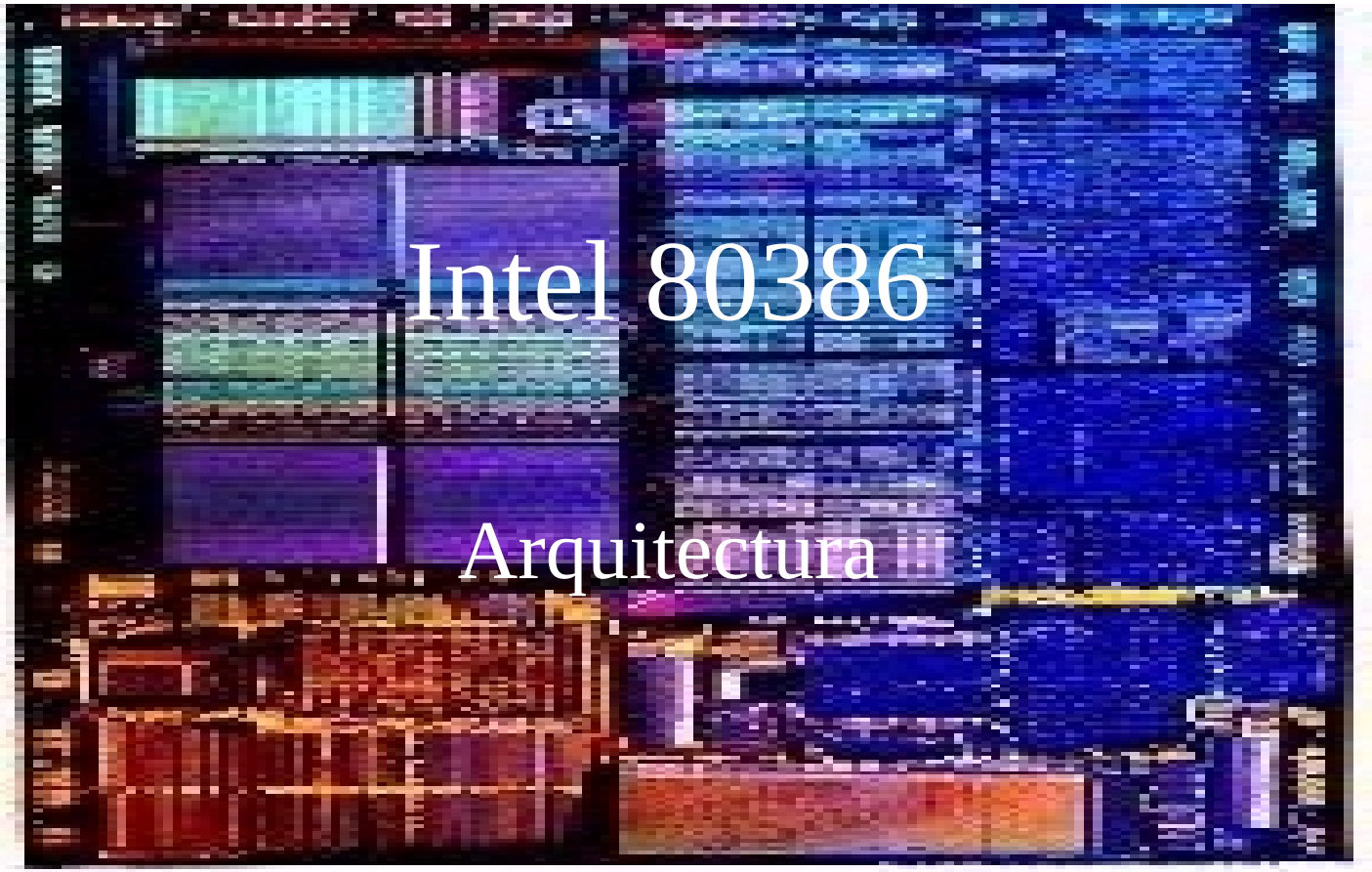


Intel 80386

Arquitectura

Características generales

- Procesador de 32 bits
  - Bus de direcciones de 32 bits : 4 Gbyte
  - Bus de datos interno de 32 bits
- primer procesador de 32 bits de Intel
- 138 instrucciones (49 más que el 8086)
- coprocesador 80387 externo

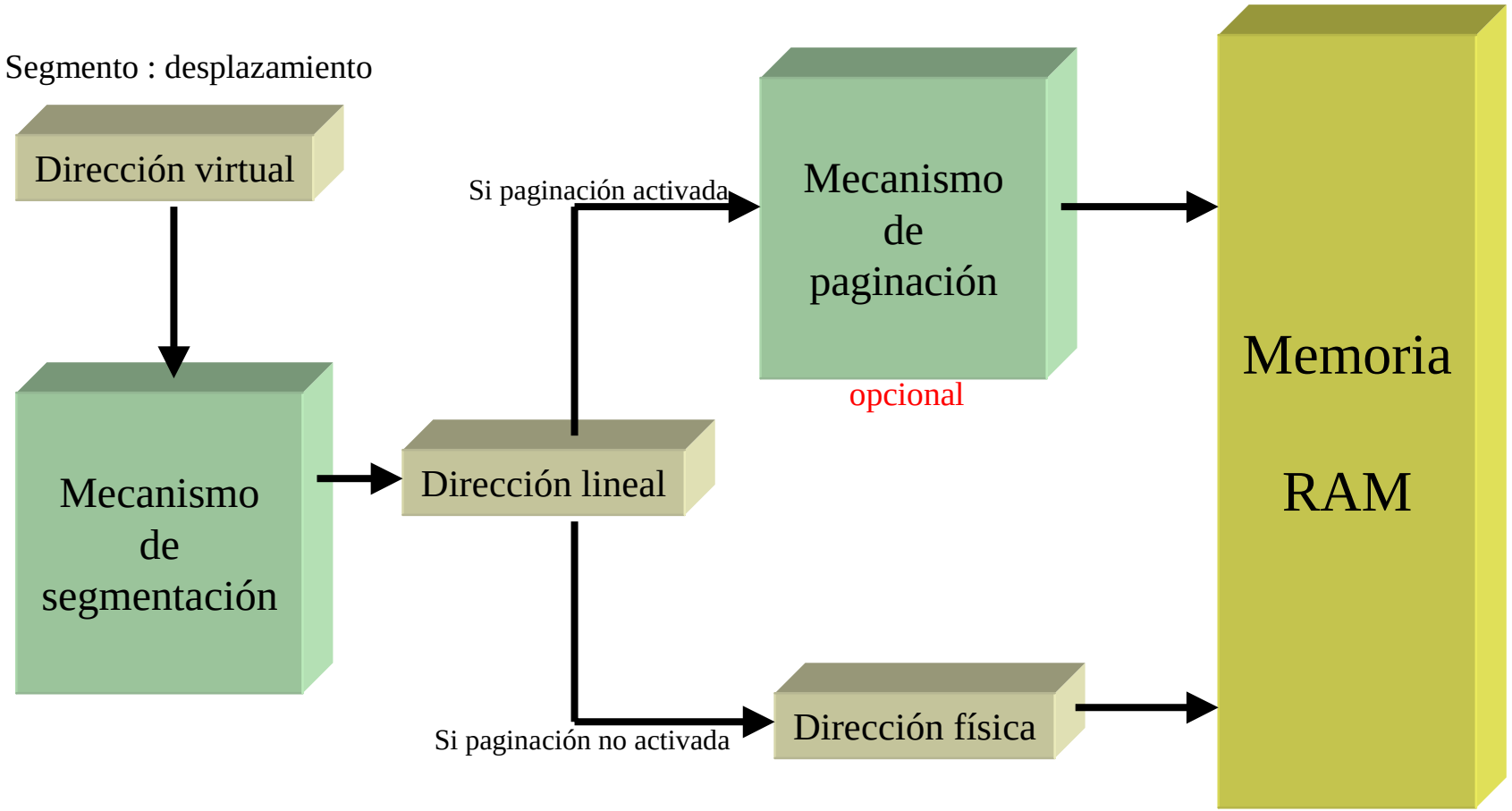
Tipos de datos

- ASCII
- BCD
- Entero sin signo
  - 8 bits 0..255
  - 16 bits 0..65535
  - 32 bits 0..4194304
- Entero con signo
  - 8 bits -128..127
  - 16 bits -32768..32767
  - 32 bits -2097152.. 2097151
- Cadenas secuencia de bytes o palabras
- Punto flotante
  - entero de palabra (16)
  - entero corto (32)
  - entero largo (64)
  - BCD empaqutado (80)
  - real corto (32)
  - real largo (64)
  - real temporal (80)

