceso
- Memoria particionada contigua
  - Multiprogramación con número fijo de tareas (MFT)
    - Particiones de tamaño fijo
    - Se crean al arrancar el sistema
  - Multiprogramación con número variable de tareas (MVT)
    - Particiones de tamaño variable
    - Se crean cuando un proceso lo necesita
- Memoria particionada no contigua 

Memoria particionada no contigua

- El contenido de un proceso puede ser distribuído entre diferentes particiones separadas en memoria
- La memoria está organizada en particiones:
  - De tamaño variable ⇒ segmentos
  - De tamaño fijo ⇒ marcos

Tabla de descripción de particiones

- Independiente por proceso
- Se construye en tiempo de carga del proceso en memoria

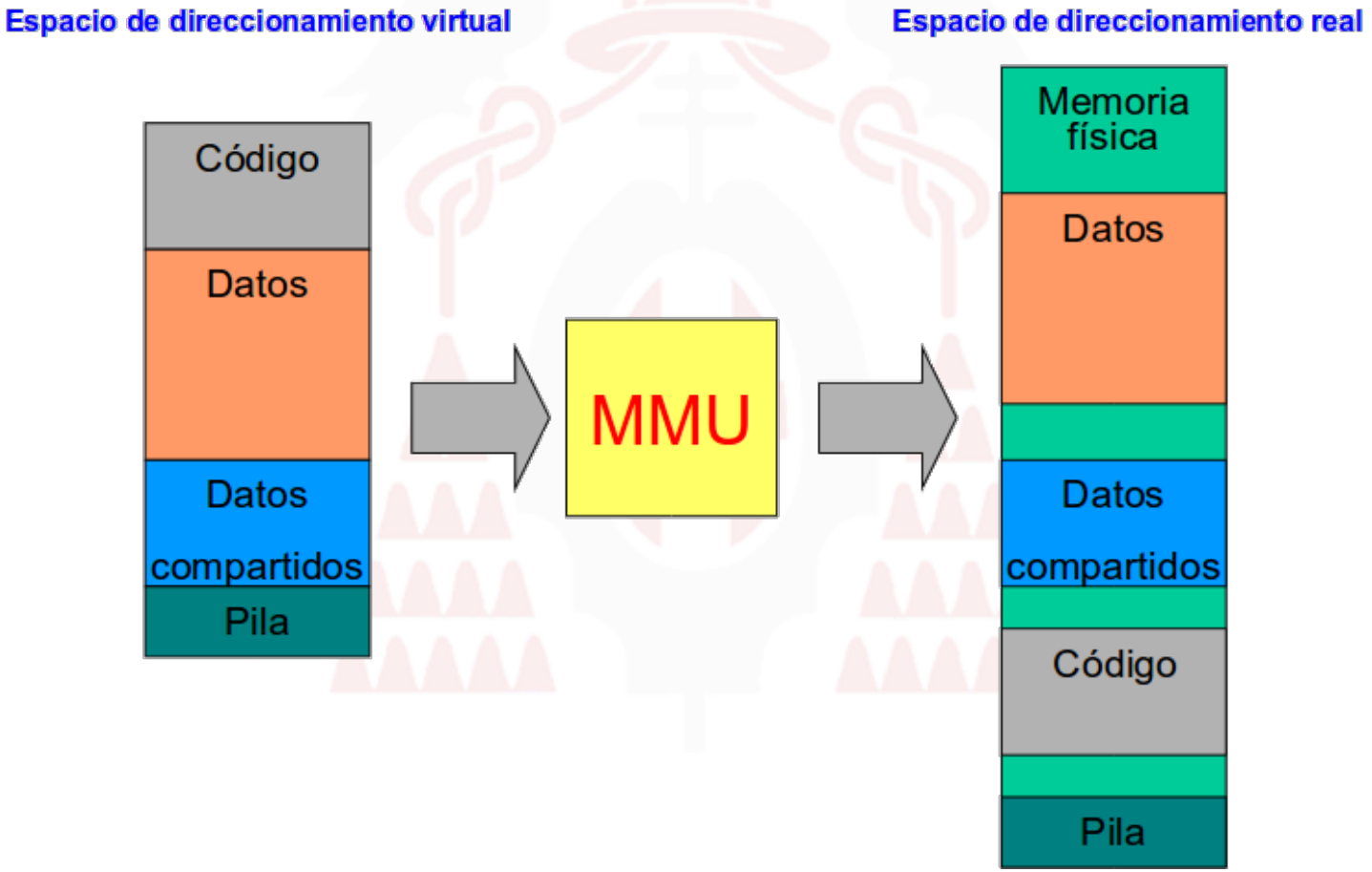


Número de la partición	Base de la partición	Tamaño de la partición	Estado de la partición
0	0K	100K	ASIGNADA
1	100K	300K	LIBRE
2	400K	100K	ASIGNADA
3	500K	250K	ASIGNADA
4	750K	150K	ASIGNADA
5	900K	100K	LIBRE

