

# Protección

## Niveles de privilegio y verificación de permisos

### Programación de Sistemas Operativos

David Alejandro González Márquez

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

# Introducción

- El objetivo de los mecanismos de protección en el procesador es:
  - Restringir instrucciones.

# Introducción

- El objetivo de los mecanismos de protección en el procesador es:
  - Restringir instrucciones.
  - Limitar el acceso a la memoria.

# Introducción

- El objetivo de los mecanismos de protección en el procesador es:
  - Restringir instrucciones.
  - Limitar el acceso a la memoria.
  - Limitar acciones sobre recursos.

# Introducción

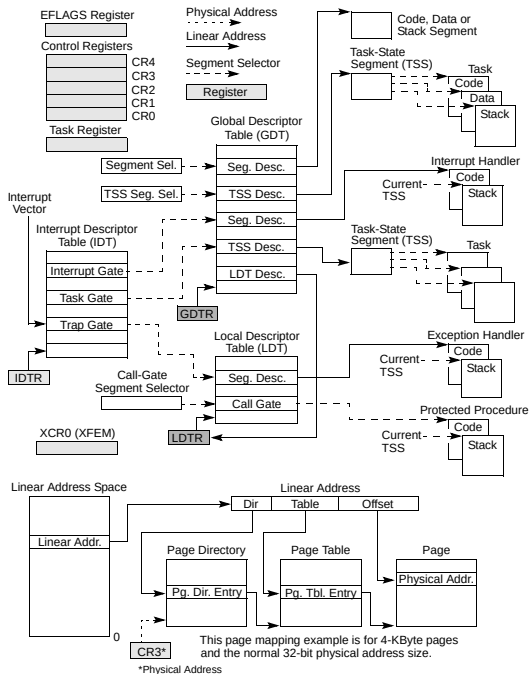
- El objetivo de los mecanismos de protección en el procesador es:
  - Restringir instrucciones.
  - Limitar el acceso a la memoria.
  - Limitar acciones sobre recursos.
- En todos los sistemas operativos modernos se requiere minimamente de dos niveles de protección: **Usuario** y **Supervisor**

# Introducción

- El objetivo de los mecanismos de protección en el procesador es:
  - Restringir instrucciones.
  - Limitar el acceso a la memoria.
  - Limitar acciones sobre recursos.
- En todos los sistemas operativos modernos se requiere minimamente de dos niveles de protección: **Usuario** y **Supervisor**

Vamos a ver protección a nivel de **Segmentación**, **Paginación**, **Tareas** e **Interrupciones**.

# ¿Donde configurar privilegios?



# ¿Donde configurar privilegios?

## - DPL - Descriptor Privilege Level:

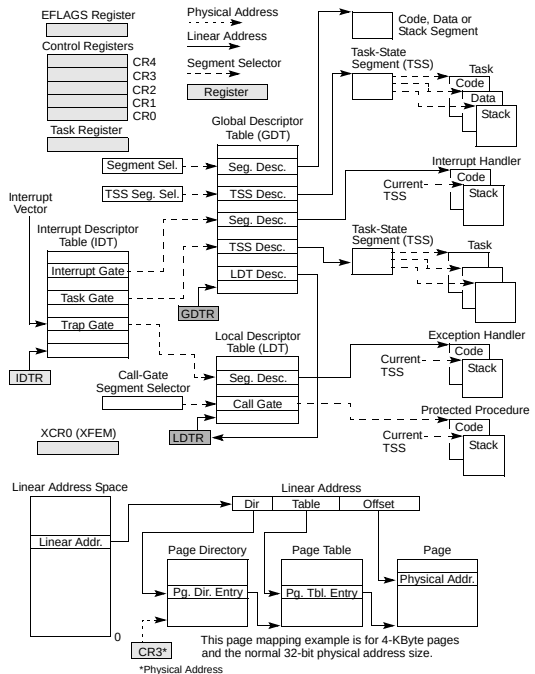
Parte de todas las entradas en tablas que describen memoria o estructuras del procesador.

## - RPL - Requested Privilege Level:

Parte de todos los selectores, permiten identificar descriptores.