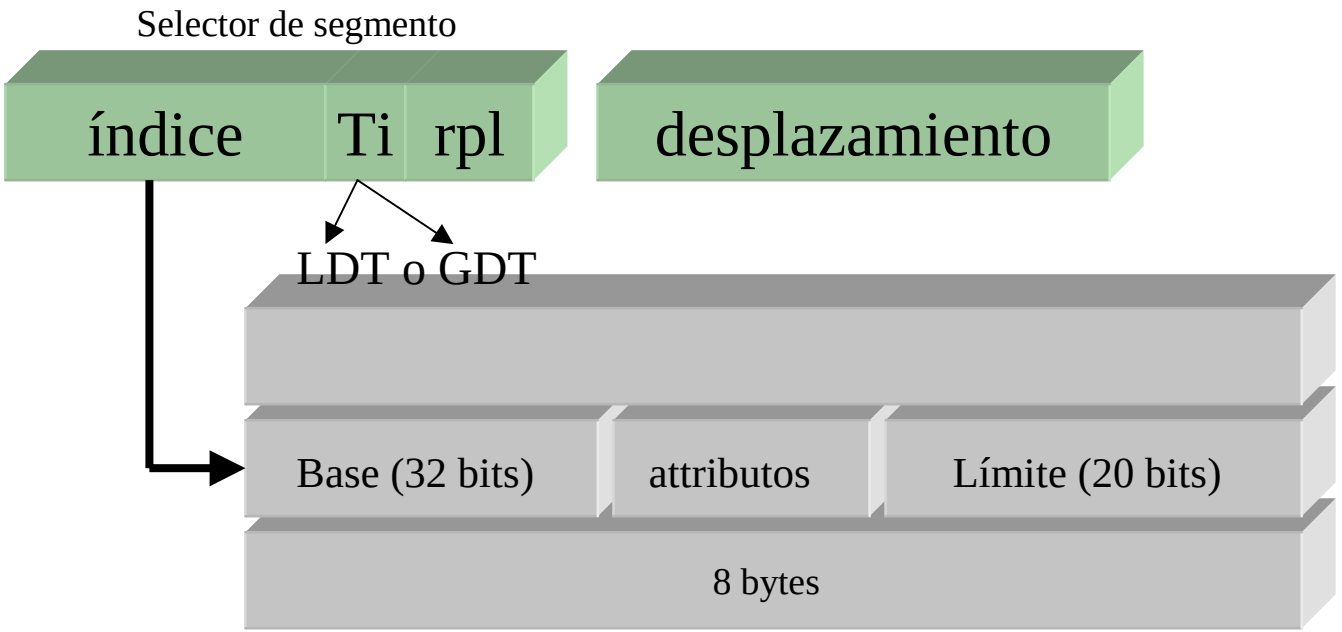
Modos de funcionamiento

- Modo real
  - igual que el 8086 pero con más instrucciones
- Modo protegido
  - se habilitan los registros extendidos
  - se habilita el modo de direccionamiento extendido
  - se habilita el sistema de segmentación y paginación
  - Funcionamiento multitarea
- Modo virtual 8086
  - emula el modo real dentro del modo protegido
  - acepta todas las instrucciones (excepto las protegidas)
  - acepta juego de registros extendido
  - acepta modos de direccionamiento extendido

Manejo de memoria

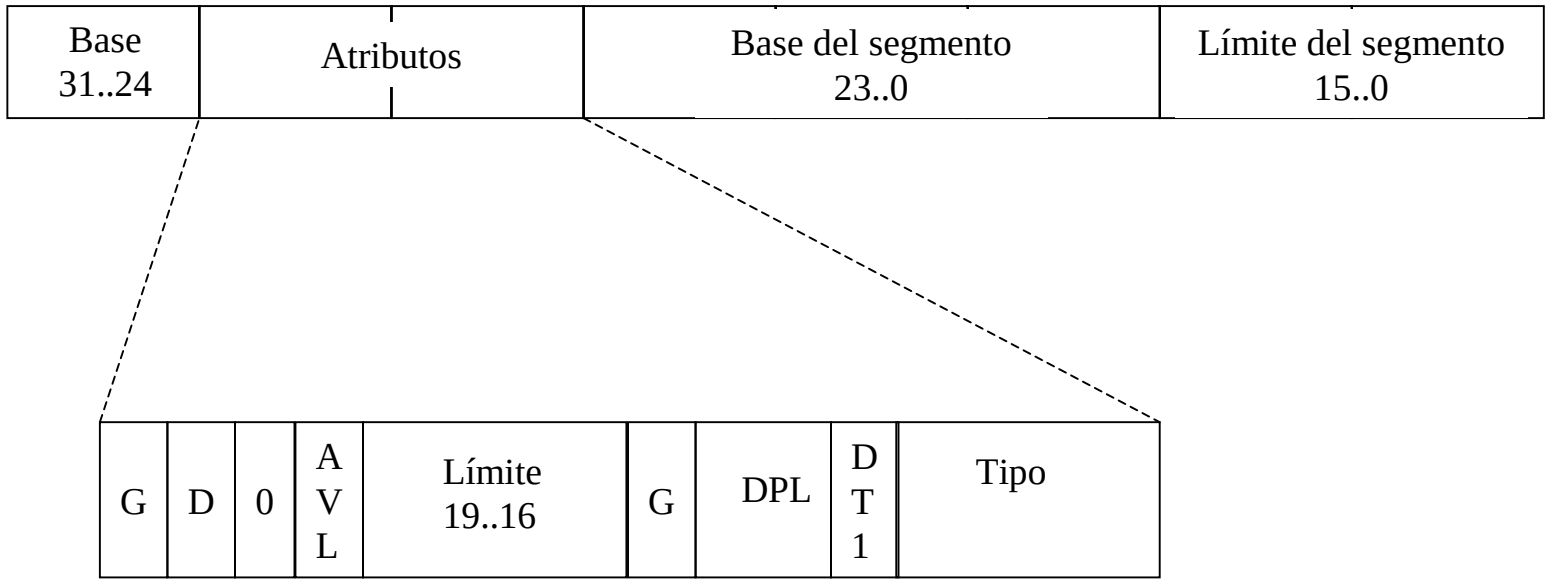


Mecanismo de segmentación



If desplazamiento < límite then
dirección lineal = base + desplazamiento

Formatos de los descriptores de segmento



G: granularidad
D: código 286/386
AVL: disponible para el software
tipo: tipo del descriptor de memoria

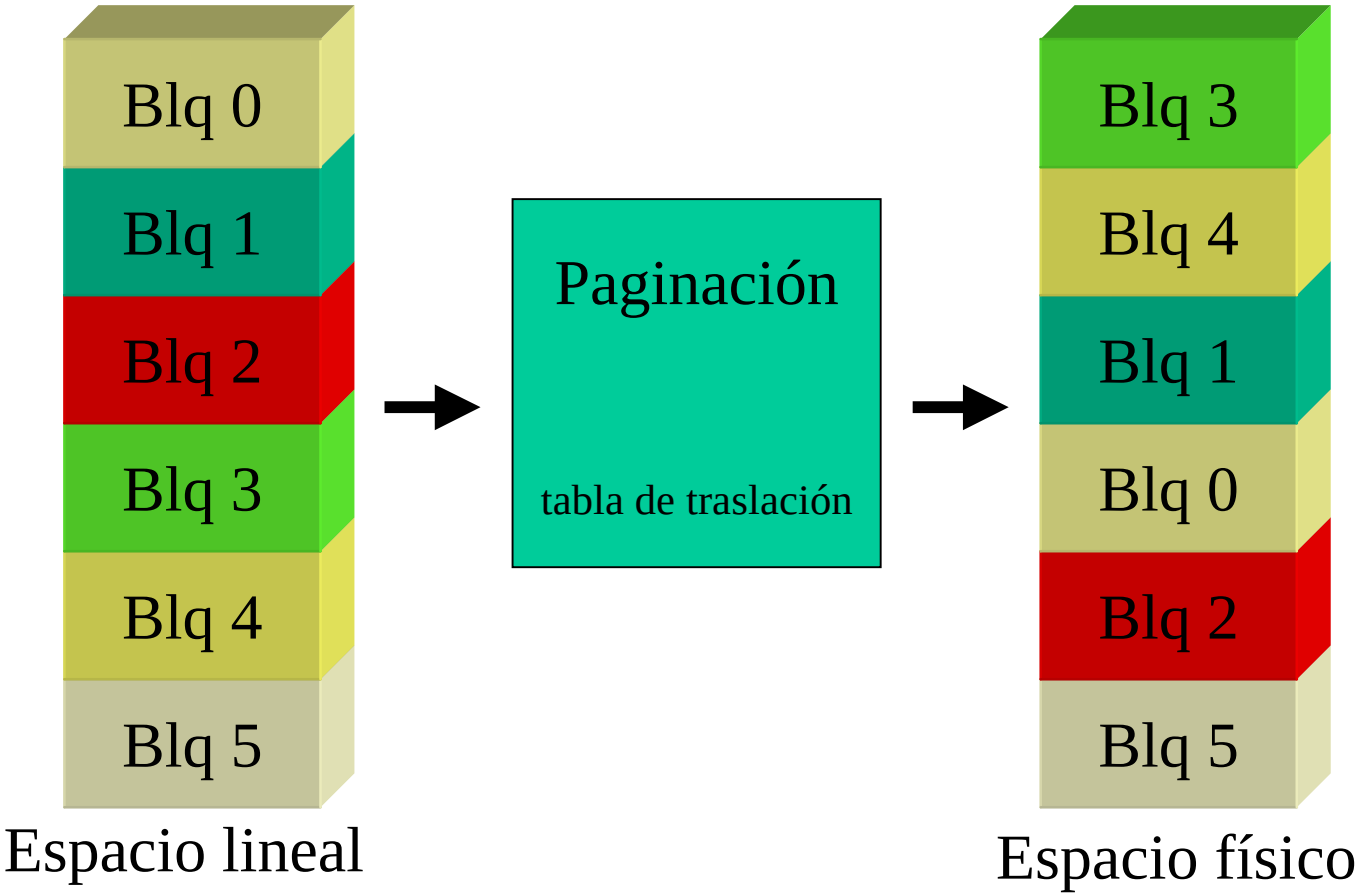
P: presencia
DPL: nivel de privilegio del descriptor
DT: segmentos de memoria o del sistema