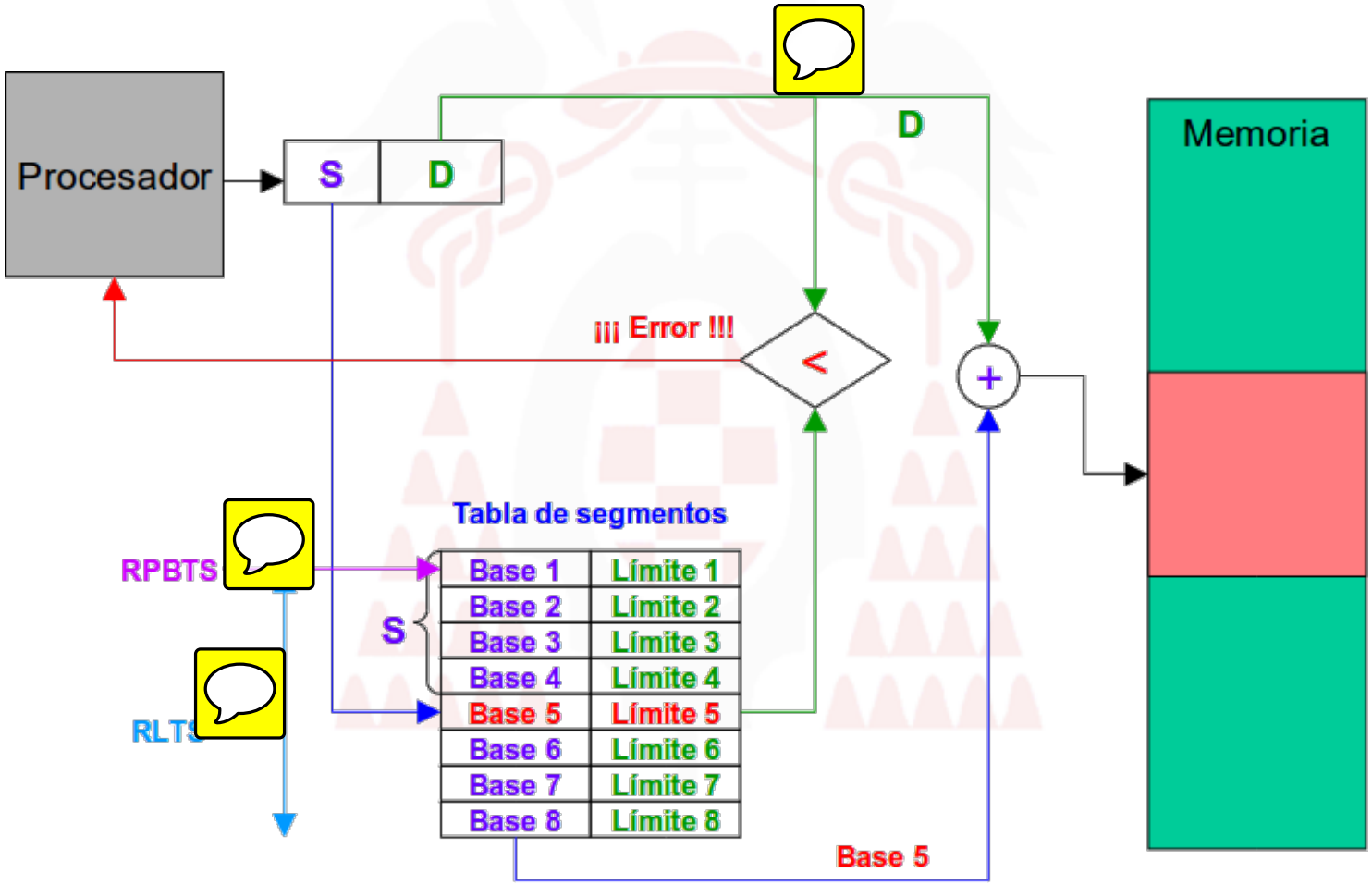
Segmentación

- Inicialmente la memoria física está organizada como un único bloque vacío donde se crean particiones de tamaño variable (segmentos) a medida que se van necesitando
- El espacio de direccionamiento virtual se organiza en segmentos
- Posee mecanismo de protección y permite uso compartido
- Las direcciones virtuales tienen dos componentes: número de segmento y desplazamiento
- Tabla de particiones denominada Tabla de Segmentos (TDS)
- Si la TDS es muy grande, es necesario almacenarla en memoria principal apuntada por un registro (RPBTS) $\Rightarrow$ necesarias dos referencias a memoria por acceso