## - U/S - User/Supervisor:

Indican el nivel de privilegio en paginación.





# ¿Donde configurar privilegios?

## - DPL - Descriptor Privilege Level:

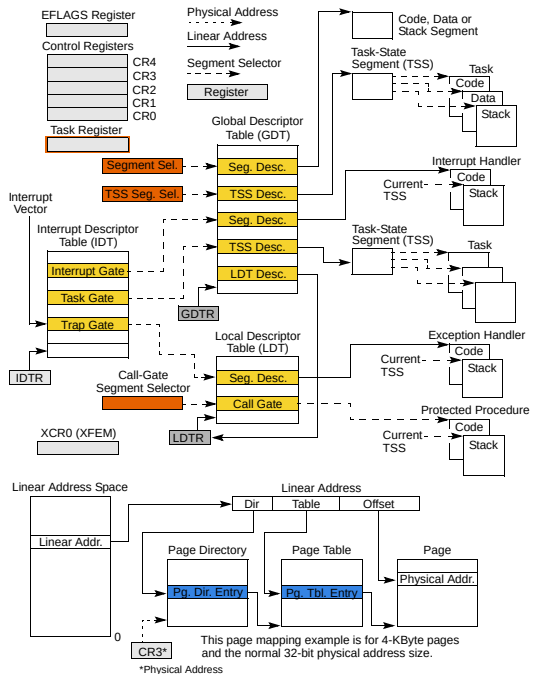
Parte de todas las entradas en tablas que describen memoria o estructuras del procesador.

## - RPL - Requested Privilege Level:

Parte de todos los selectores, permiten identificar descriptores.

## - U/S - User/Supervisor:

Indican el nivel de privilegio en paginación.



## Acceso a memoria

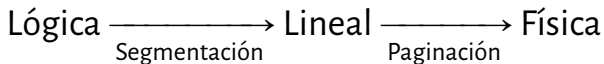
- Las unidades de segmentación y paginación transforman direcciones de memoria.

## Acceso a memoria

- Las unidades de segmentación y paginación transforman direcciones de memoria.
- Realizan las verificaciones de protección, determinando si el acceso puede o no realizarse.

## Acceso a memoria

- Las unidades de segmentación y paginación transforman direcciones de memoria.
- Realizan las verificaciones de protección, determinando si el acceso puede o no realizarse.



- Lógica: **selector:offset** (16:32 bits)
- Lineal: **offset** (32 bits)
- Física: **offset** (32 bits)

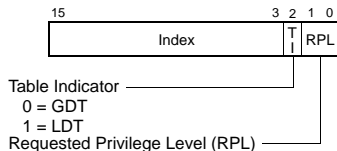
Nota: Suponiendo paginación no PAE, con paginas de 4k.

# Segmentación

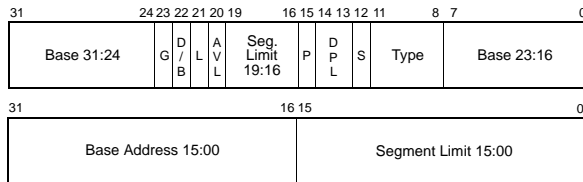
Lógica  $\xrightarrow{\text{Segmentación}}$  Lineal  $\xrightarrow{\text{Paginación}}$  Física

Estructuras necesarias para administrar segmentación.

## Selector de segmento

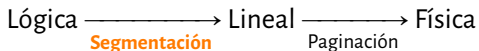


## Descriptor GDT



- L — 64-bit code segment (IA-32e mode only)
- AVL — Available for use by system software
- BASE — Segment base address
- D/B — Default operation size (0 = 16-bit segment; 1 = 32-bit segment)
- DPL — Descriptor privilege level
- G — Granularity
- LIMIT — Segment Limit
- P — Segment present
- S — Descriptor type (0 = system; 1 = code or data)
- TYPE — Segment type