Formatos de los descriptores de segmento

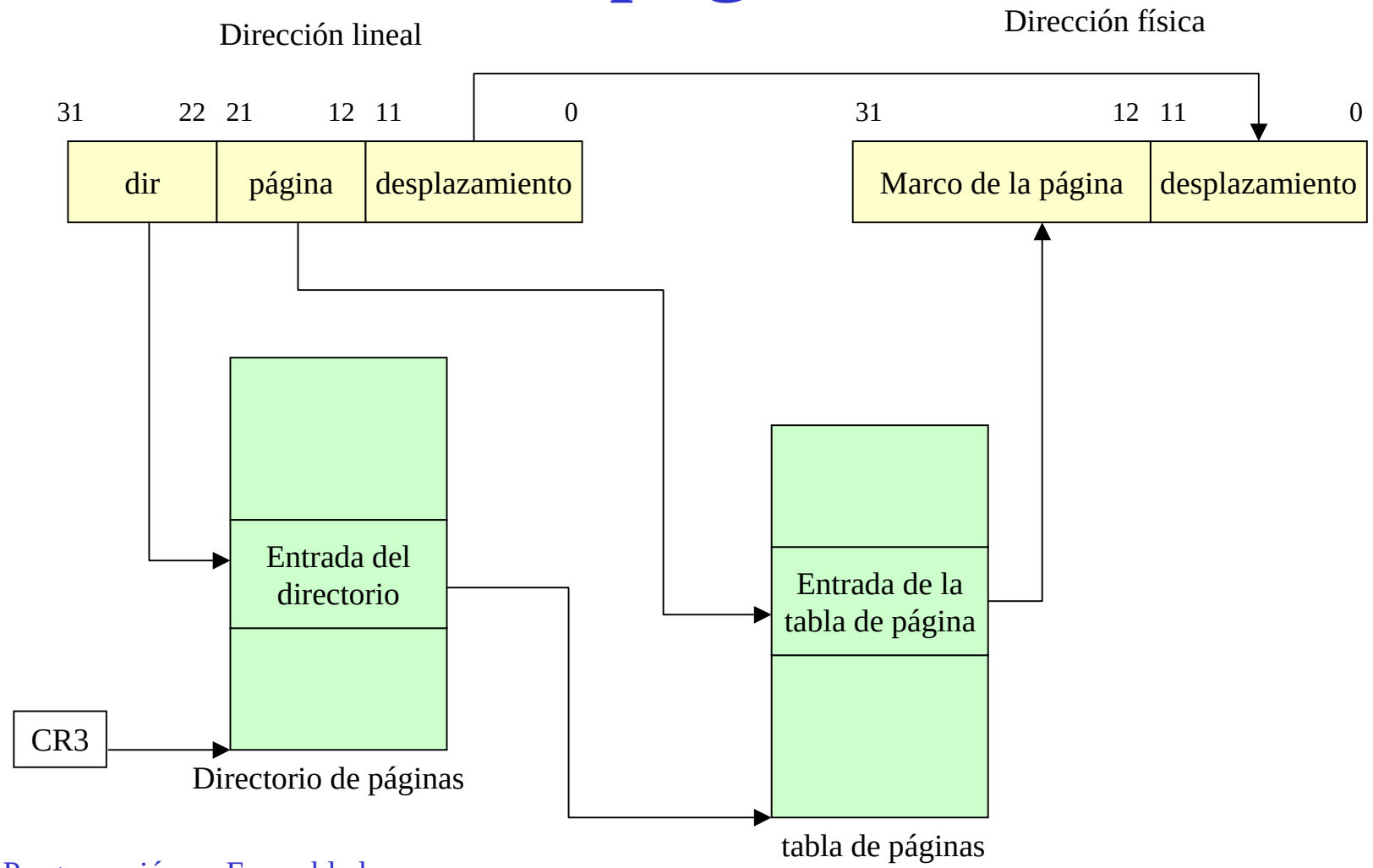
Campo de tipo:

0	Solo lectura
1	Solo lectura, accedido
2	Lectura/escritura
3	Lectura/escritura, accedido
4	Solo lectura, límite expandible hacia abajo
5	Solo lectura, límite expandible hacia abajo, accedido
6	Lectura/escritura, límite expandible hacia abajo
7	Lectura/escritura, límite expandible hacia abajo, accedido
8	Solo ejecución
9	Solo ejecución, accedido
A	Ejecución/lectura
B	Ejecución/lectura, accedido
C	Solo ejecución, de conformidad
D	Solo ejecución, de conformidad, accedido
E	Ejecución/lectura, de conformidad
F	Ejecución/lectura, de conformidad, accedido

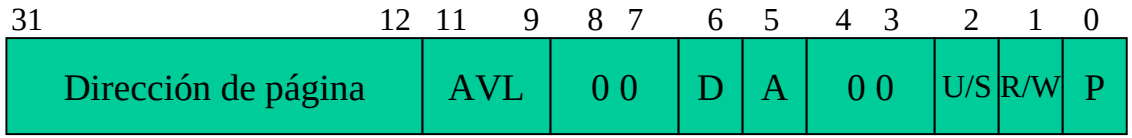
Mecanismo de paginación



Mecanismo de paginación



Formato de los descriptores de página



P: presente
R/W: read/write
U/S: usuario o supervisor
A: accedido
D: sucio (dirty)
AVL: disponible para el software

Mecanismos de protección

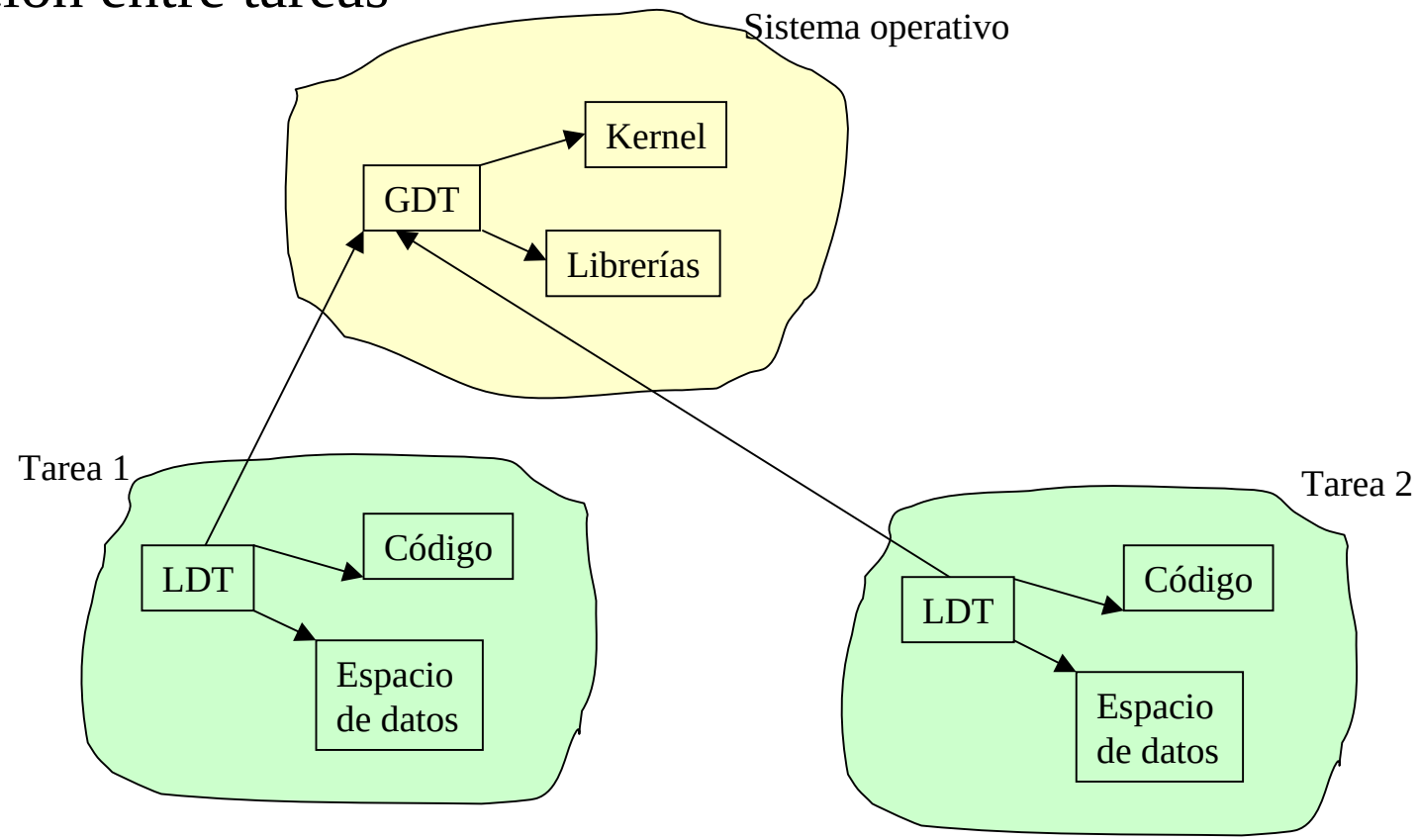
2 tipos

- Protección entre tareas:
  - Se asigna a cada tarea un espacio de direcciones virtual diferente
  - Cada tarea tiene una tabla local de descriptores LDT
  - El sistema operativo se mapea en una tabla global GDT

- Protección dentro de una tarea
  - Cuatro niveles de privilegio de acceso
  - se restringe el acceso a los datos según la sensibilidad del proceso

