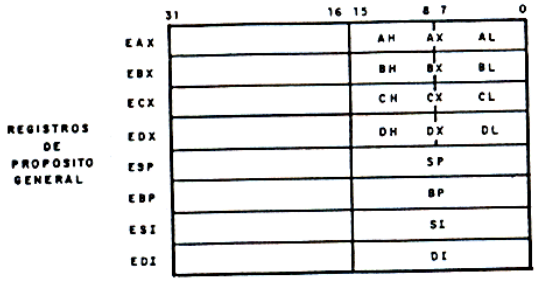
Estructura de los computadores II

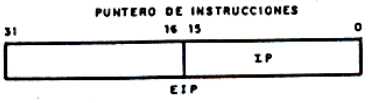
Introducción al modo protegido

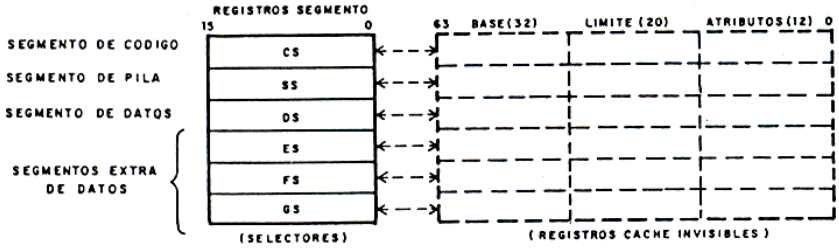
Antonio Cañas Vargas

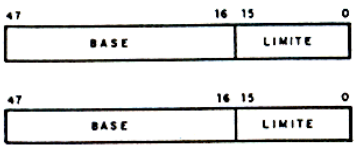
(Esta parte no entra en el examen)

REGISTROS DEL 386





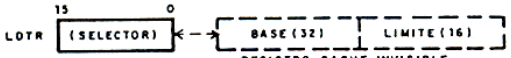




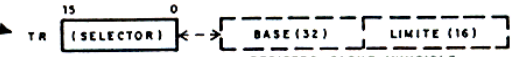
GDTR : Apunta a la tabla de descriptores globales (GDT)

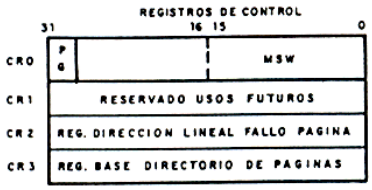
IDTR : Apunta a la tabla de descriptores de interrupción (IDT)

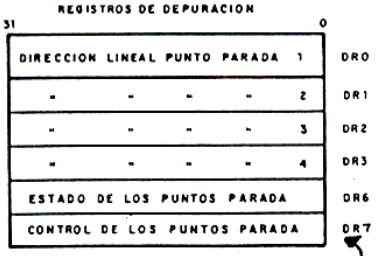
Selector del descriptor de la GDT que apunta a la tabla de descriptores locales (LDT)



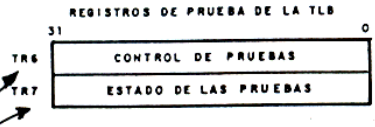
Selector del descriptor de la GDT que apunta al segmento de estado de la tarea (TSS)