# Segmentación

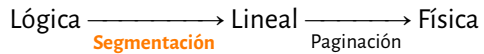


## Definiciones:

- **DPL (Descriptor Privilege Level):**  
Nivel de privilegio del segmento para ser accedido (se ubica en su descriptor).
- **RPL (Requested Privilege Level):**  
Nivel de privilegio indicado en el selector de segmento como solicitud de acceso.
- **CPL (Current Privilege Level):**  
El DPL del segmento de código que se está ejecutando. Es nuestro nivel de privilegio.
- **EPL (Effective Privilege Level):**  
Máximo numérico entre el CPL y el RPL, es decir,  $EPL = \max(CPL, RPL)$

# Segmentación

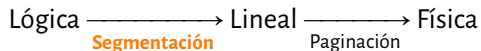
Verificación direcciones → `selector:offset`



# Segmentación

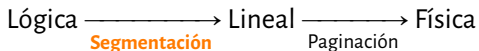
Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente (P=1).





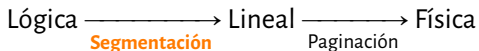
# Segmentación



Verificación direcciones  $\rightarrow$  `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$

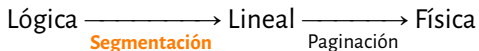
# Segmentación



Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos →  $EPL \leq DPL$

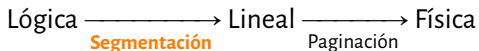
# Segmentación



Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos →  $EPL \leq DPL$
  - Si es un segmento de código:
    - Non-conforming →  $CPL = DPL$
    - Conforming →  $CPL \geq DPL$

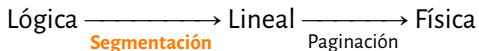
# Segmentación



Verificación direcciones  $\rightarrow$  `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos  $\rightarrow EPL \leq DPL$
  - Si es un segmento de código:
    - Non-conforming  $\rightarrow CPL = DPL$
    - Conforming  $\rightarrow CPL \geq DPL$
4. Verificar acción a realizar

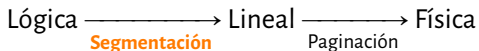
# Segmentación



Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos →  $EPL \leq DPL$
  - Si es un segmento de código:
    - Non-conforming →  $CPL = DPL$
    - Conforming →  $CPL \geq DPL$
4. Verificar acción a realizar
  - Leer: Cualquier segmento de datos, restringido en segmentos de código.

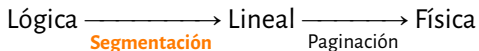
# Segmentación



Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos →  $EPL \leq DPL$
  - Si es un segmento de código:
    - Non-conforming →  $CPL = DPL$
    - Conforming →  $CPL \geq DPL$
4. Verificar acción a realizar
  - Leer: Cualquier segmento de datos, restringido en segmentos de código.
  - Escribir: Restringido en segmentos de datos, prohibido en segmentos de código.

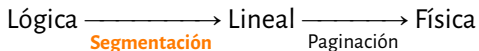
# Segmentación



Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos →  $EPL \leq DPL$
  - Si es un segmento de código:
    - Non-conforming →  $CPL = DPL$
    - Conforming →  $CPL \geq DPL$
4. Verificar acción a realizar
  - Leer: Cualquier segmento de datos, restringido en segmentos de código.
  - Escribir: Restringido en segmentos de datos, prohibido en segmentos de código.
  - Ejecutar: Permitido solo en segmentos de código.

# Segmentación



Verificación direcciones → `selector:offset`

1. Verificar que el segmento este presente ( $P=1$ ).
2. Verificar límite del segmento.
  - Si  $G=0$ :  $\text{offset} < (\text{segment.limit}+1)$
  - Si  $G=1$ :  $\text{offset} < (\text{segment.limit}+1) * 0x1000$
3. Verificar el nivel para acceder al segmento. (Las comparaciones  $=$ ,  $\geq$ ,  $\leq$  son por valor numérico)
  - Si el segmento es de datos →  $EPL \leq DPL$
  - Si es un segmento de código:
    - Non-conforming →  $CPL = DPL$
    - Conforming →  $CPL \geq DPL$
4. Verificar acción a realizar
  - Leer: Cualquier segmento de datos, restringido en segmentos de código.
  - Escribir: Restringido en segmentos de datos, prohibido en segmentos de código.
  - Ejecutar: Permitido solo en segmentos de código.