Segmentación: esquema lógico



Segmentación: esquema físico



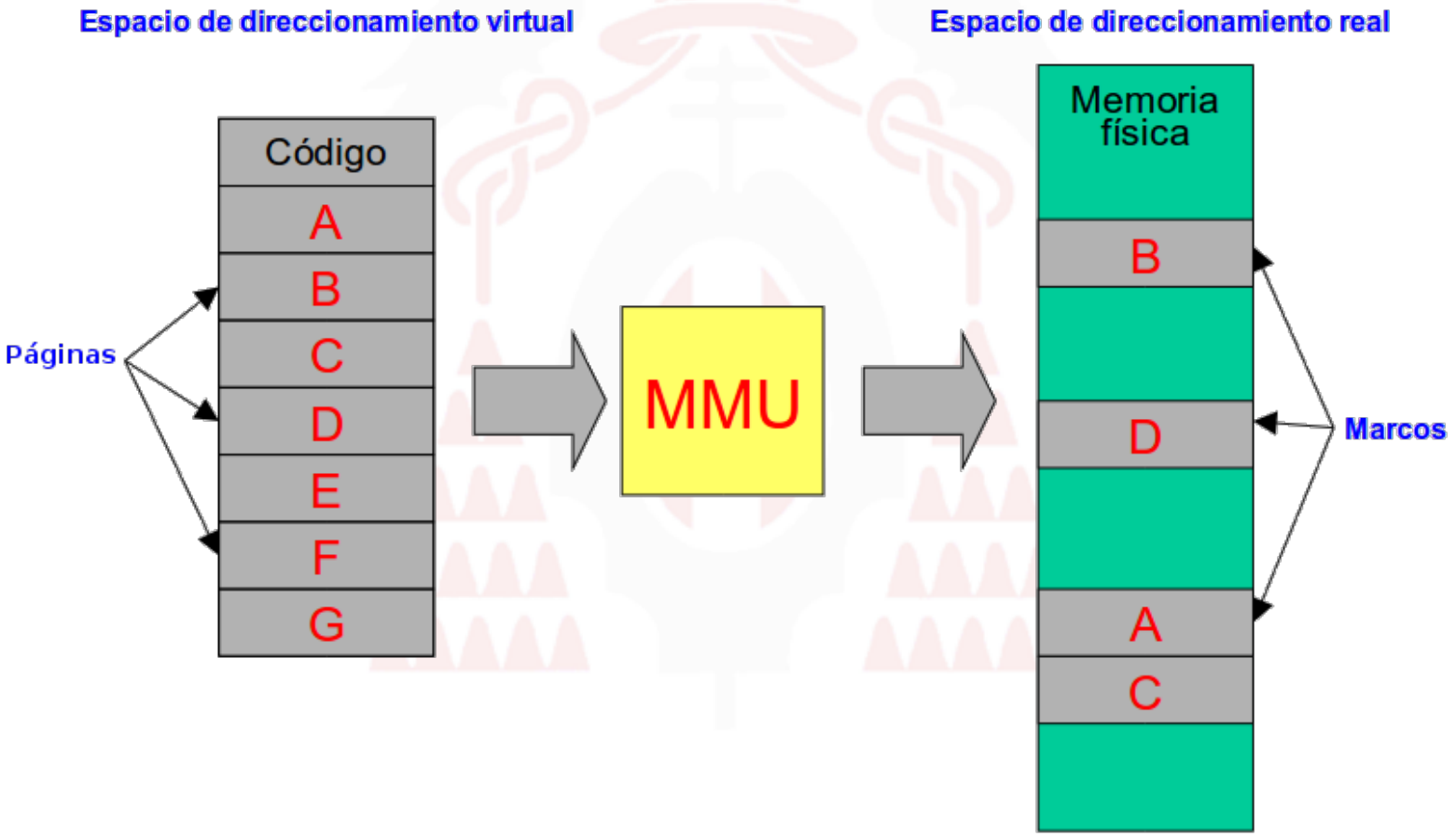
Consideraciones segmentación

- Ventajas:
  - No produce fragmentación interna
  - Crecimiento dinámico de los segmentos
- Inconvenientes:
  - Necesita compactación de memoria
  - Puede aparecer fragmentación externa