Permiten especificar hasta 4 puntos de parada/ruptura hardware

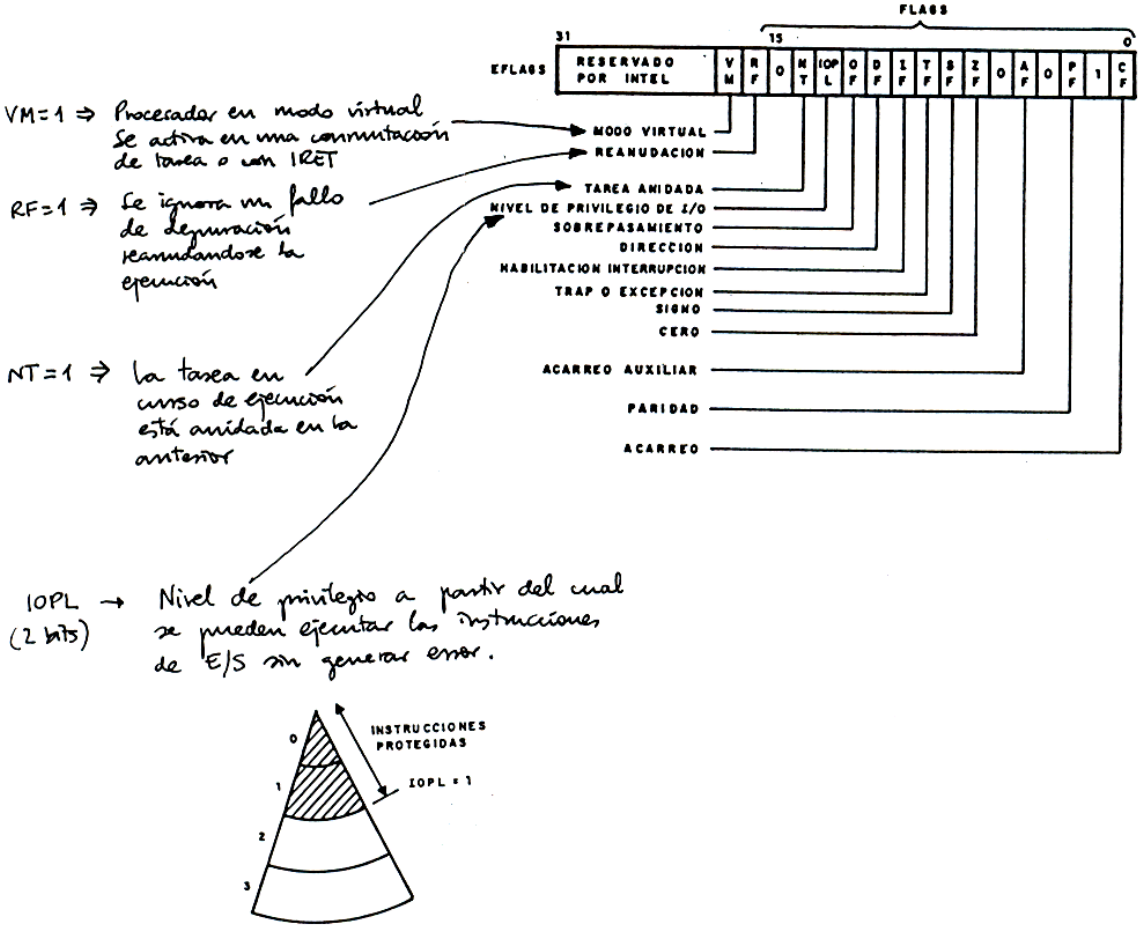


Sirven para comprobar la TLB de la unidad de paginación, leyendo de o escribiendo en ella

Permite determinar al depurador las condiciones que han producido el error.

Define y habilita o no, las condiciones de depuración (ruptura en ejecución de una instrucción, en escritura de datos, en lectura de datos, etc.)

REGISTRO EFLAGS



VM=1 ⇒ Procesador en modo virtual
Se activa en una conmutación de tarea o con IRET

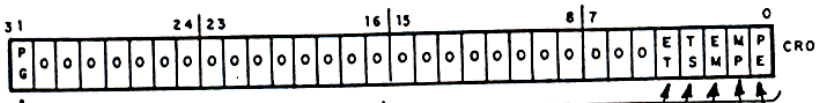
RF=1 ⇒ Se ignora un fallo de depuración reanudándose la ejecución

NT=1 ⇒ La tarea en curso de ejecución está anidada en la anterior

IOPL → Nivel de privilegio a partir del cual se pueden ejecutar las instrucciones de E/S sin generar error.
(2 bits)

REGISTROS DE CONTROL

CR0 : Doble palabra de estado de la máquina



MSW — Palabra de estado de la máquina

PE: Habilitación del modo protegido

MP: Coprocesador presente
= 1 ⇒ existe un coprocesador matemático en el sistema

PG: = 1 ⇒ habilitada la unidad de paginación

ET: Tipo de coprocesador
= 0 ⇒ 287
= 1 ⇒ 387

EM: Emulación
= 1 ⇒ se emplea un emulador software para ejecutar las instrucciones del coprocesador

TS: Tarea conmutada
Se pone a 1 al producirse una conmutación de tarea.
Después, cuando se encuentra una instrucción del coprocesador, se cambia de contexto (tarea) en éste y se pone TS a 0.