Mecanismos de protección

Protección entre tareas



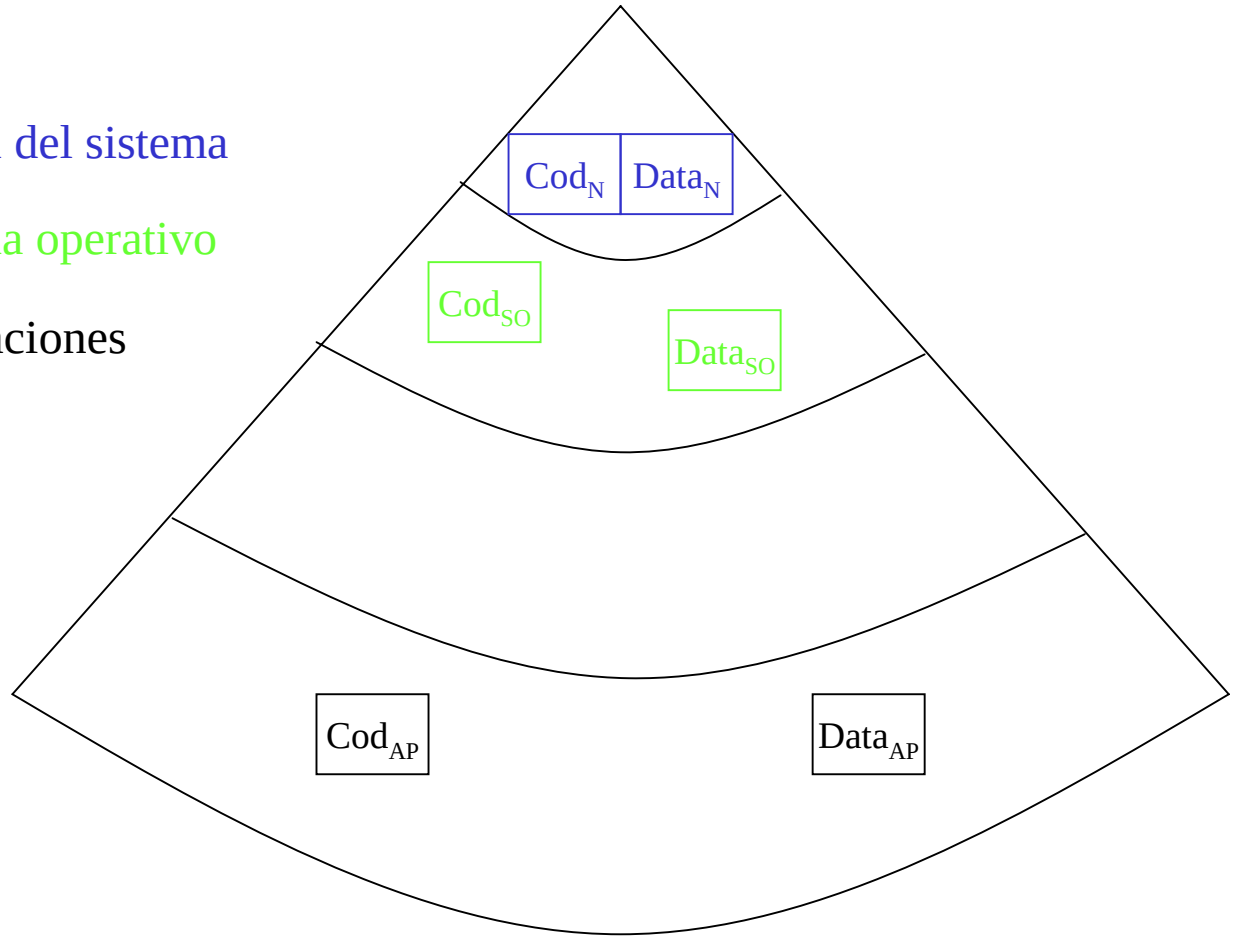
Mecanismos de protección

Protección dentro de una tarea

Kernel del sistema

Sistema operativo

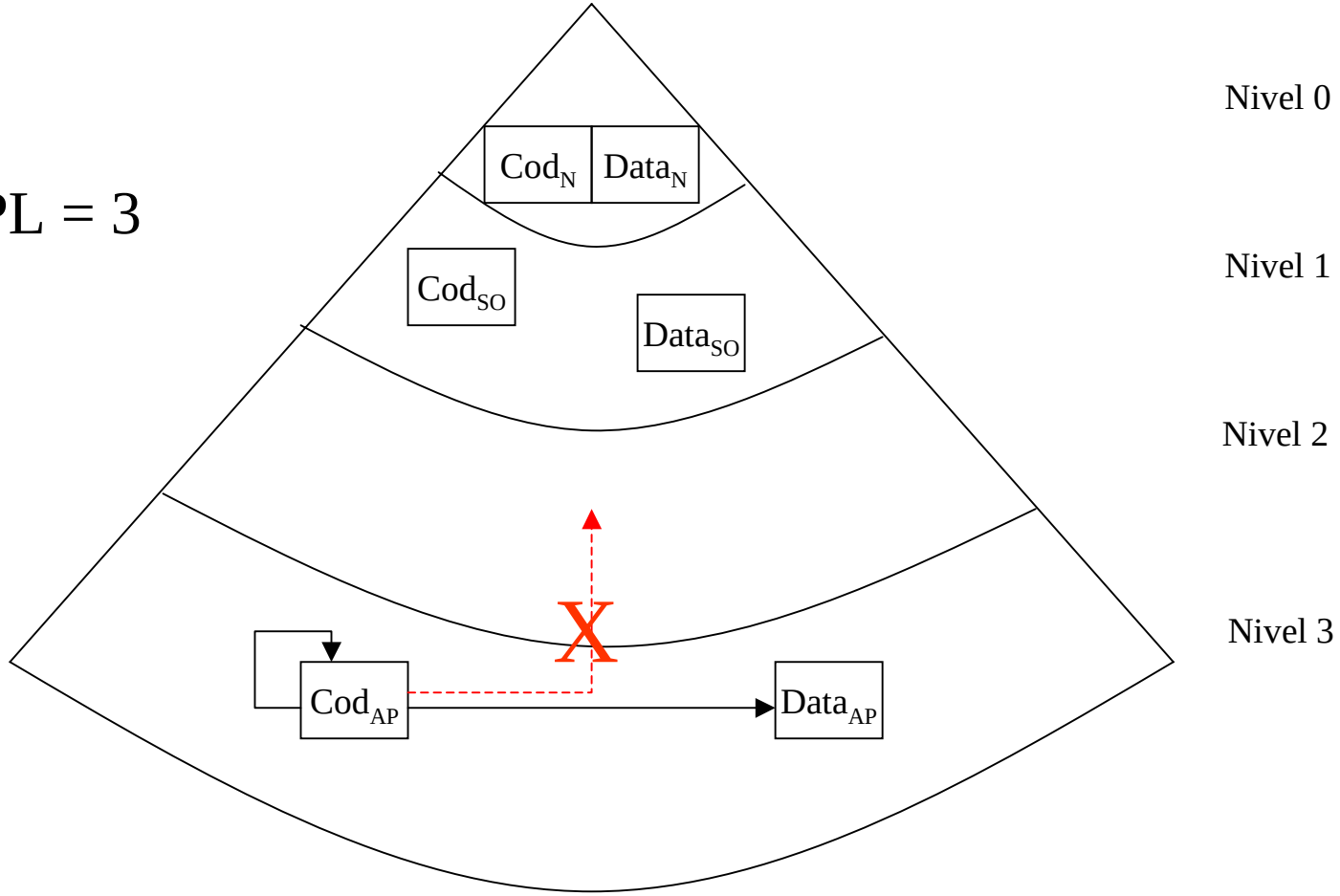
Aplicaciones



Mecanismos de protección

Protección dentro de una tarea

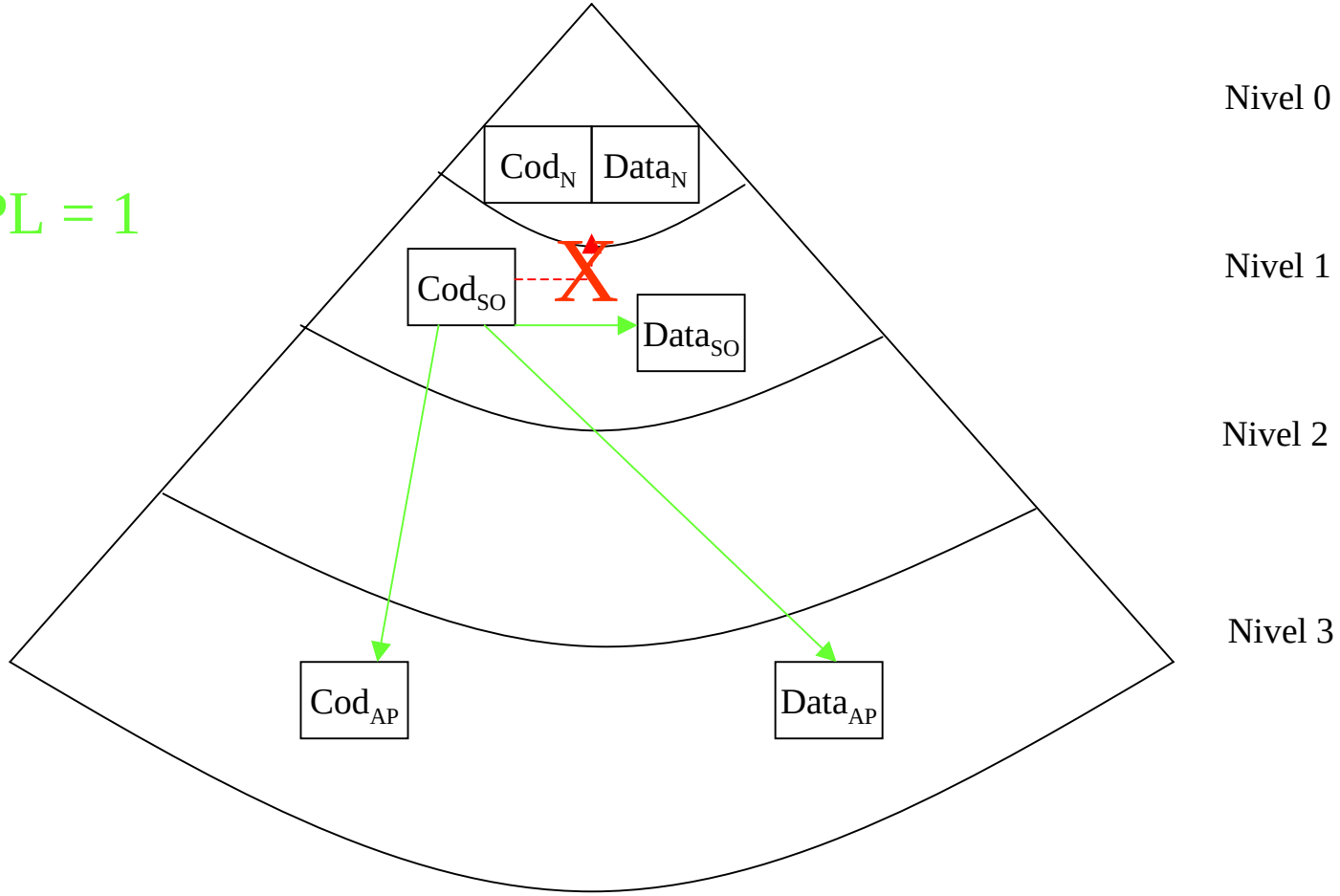
CPL = 3



Mecanismos de protección

Protección dentro de una tarea

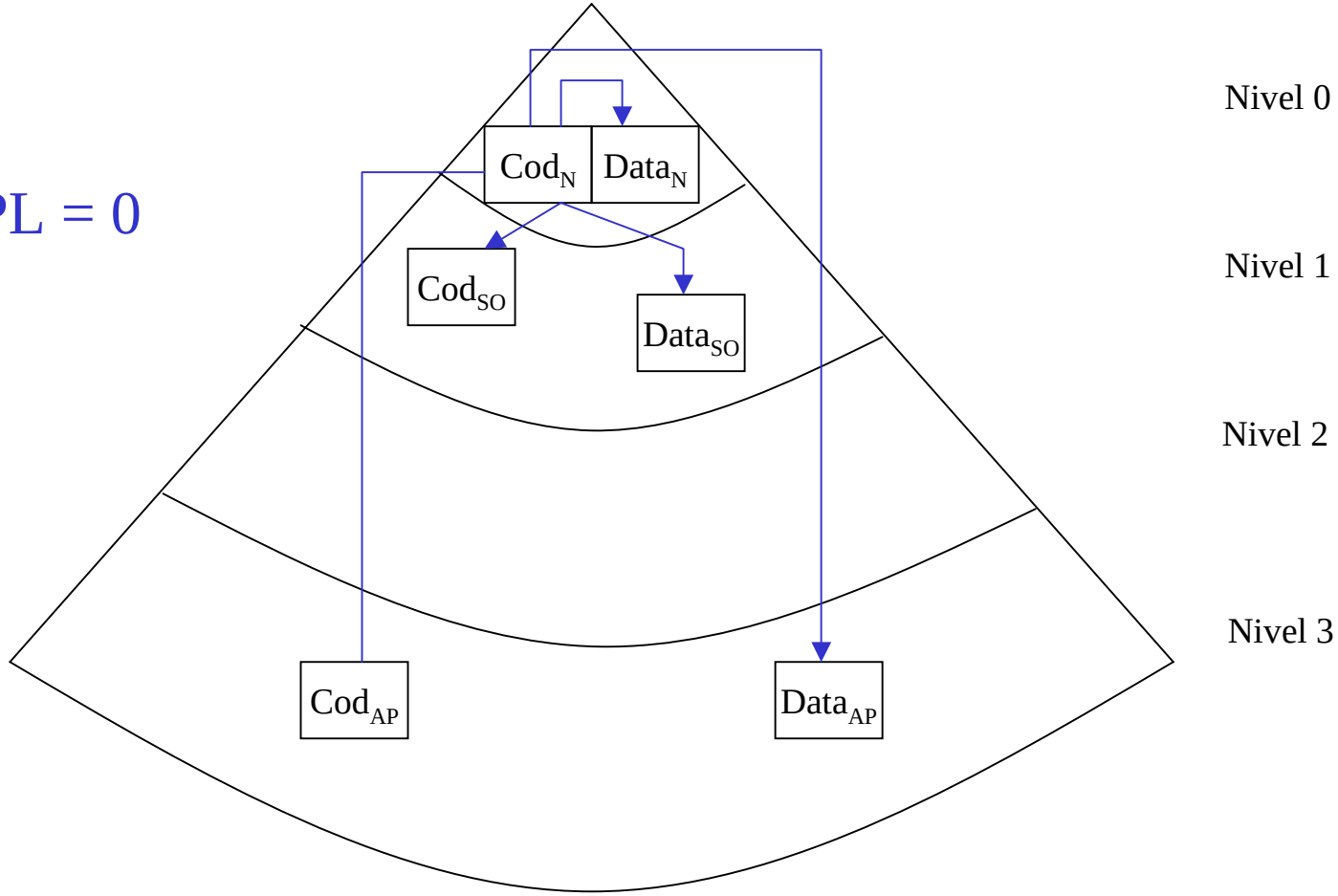
CPL = 1



Mecanismos de protección

Protección dentro de una tarea

CPL = 0



Selección de niveles de privilegio

3 indicadores:

- CPL : Current privilege level
  Esta en el campo RPL del selector de segmento CS

- RPL : Request privilege level
  Esta en todos los selectores de segmento

- DPL : Data privilege level