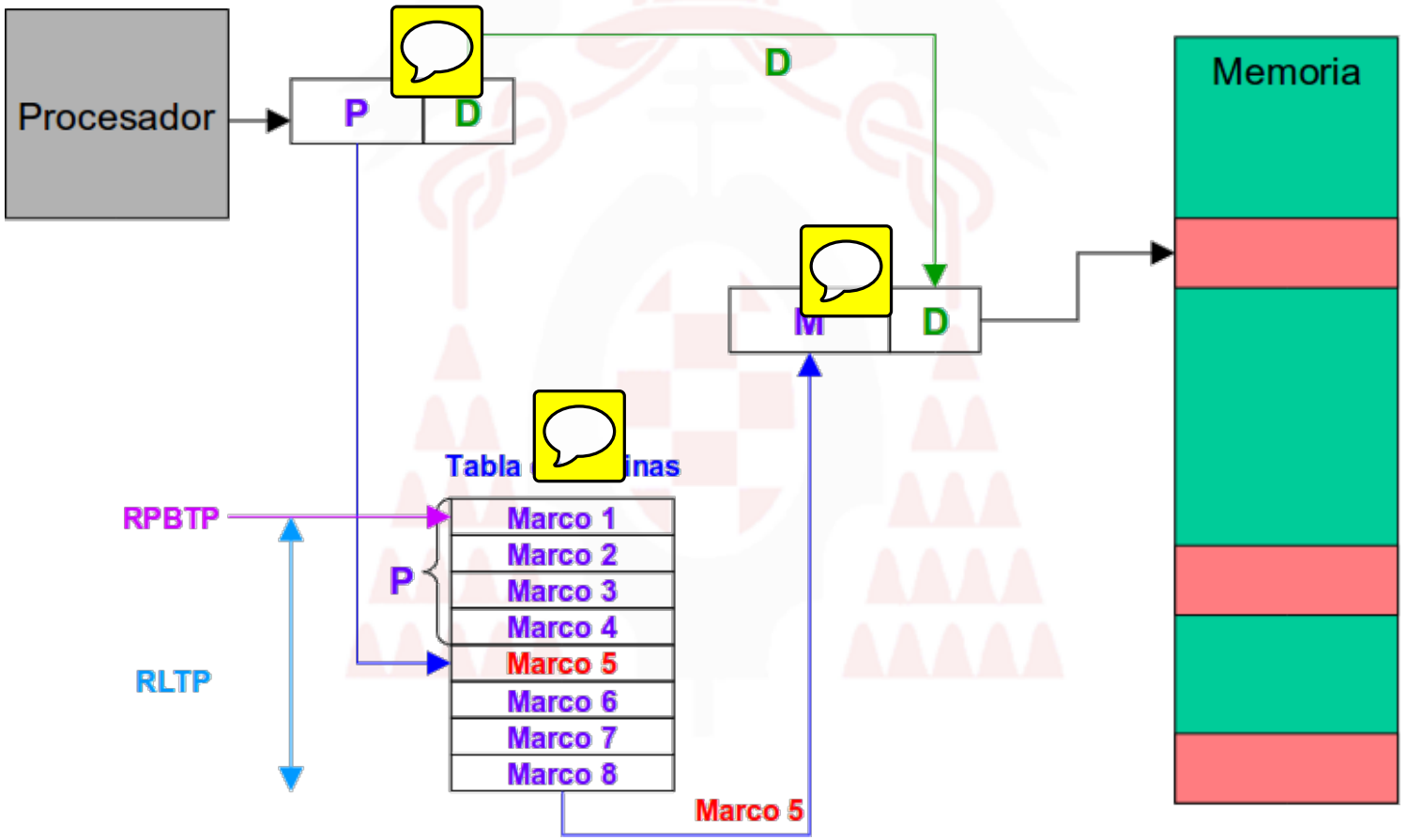
Paginación

- Inicialmente la memoria física está organizada en particiones de tamaño fijo (marcos)
- El espacio de direcciones virtuales de un proceso está dividido en bloques de tamaño fijo llamados páginas
- Las direcciones virtuales tienen dos componentes: número de página virtual y desplazamiento
- Posee mecanismo de protección y permite uso compartido
- Tabla de particiones denominada Tabla de Mapa de Páginas (TMP)
- Si la TMP es muy grande, es necesario almacenarla en memoria principal apuntada por un registro (RPBTP)