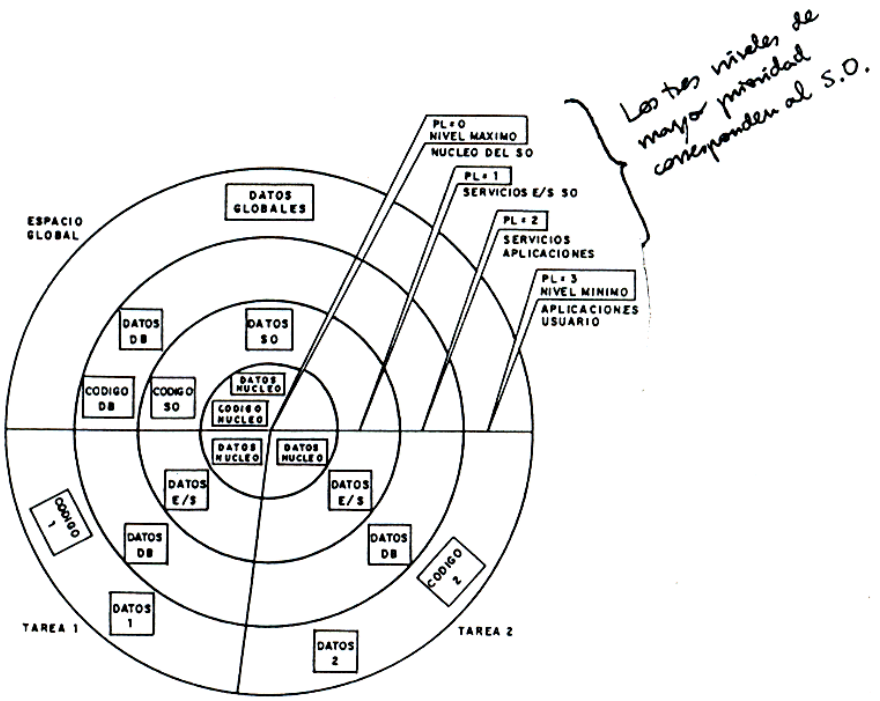
CR2 : Almacena la dirección lineal que se introdujo en la unidad de paginación para traducirla a dirección física, y que ocasionó un fallo de página o un error.
Utilizada por el manipulador de fallo de página (INT 14)

CR3 : Dirección física en la que comienza el directorio de tablas de páginas de la tarea en curso.

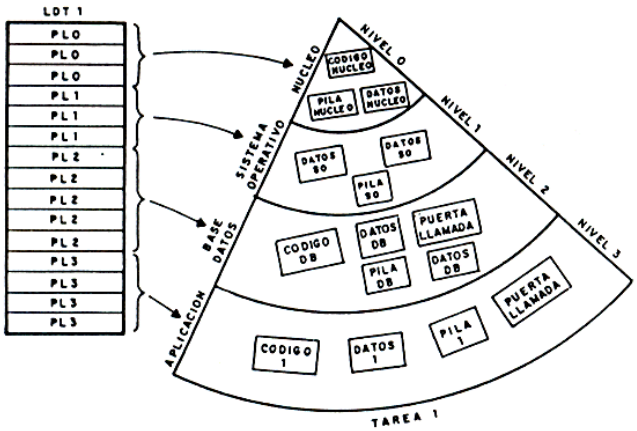
PROTECCIÓN EN LA SEGMENTACIÓN

Existen 4 niveles de privilegio.

{ Hay un espacio de segmentos global a todas las tareas.
" " " " " local para cada tarea.



Segmentos de una tarea (locales):

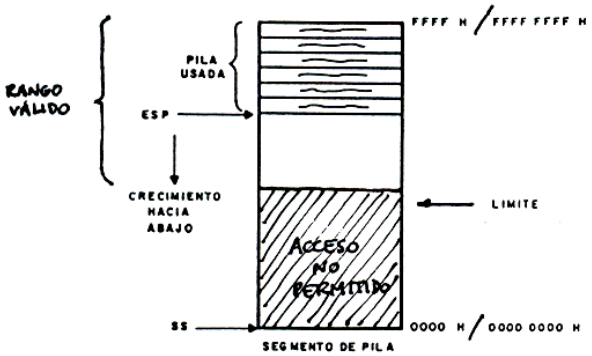


Cada uno de los segmentos de la tarea
Tiene asignado un nivel de privilegio
especificado en el campo DPL del descriptor de la LDT
que referencia a ese segmento

PROTECCIÓN EN LOS SEGMENTOS

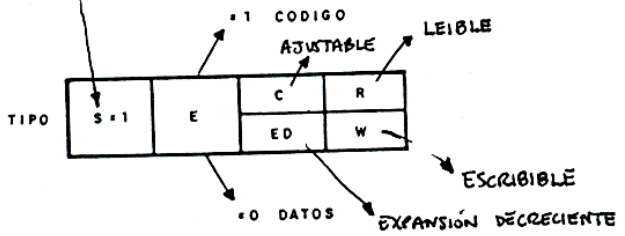
1. PROTECCIÓN DEL LÍMITE

- Si el desplazamiento sobrepasa el tamaño del segmento especificado por el límite ⇒ excepción
- En segmentos de pila (expansión decreciente) el rango válido de direcciones está entre 4GB (G=1) / 64KB (G=0) y límite+1



2. PROTECCIÓN DEL TIPO

(EN SEGMENTOS NORMALES)



El mecanismo de protección genera excepción en los siguientes casos:

1) Si se intenta escribir en un segmento de código (E=1)
2) Si se intenta leer de un segmento de código con R=0.
3) Si se intenta cargar CS con el valor de un selector que corresponda a un descriptor con E=0.
4) Si se intenta escribir en un segmento de datos con W=0.
5) Si se intenta cargar SS con el valor de un selector que corresponda a un descriptor con E=1 o W=0.