  En el campo de atributos de cada descriptor de segmento

Selección de niveles de privilegio

En todo acceso a un segmento se verifica:

si CPL < DPL => acceso permitido
sino error

(si el privilegio de ejecución actual es mayor que el del segmento accedido)

si RPL <> CPL => nuevo CPL = max(CPL , RPL)

(si el privilegio de ejecución requerido es menor que el actual entonces el CPL se debilita para adaptarse al nuevo privilegio)

Hagamos un ejemplo

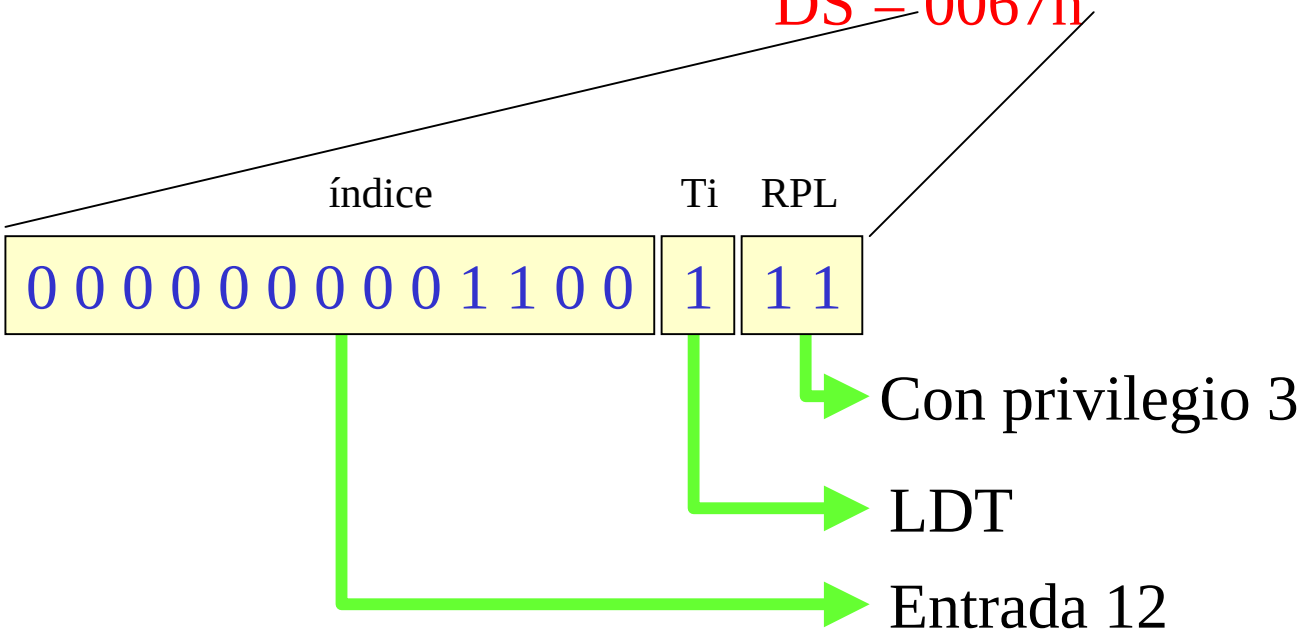
Supongamos la instrucción

mov ax, [bx]

Para este caso

BX = 00000567h

DS = 0067h



Hagamos un ejemplo

La entrada 12 de la LDT tiene:

Base = 12340000h
Límite = 2
Granularidad = 1 (página)
DPL = 3

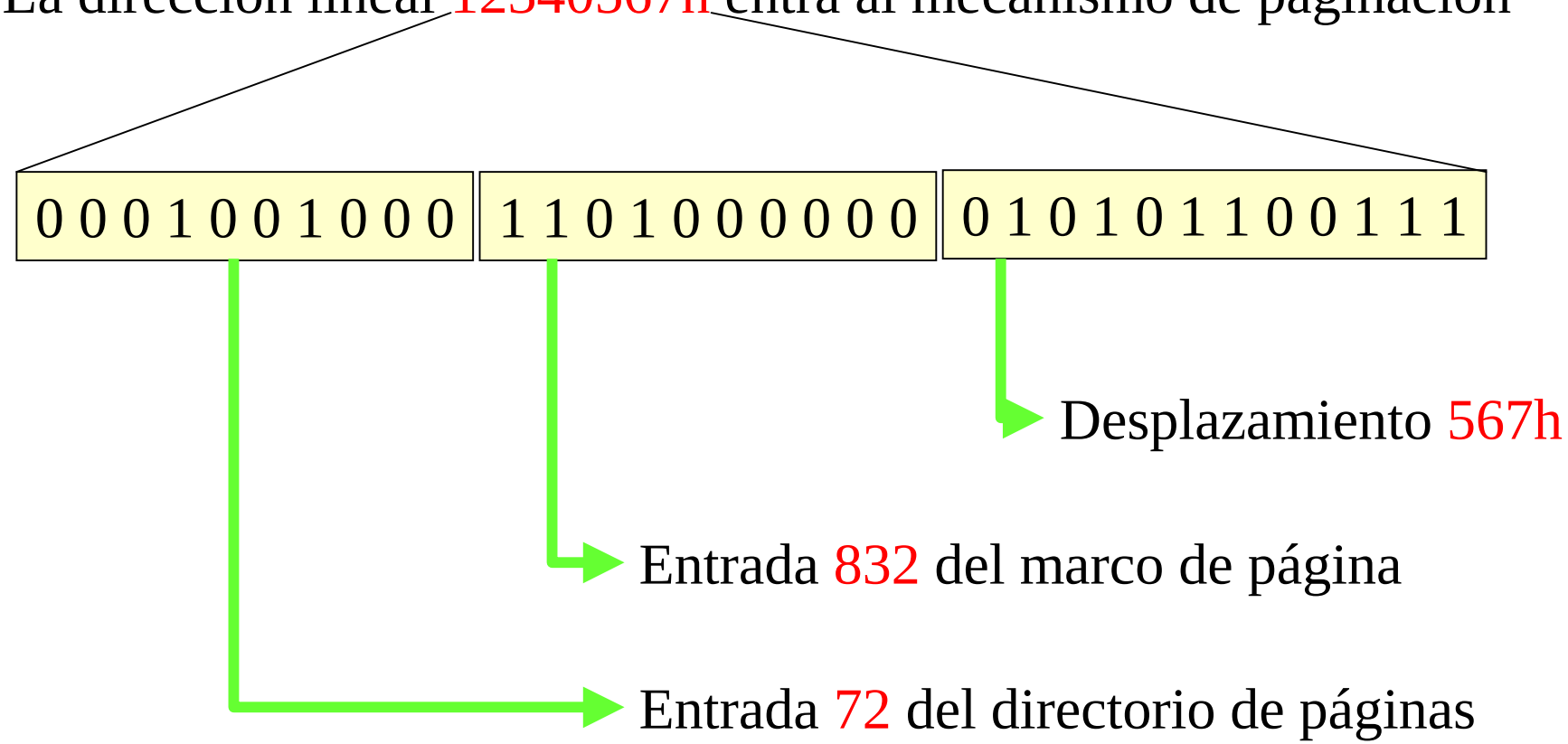
Entonces:

1) si granularidad=1 => límite=8192 bytes
2) comparo BX con el límite: 567 < 8192
3) RPL = DPL => acceso permitido
4) direccción lineal igual a base + offset

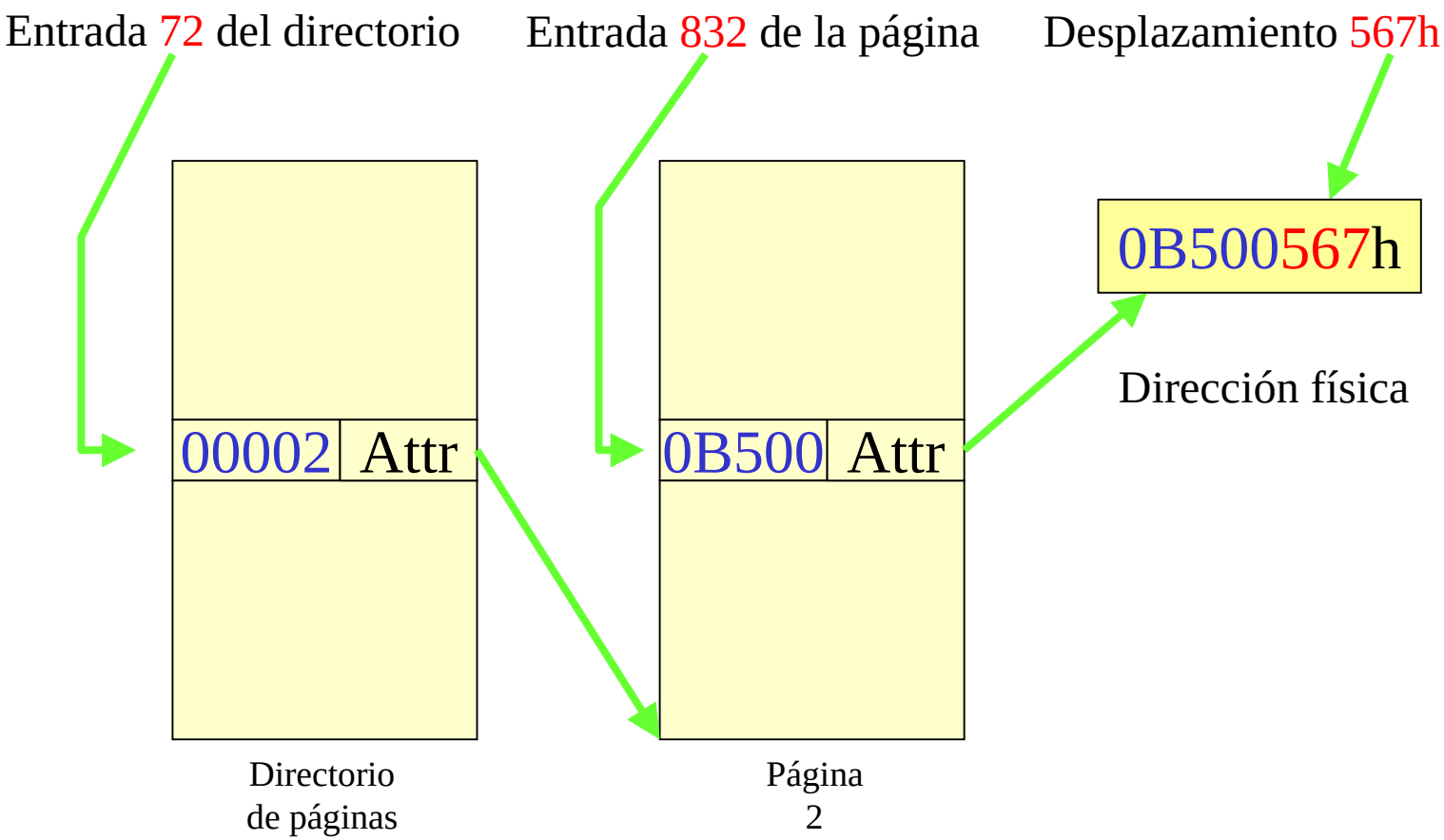
12340000h + 00000567h = 12340567h

Hagamos un ejemplo

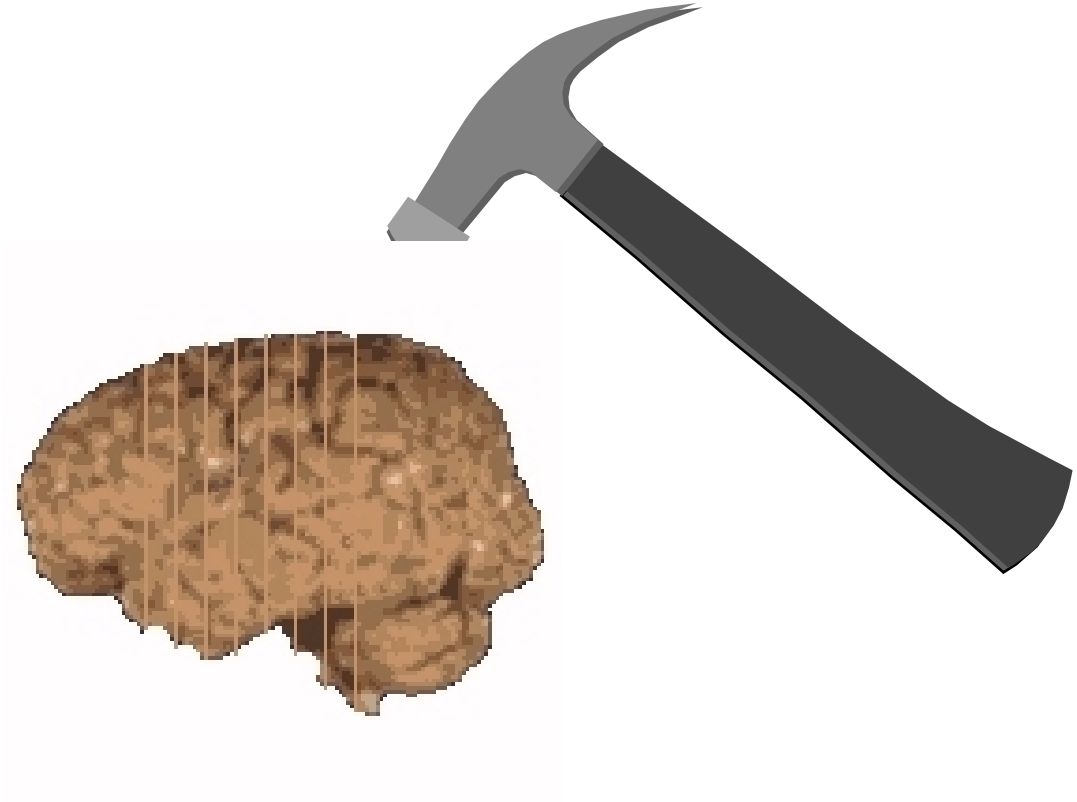
La dirección lineal 12340567h entra al mecanismo de paginación



Hagamos un ejemplo



Un minuto de descanso para el cerebro

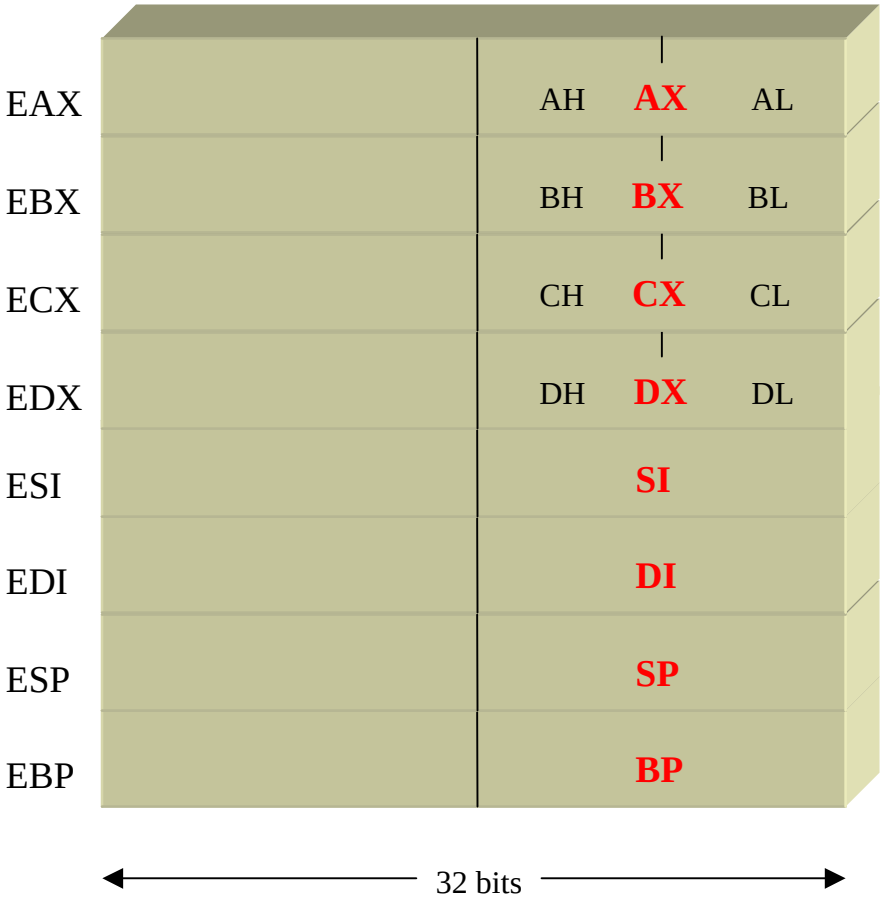


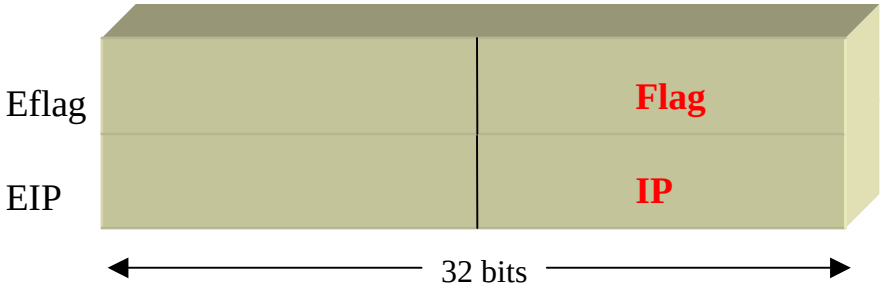
Programación en Ensamblador
Ing. Marcelo Tosini - 2001