Paginación: esquema lógico



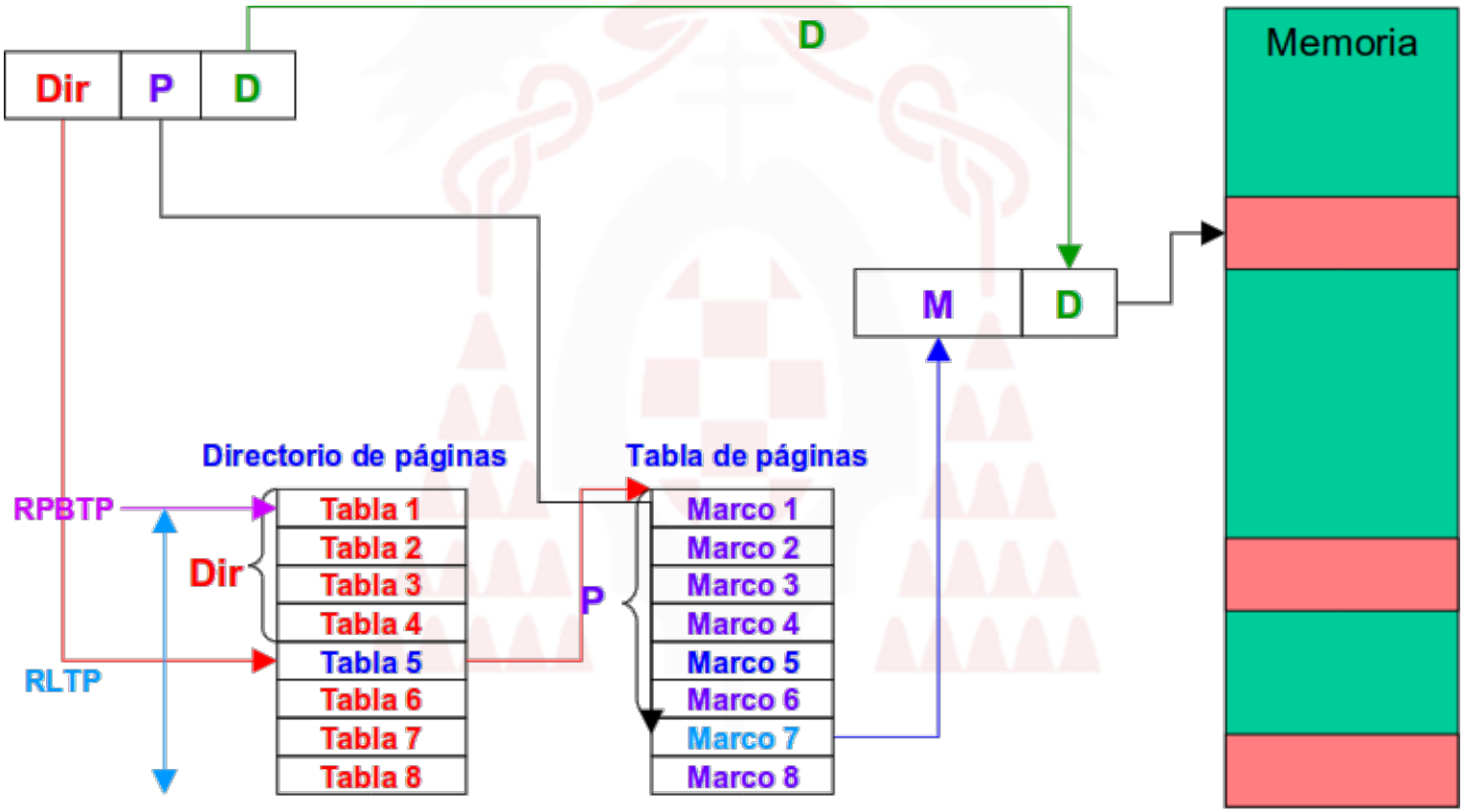
Paginación: esquema físico



Consideraciones paginación

- Ventajas:
  - No produce fragmentación externa
- Inconvenientes:
  - Puede aparecer fragmentación interna
- Con páginas grandes aumenta la fragmentación interna pero disminuye el tamaño de la TMP y viceversa
- Si el número de páginas es grande, la zona de memoria ocupada por la TMP puede ser excesiva, por lo que hay que paginar la tabla de páginas