Si alguna de las verificaciones falla: → General Protection Fault (#GP)



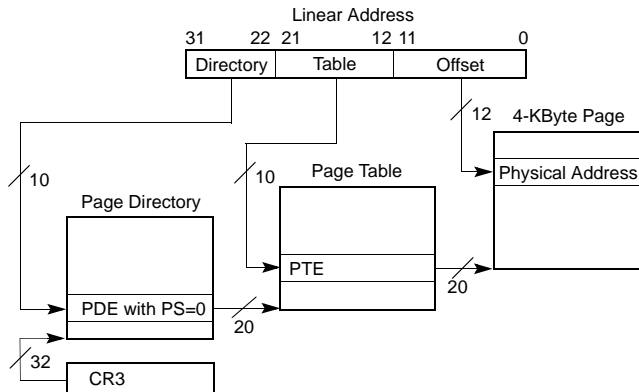
## Tipos de segmentos de código o datos

Type Field					Descriptor Type	Description
Decimal	11	10 E	9 W	8 A		
0	0	0	0	0	Data	Read-Only
1	0	0	0	1	Data	Read-Only, accessed
2	0	0	1	0	Data	Read/Write
3	0	0	1	1	Data	Read/Write, accessed
4	0	1	0	0	Data	Read-Only, expand-down
5	0	1	0	1	Data	Read-Only, expand-down, accessed
6	0	1	1	0	Data	Read/Write, expand-down
7	0	1	1	1	Data	Read/Write, expand-down, accessed
		<b>C</b>	<b>R</b>	<b>A</b>		
8	1	0	0	0	Code	Execute-Only
9	1	0	0	1	Code	Execute-Only, accessed
10	1	0	1	0	Code	Execute/Read
11	1	0	1	1	Code	Execute/Read, accessed
12	1	1	0	0	Code	Execute-Only, conforming
13	1	1	0	1	Code	Execute-Only, conforming, accessed
14	1	1	1	0	Code	Execute/Read, conforming
15	1	1	1	1	Code	Execute/Read, conforming, accessed

# Paginación

Lógica  $\xrightarrow{\text{Segmentación}}$  Lineal  $\xrightarrow{\text{Paginación}}$  Física

Las direcciones en paginación son resueltas por medio de directorios y tablas de paginas. El nivel de acceso está dado por atributos en las entradas de estas tablas (PDE y PTE).



# Paginación

Lógica  $\xrightarrow{\text{Segmentación}}$  Lineal  $\xrightarrow{\text{Paginación}}$  Física

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

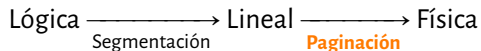
# Paginación

Lógica  $\xrightarrow{\text{Segmentación}}$  Lineal  $\xrightarrow{\text{Paginación}}$  Física

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (PDE . P=1 y PTE . P=1).

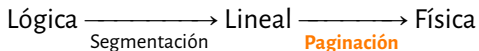
# Paginación



Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (PDE . P=1 y PTE . P=1).
2. Verificar el nivel para acceder a la página.

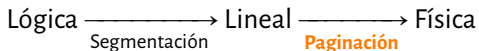
# Paginación



Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE.P=1` y `PTE.P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE.US=0` or `PTE.US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).

# Paginación



Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE.P=1` y `PTE.P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE.US=0` or `PTE.US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE.US=1` and `PTE.US=1` la página tiene permisos de usuario (`CPL=3`).

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE.P=1` y `PTE.P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE.US=0` or `PTE.US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE.US=1` and `PTE.US=1` la página tiene permisos de usuario (`CPL=3`).
3. Verificar acción a realizar



Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE.P=1` y `PTE.P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE.US=0` or `PTE.US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE.US=1` and `PTE.US=1` la página tiene permisos de usuario (`CPL=3`).
3. Verificar acción a realizar
  - Leer: Cualquier pagina puede ser leída.

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE.P=1` y `PTE.P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE.US=0` or `PTE.US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE.US=1` and `PTE.US=1` la página tiene permisos de usuario (`CPL=3`).
3. Verificar acción a realizar
  - Leer: Cualquier pagina puede ser leída.
  - Ejecutar: Cualquier pagina puede ser ejecutada.