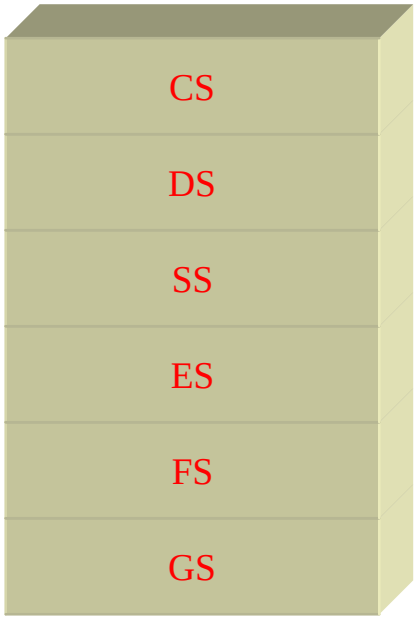
Juego de registros del 386

- Los 10 del 8086 pero de 32 bits
  - 4 generales: EAX, EBX, ECX, EDX
  - 2 índices: ESI, EDI
  - 2 punteros: ESP, EBP
  - 1 estado: EFlag
  - 1 contador de programa: EIP
- Selectores de segmento de 16 bits
  - 6 segmentos: DS, CS, ES, SS, FS, GS
- 16 nuevos registros
  - 4 registros de control: CR0..CR3
  - base GDT: GDTR
  - base LDT: LDTR
  - base vector interrupciones: IDTR
  - base desc. tareas: TR
  - 8 de depuración: DR0..DR7
- 8 registros de sombra

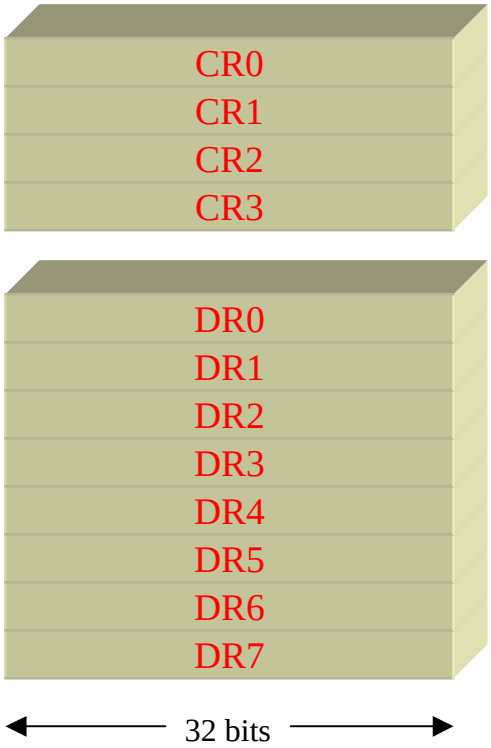
Juego de registros

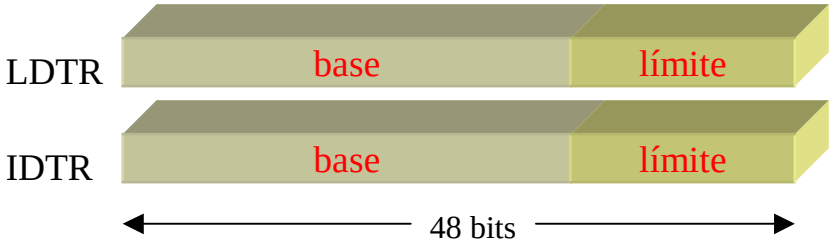


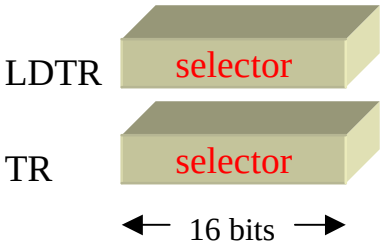


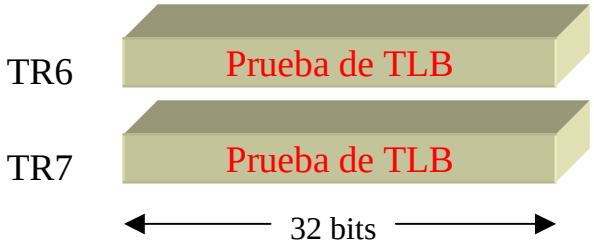


Juego de registros









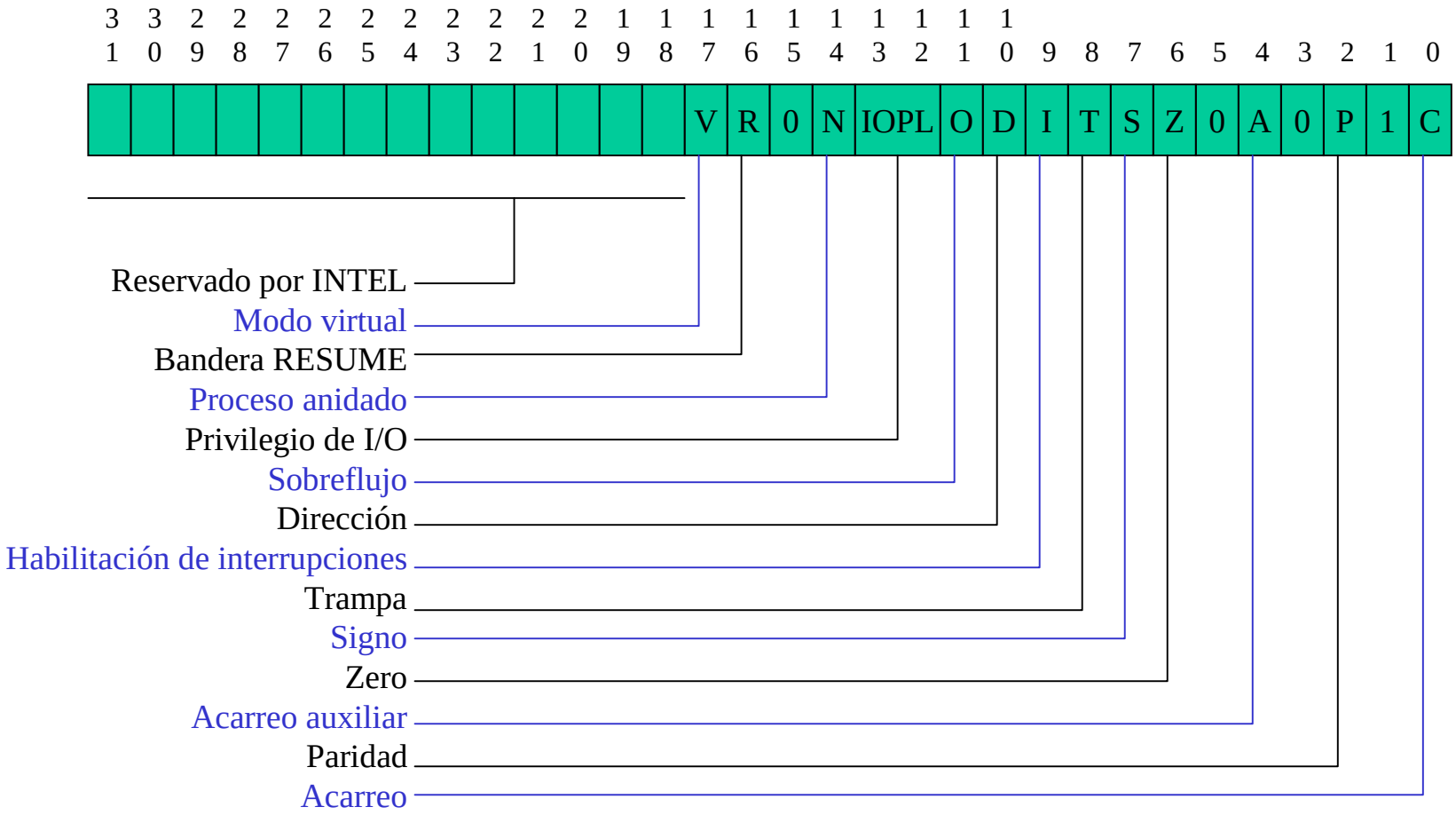
Juego de registros

Registros de sombra

- No accesibles al programador
- Almacenan los descriptores apuntados por los selectores actuales
- Cuando se carga un selector de segmento el procesador carga su registro de sombra correspondiente con el descriptor al que apunta