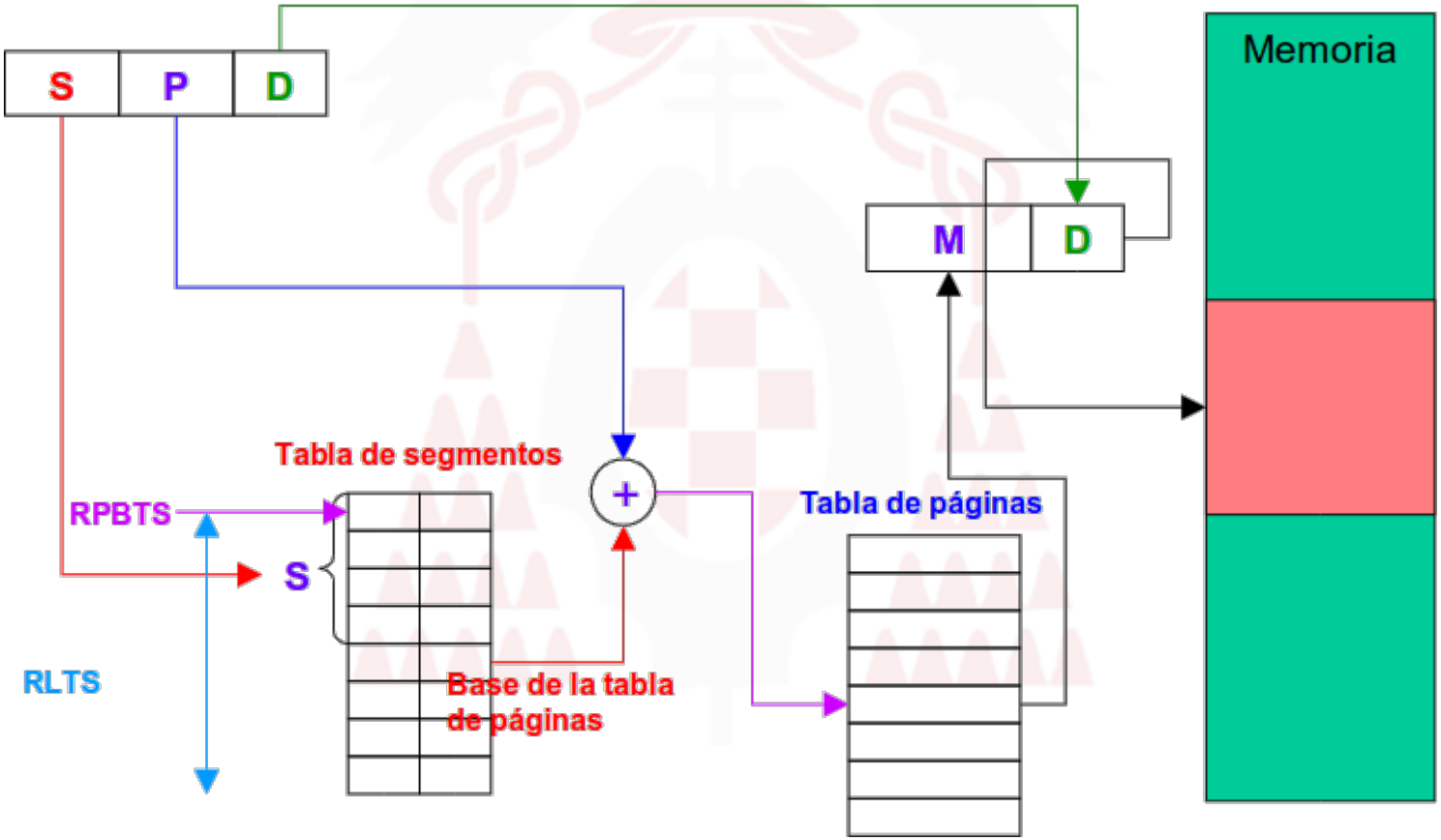
Paginación paginada



Combinación de mecanismos

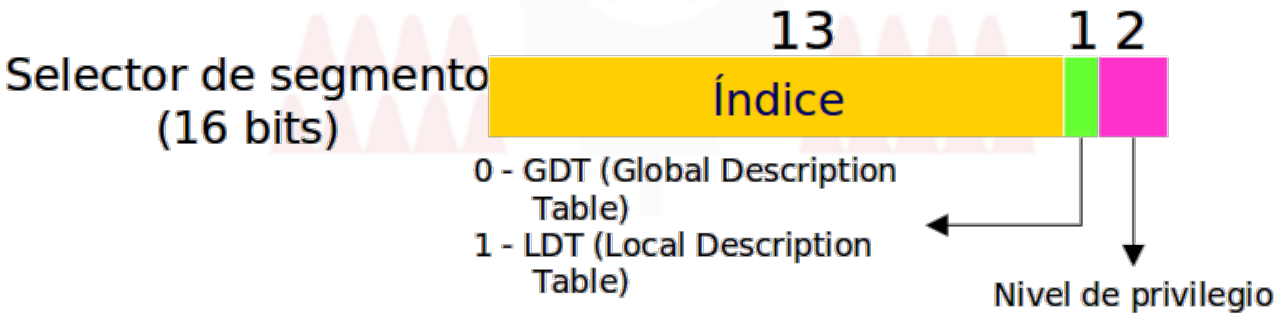
- Es posible combinar los esquemas de paginación y segmentación
- Se obtienen las ventajas de ambos esquemas a costa de complicar el hardware
- Posibles combinaciones:
  - Segmentación paginada
  - Paginación segmentada (no se emplea en la práctica)

Segmentación paginada: esquema lógico



MMU del Pentium

- El Pentium soporta segmentación, paginación y segmentación paginada (la más habitual)
- La dirección lógica está compuesta por un selector de segmento (13+1 bits) y un desplazamiento (32 bits)
- El selector de segmento es uno de los siguientes registros: CS, DS, ES, SS, FS, GS