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE . P=1` y `PTE . P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE . US=0` or `PTE . US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE . US=1` and `PTE . US=1` la página tiene permisos de usuario (`CPL=3`).
3. Verificar acción a realizar
  - Leer: Cualquier pagina puede ser leída.
  - Ejecutar: Cualquier pagina puede ser ejecutada.
  - Escribir:
    - Si `PDE . RW=0` or `PTE . RW=0` la página no puede ser escrita.

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE . P=1` y `PTE . P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE . US=0` or `PTE . US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE . US=1` and `PTE . US=1` la página tiene permisos de usuario (`CPL=3`).
3. Verificar acción a realizar
  - Leer: Cualquier pagina puede ser leída.
  - Ejecutar: Cualquier pagina puede ser ejecutada.
  - Escribir:
    - Si `PDE . RW=0` or `PTE . RW=0` la página no puede ser escrita.
    - Si `PDE . RW=1` and `PTE . RW=1` la página puede ser escrita.

Verificación direcciones  $\rightarrow$  `linear`  $\rightarrow$  `PDindex:PTindex:offset`

1. Verificar que la pagina este presente (`PDE.P=1` y `PTE.P=1`).
2. Verificar el nivel para acceder a la página.
  - Si `PDE.US=0` or `PTE.US=0` la página tiene permisos de supervisor (`CPL=0` o `1` o `2`).
  - Si `PDE.US=1` and `PTE.US=1` la página tiene permisos de usuario (`CPL=3`).
3. Verificar acción a realizar
  - Leer: Cualquier pagina puede ser leída.
  - Ejecutar: Cualquier pagina puede ser ejecutada.
  - Escribir:
    - Si `PDE.RW=0` or `PTE.RW=0` la página no puede ser escrita.
    - Si `PDE.RW=1` and `PTE.RW=1` la página puede ser escrita.

Si alguna de las verificaciones falla:  $\rightarrow$  Page Fault (#PF)

**Table 5-3. Combined Page-Directory and Page-Table Protection**

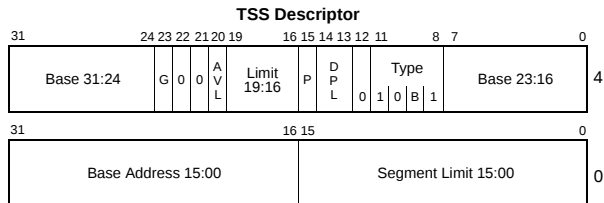
Page-Directory Entry		Page-Table Entry		Combined Effect	
Privilege	Access Type	Privilege	Access Type	Privilege	Access Type
User	Read-Only	User	Read-Only	User	Read-Only
User	Read-Only	User	Read-Write	User	Read-Only
User	Read-Write	User	Read-Only	User	Read-Only
User	Read-Write	User	Read-Write	User	Read/Write
User	Read-Only	Supervisor	Read-Only	Supervisor	Read/Write*
User	Read-Only	Supervisor	Read-Write	Supervisor	Read/Write*
User	Read-Write	Supervisor	Read-Only	Supervisor	Read/Write*
User	Read-Write	Supervisor	Read-Write	Supervisor	Read/Write
Supervisor	Read-Only	User	Read-Only	Supervisor	Read/Write*
Supervisor	Read-Only	User	Read-Write	Supervisor	Read/Write*
Supervisor	Read-Write	User	Read-Only	Supervisor	Read/Write*
Supervisor	Read-Write	User	Read-Write	Supervisor	Read/Write
Supervisor	Read-Only	Supervisor	Read-Only	Supervisor	Read/Write*
Supervisor	Read-Only	Supervisor	Read-Write	Supervisor	Read/Write*
Supervisor	Read-Write	Supervisor	Read-Only	Supervisor	Read/Write*
Supervisor	Read-Write	Supervisor	Read-Write	Supervisor	Read/Write

**NOTE:**

\* If CR0.WP = 1, access type is determined by the R/W flags of the page-directory and page-table entries. IF CR0.WP = 0, supervisor

## Intercambio de tareas

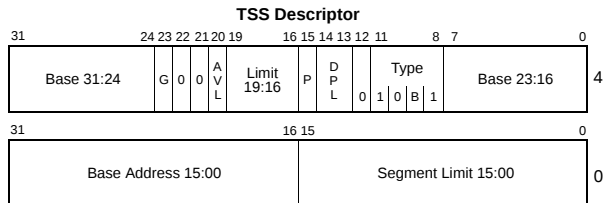
- El DPL de los descriptores de TSS indica el nivel de privilegio necesario para ejecutar la tarea.



AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type

# Intercambio de tareas

- El DPL de los descriptores de TSS indica el nivel de privilegio necesario para ejecutar la tarea.
- Verificar si es posible saltar a una tarea:

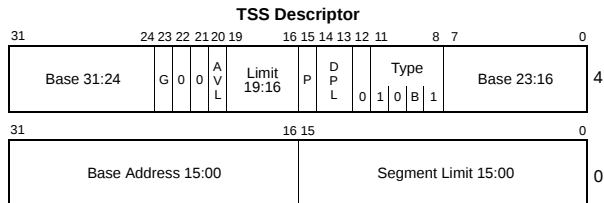


AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type



# Intercambio de tareas

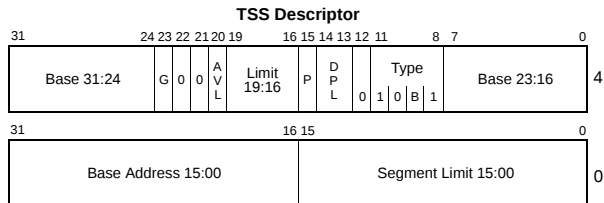
- El DPL de los descriptores de TSS indica el nivel de privilegio necesario para ejecutar la tarea.
- Verificar si es posible saltar a una tarea:
  1. Verificar TSS presente: P=1.



AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type

# Intercambio de tareas

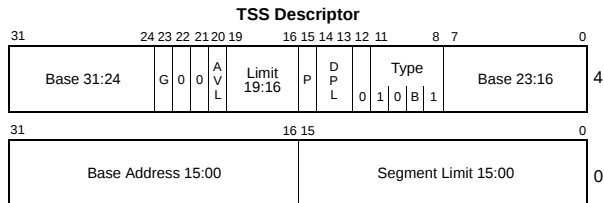
- El DPL de los descriptores de TSS indica el nivel de privilegio necesario para ejecutar la tarea.
- Verificar si es posible saltar a una tarea:
  1. Verificar TSS presente:  $P=1$ .
  2. Verificar privilegios:  $CPL \leq DPL$



AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type

# Intercambio de tareas

- El DPL de los descriptores de TSS indica el nivel de privilegio necesario para ejecutar la tarea.
- Verificar si es posible saltar a una tarea:
  1. Verificar TSS presente:  $P=1$ .
  2. Verificar privilegios:  $CPL \leq DPL$
  3. Verificar bit Busy en cero.



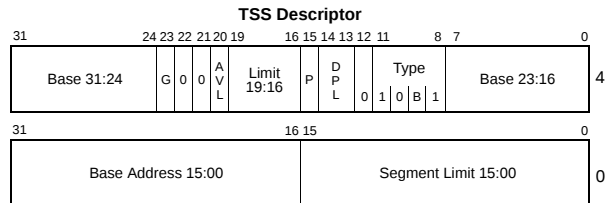
AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type

# Intercambio de tareas

- El DPL de los descriptores de TSS indica el nivel de privilegio necesario para ejecutar la tarea.
- Verificar si es posible saltar a una tarea:
  1. Verificar TSS presente:  $P=1$ .
  2. Verificar privilegios:  $CPL \leq DPL$
  3. Verificar bit Busy en cero.