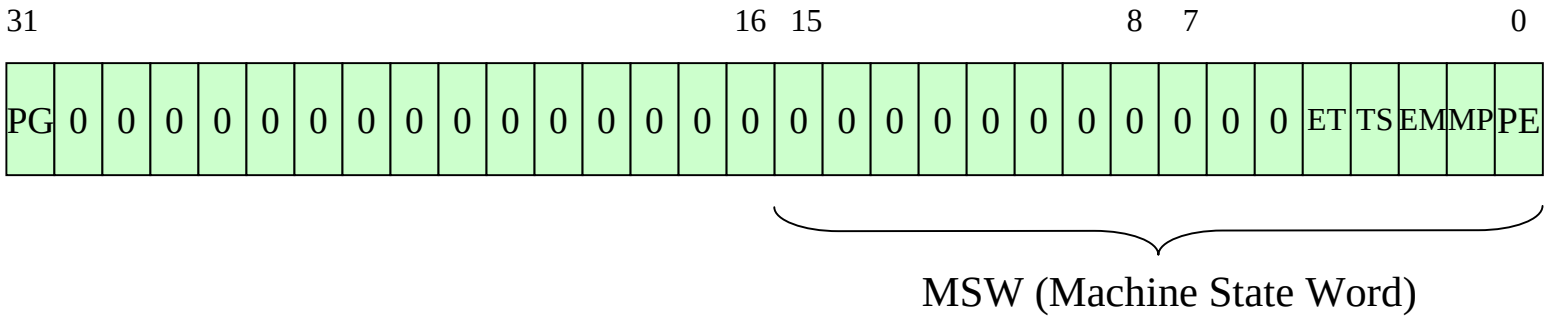
CS	Base	Límite	Atributos
DS	Base	Límite	Atributos
SS	Base	Límite	Atributos
ES	Base	Límite	Atributos
FS	Base	Límite	Atributos
GS	Base	Límite	Atributos
LDTR	Base	Límite	Atributos
TR	Base	Límite	Atributos

Registro de flags



Registros de control (CR0..CR3)

CR0 : Doble palabra de estado de la máquina



PE : indicador de modo protegido/real
MP : coprocesador presente
EM : igual a 1 indica emulación de coprocesador
TS : tarea conmutada. Igual a 1 al producirse una conmutación de tarea
ET : determina tipo de coprocesador. 0 = 287 - 1=387. En el 486 no existe
PG : habilita el mecanismo de paginación

Registros de control (CR0..CR3)

CR1 : No utilizado para 386 y 486

CR2 : dirección lineal del fallo de página

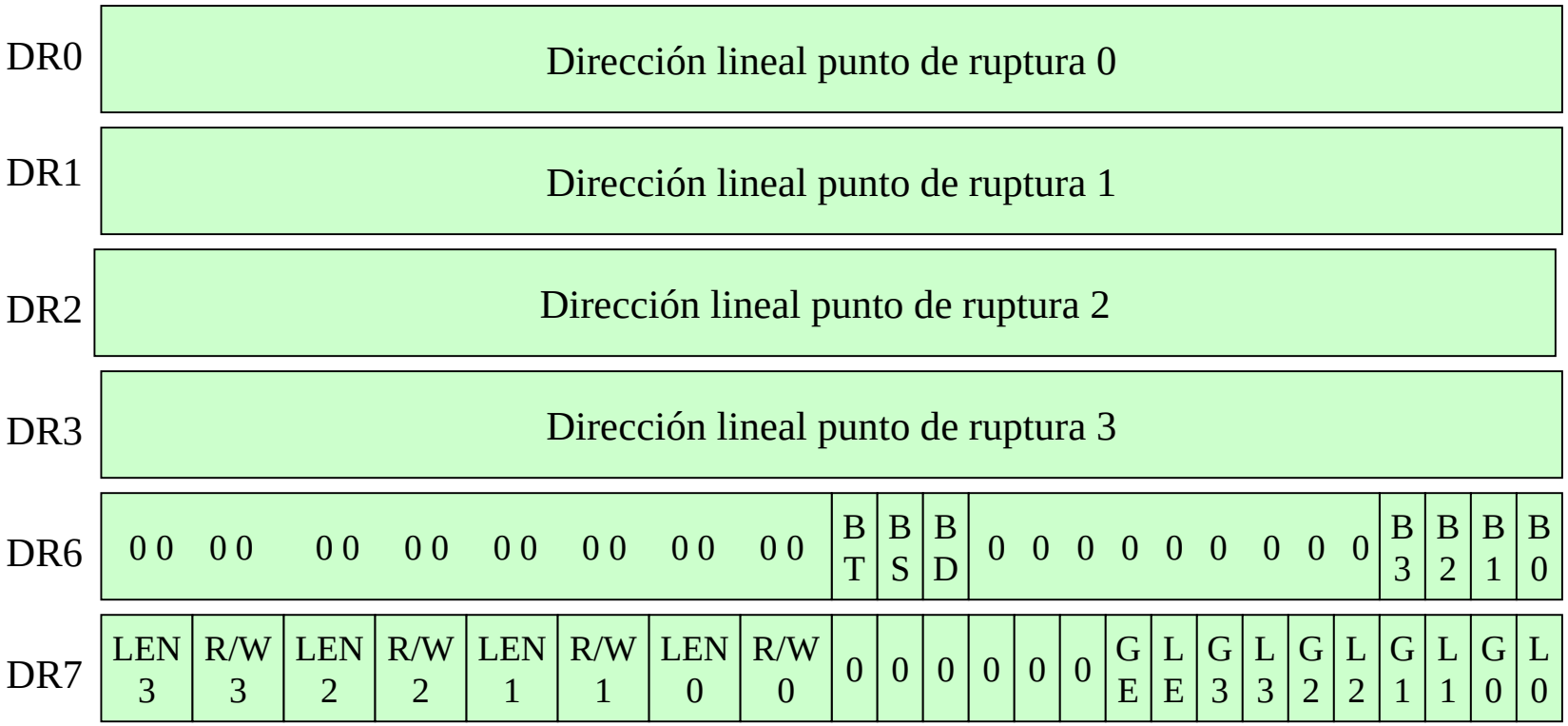
Se almacena la dirección lineal que se introdujo en la Unidad de Paginación para traducirla a dirección física y que ocasionó u error o fallo de página

CR3 : base del directorio de las tablas de páginas

Dirección física en la que comienza el directorio de las tablas de páginas de la tarea en curso

Dirección lineal de fallo de página		CR2
Base del directorio de páginas	000000000000	CR3

Registros de depuración (DR0..DR7)



DR4 y DR5 son reservados

Registro DR7 (registro de control)

DR7	LEN 3	R/W 3	LEN 2	R/W 2	LEN 1	R/W 1	LEN 0	R/W 0	0	0	0	0	0	0	G E	L E	G 3	L 3	G 2	L 2	G 1	L 1	G 0	L 0

R/Wn

00 : ruptura en ejecución de una instrucción
01 : ruptura en escritura de datos
10 : no usado
11 : ruptura en escritura o lectura de datos pero no en búsqueda de instr.

LENn

00 : longitud de 1 byte
01 : longitud de 2 bytes
10 : no usado
11 : longitud de 4 bytes

Modos de direccionamiento

En general:

BASE	**+**	**INDICE * escala**	**+**	**DESP**
Ninguno		Ninguno		
EAX ECX EDX EBX ESP[1] EBP[1] ESI EDI	+	EAX ECX EDX EBX ___[2] EBP ESI EDI	* 1 2 4 8	+ Ninguno 8 bits 32 bits

1 SS es el segmento por defecto para ESP o EBP
2 ESP no puede usarse cono registro índice

Grupos de instrucciones

- 25 de transferencia de datos (14)
- 23 aritméticas (20)
- 18 de manipulación de bits (10)
- 7 de cadenas (5)
- 37 de transferencia de programa (29)
- 28 de control del procesador (11)

49 instrucciones
más que el 8086!!!

Transferencia de datos

LFS	carga FS y registro de 16 bits con los datos de memoria de 32 bits
LGS	carga GS y registro de 16 bits con los datos de memoria de 32 bits
LSS	carga SS y registro de 16 bits con los datos de memoria de 32 bits
POPA	recupera todos los registros de la pila
POPAD	recupera todos los registros de doble pila
POPD	recupera una palabra doble de la pila
POPFD	recupera los indicadores ampliados de la pila
PUSHA	salva todos los registros en la pila
PUSHAD	salva todos los registros de dobles palabras en la pila
PUSHD	salva doble palabra en la pila
PUSHFD	salva banderas ampliadas en la pila

Instrucciones aritméticas

CDQ	convierte doble palabra a cuadruple palabra
MOVSX	cargar, ampliar y poner signo a los datos
MOVZX	cargar y ampliar datos con ceros

Manipulación de bits

BSF	rastrear bits hacia el frente
BSR	rastrear bits hacia atras
BT	instrucción para prueba de bit
BTC	probar bit y complementarlo
BTR	probar bit y resetearlo
BTS	probar bit y setearlo
SHLD	corrimiento a la izquierda en precisión doble
SHRD	corrimiento a la derecha en precisión doble

Cadenas

INS	meter datos de I/O a la memoria
OUTS	sacar datos de la memoria al espacio de I/O

Transferencia de programa

BOUND	comprobación de límite
ENTER	entrar al procedimiento
IRETD	retornar de una interrupción
LEAVE	abandonar el procedimiento
LOOPD	repite el ciclo ECX veces
LOOPED	repite el ciclo mientras sea igual (ECX = contador)
LOOPNED	repite el ciclo mientras no sea igual (ECX = contador)
JECZX	salto si ECX es cero

Control del procesador

ARPL	ajusta el grado solicitado de privilegio
CTS	borra bandera de conmutación de tarea
ESC	instrucción para el coprocesador
LAR	carga derechos de acceso
LGDT	carga de tabla de descriptores globales
LIDT	carga de tabla descriptores de interrupción
LLDT	carga de tabla de descriptores locales
LSL	carga límite de segmento
LTR	carga registro de tarea
SGDT	almacena tabla de registros de descriptores globales
SIDT	almacena tabla de registros de descriptores de interrupción
SLDT	almacena tabla de registros de descriptores locales
STR	almacena registro de tarea
VERR	verificar acceso para lectura
VERW	verificar acceso para escritura