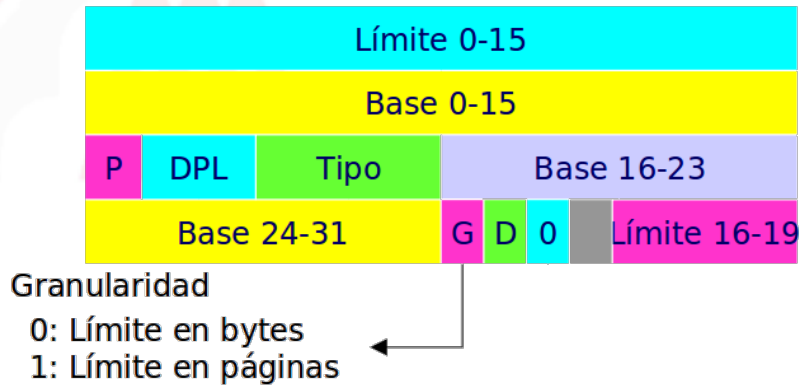


Formato del descriptor de segmento

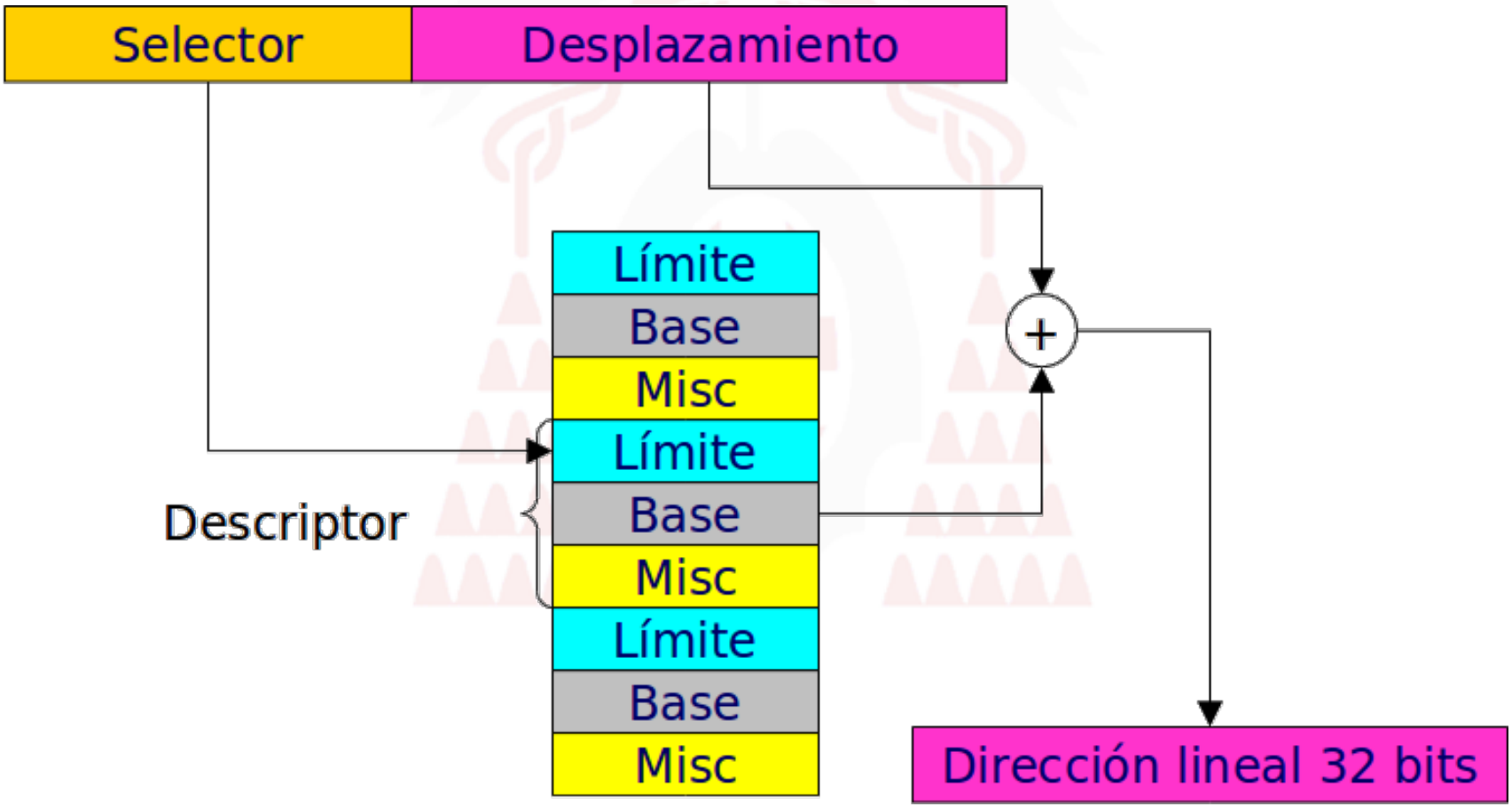
- LDT (Local Descriptor Table) $\Rightarrow$ una por proceso
- GDT (Global Descriptor Table) $\Rightarrow$ una por sistema
- Número máximo de entradas en cada tabla $\Rightarrow 2^{13}$
- Cada entrada en la tabla de segmentos se denomina descriptor
- Tamaño del descriptor $\Rightarrow$ 8 bytes