Si alguna de las verificaciones falla:

→ General Protection Fault (#GP)

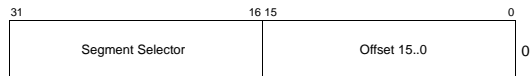
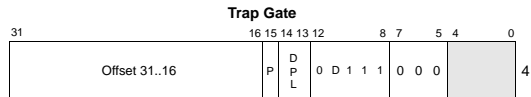
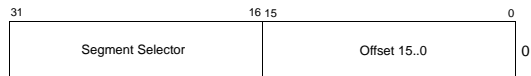
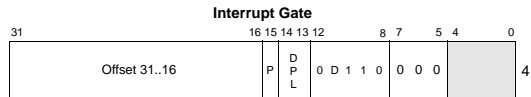
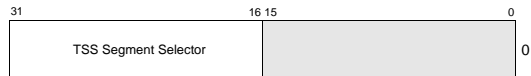
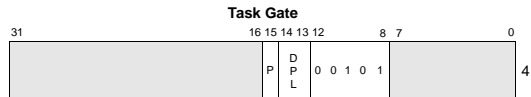


AVL	Available for use by system software
B	Busy flag
BASE	Segment Base Address
DPL	Descriptor Privilege Level
G	Granularity
LIMIT	Segment Limit
P	Segment Present
TYPE	Segment Type

# Interrupciones

## Descriptores de IDT

- Cada entrada de interrupciones tiene su propio DPL.
- Indica el nivel de privilegio necesario para poder acceder a la interrupción.



DPL	Descriptor Privilege Level
Offset	Offset to procedure entry point
P	Segment Present flag
Selector	Segment Selector for destination code segment
D	Size of gate: 1 = 32 bits; 0 = 16 bits

# Interrupciones

Verificación de privilegios para generar una interrupción.

# Interrupciones

Verificación de privilegios para generar una interrupción.

1. Verificar que la interrupción esté presente  $\rightarrow P=1$ .

# Interrupciones

Verificación de privilegios para generar una interrupción.

1. Verificar que la interrupción esté presente  $\rightarrow P=1$ .
2. Verificar el nivel para acceder a la interrupción



# Interrupciones

Verificación de privilegios para generar una interrupción.

1. Verificar que la interrupción esté presente  $\rightarrow P=1$ .
2. Verificar el nivel para acceder a la interrupción
  - Si la interrupción es generada por código:  $\rightarrow DPL \geq CPL$

# Interrupciones

Verificación de privilegios para generar una interrupción.

1. Verificar que la interrupción esté presente  $\rightarrow P=1$ .
2. Verificar el nivel para acceder a la interrupción
  - Si la interrupción es generada por código:  $\rightarrow DPL \geq CPL$
  - Si la interrupción es generada externamente: No se verifica.

# Interrupciones

Verificación de privilegios para generar una interrupción.

1. Verificar que la interrupción esté presente  $\rightarrow P=1$ .
2. Verificar el nivel para acceder a la interrupción
  - Si la interrupción es generada por código:  $\rightarrow DPL \geq CPL$
  - Si la interrupción es generada externamente: No se verifica.
  - Si la interrupción es una excepción del procesador: No se verifica.

# Interrupciones

Verificación de privilegios para generar una interrupción.

1. Verificar que la interrupción esté presente  $\rightarrow P=1$ .
2. Verificar el nivel para acceder a la interrupción
  - Si la interrupción es generada por código:  $\rightarrow DPL \geq CPL$
  - Si la interrupción es generada externamente: No se verifica.
  - Si la interrupción es una excepción del procesador: No se verifica.

Si alguna de las verificaciones falla:  $\rightarrow$  General Protection Fault (#GP)

## Bibliografía: Fuentes y material adicional

- Convenciones de llamados a función en x86:  
[https://en.wikipedia.org/wiki/X86\\_calling\\_conventions](https://en.wikipedia.org/wiki/X86_calling_conventions)
- Notas sobre System V ABI:  
[https://wiki.osdev.org/System\\_V\\_ABI](https://wiki.osdev.org/System_V_ABI)
- Documentación de NASM:  
<https://nasm.us/doc/>
- Artículo sobre el flag -pie:  
<https://eklitzke.org/position-independent-executables>
- Documentación de System V ABI:  
[https://uclibc.org/docs/psABI-x86\\_64.pdf](https://uclibc.org/docs/psABI-x86_64.pdf)
- Manuales de Intel:  
<https://software.intel.com/en-us/articles/intel-sdm>

# ¡Gracias!

Recuerden leer los comentarios al final de este video por aclaraciones o fe de erratas.