Descriptor de segmento

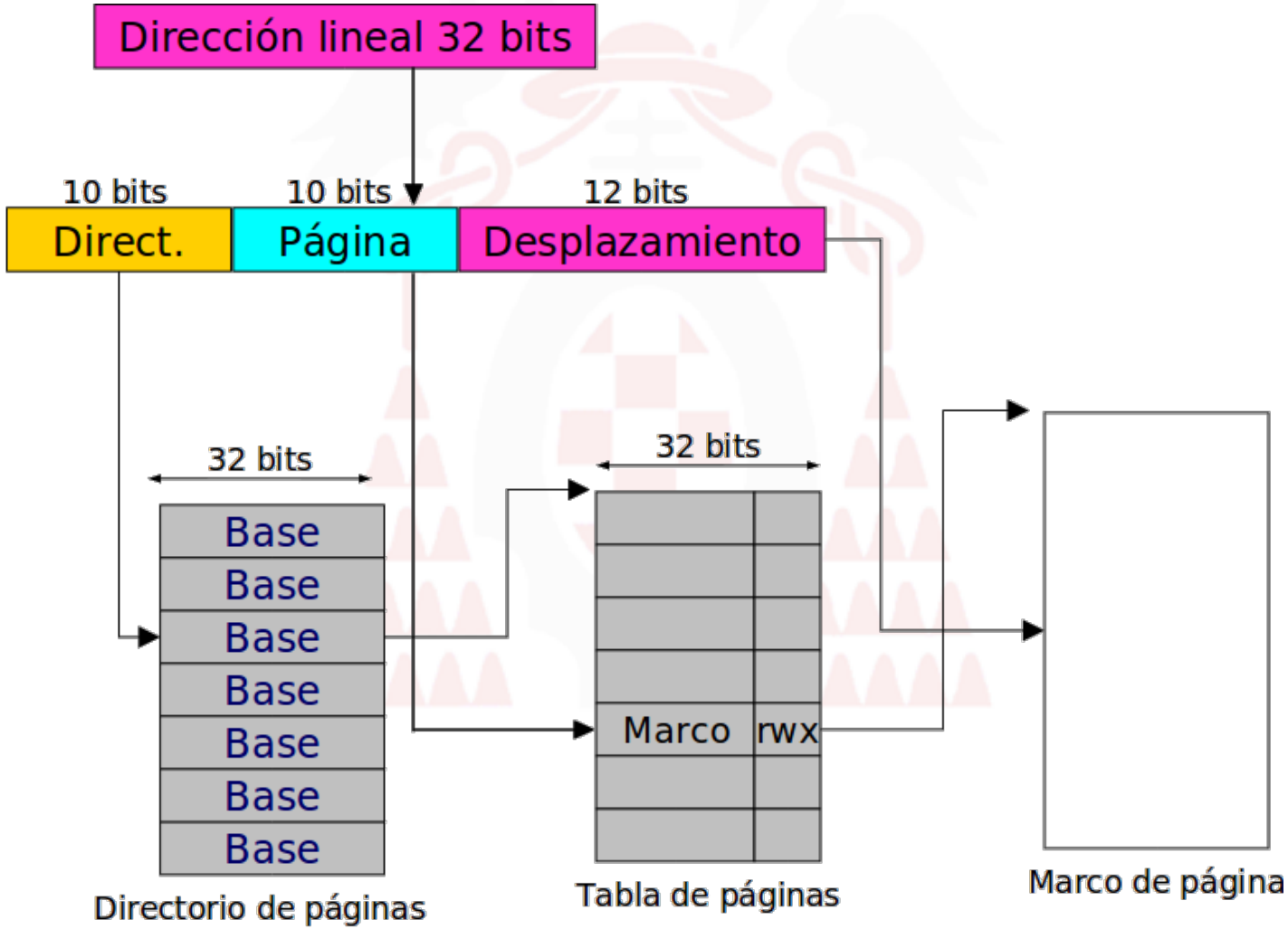
- Dirección base (32 bits)
- Límite (20 bits)
- Atributos y privilegios (12 bits)



Etapa de segmentación del Pentium



Etapa de paginación del Pentium



Referencias bibliográficas I

- [Sánchez, 2005] S. Sánchez Prieto.
  Sistemas Operativos.
  Servicio de Publicaciones de la UA, 2005.
- [Tanenbaum, 2009] A. Tanenbaum.
  Sistemas Operativos Modernos.
  Ed. Pearson Education, 2009.
- [Stallings, 1999] W. Stallings.
  Organización y arquitectura de Computadores.
  Ed. Prentice Hall, 1999.
- [Silberschatz, 2006] A. Silberschatz, P. B. Galván y G. Gagne
  Fundamentos de Sistemas Operativos.
  McGraw Hill. 2006

Referencias bibliográficas II


D.E. Knuth.
An Empirical Study of FORTRAN Programs
Software—Practice and Experience, vol. 1, 105–133, 1971.