3 • PROTECCIÓN SEGÚN EL NIVEL DE PRIVILEGIO

Sean:

DPL (Descriptor Privilege Level) ≡ Nivel de privilegio del segmento al que va a acceder una instrucción. (almacenado en la tabla de descriptores (GDT o LDT) a la que accede la instrucción como paso intermedio de acceso a memoria)

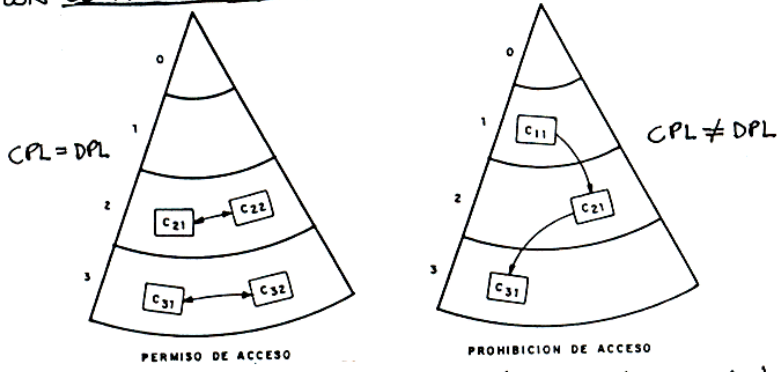
CPL (Current Privilege Level) ≡ Nivel de privilegio del segmento de código en curso, es decir, donde está la instrucción que se ejecuta. (almacenado en el registro cache asociado a CS)

RPL (Requester Privilege Level) ≡ Nivel de privilegio del segmento de código que accedió por última vez al segmento al que quiere acceder la instrucción. (almacenado en el selector del segmento al que se quiere acceder)

Existen 3 reglas de acceso a los distintos tipos de segmento:

1ª REGLA: ACCESO A SEGMENTOS DE CÓDIGO

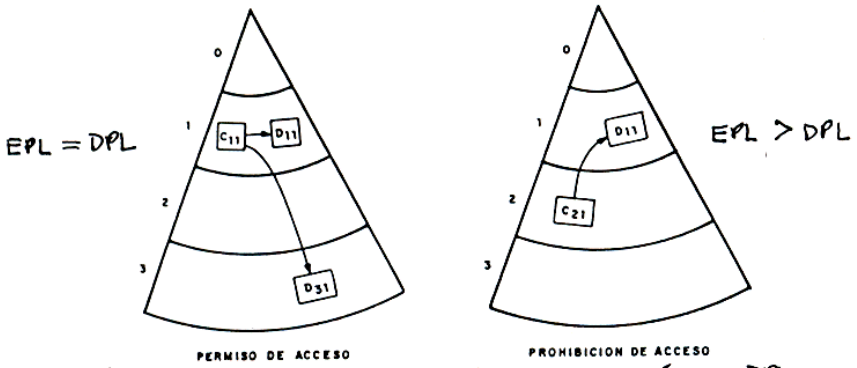
SÓLO SE PUEDE ACCEDER DE FORMA DIRECTA, MEDIANTE INSTRUCCIONES JMP Y CALL, DESDE UN SEGMENTO DE CÓDIGO CON UN DETERMINADO CPL A OTROS SEGMENTOS DE CÓDIGO CON EL MISMO NIVEL DE PRIVILEGIO (DPL).



Para llamar a rutinas con otro nivel de privilegio se dispone de las Puertas de Llamada

2ª REGLA: ACCESO A SEGMENTOS DE DATOS

DESDE UN SEGMENTO DE CÓDIGO CON UN DETERMINADO CPL, SÓLO SE PUEDE ACCEDER A SEGMENTOS DE DATOS QUE TENGAN IGUAL O MENOR (nº mayor) NIVEL DE PRIVILEGIO (DPL)



En realidad no se utiliza CPL en la comparación con DPL, sino Nivel de privilegio efectivo → **EPL** = máx(CPL, RPL)

3ª REGLA: ACCESO A SEGMENTOS DE PILA

SÓLO SE PUEDE EFECTUAR EL ACCESO A SEGMENTOS DE PILA CON IGUAL NIVEL (DPL) DE PRIVILEGIO QUE EL SEGMENTO DE CÓDIGO QUE LO SOLICITA (CPL).

PROTECCIÓN DE LAS PÁGINAS

Sólo hay dos niveles de privilegio < USUARIO (igual que 3 en segmentación)
SUPERVISOR (igual que 0, 1, 2 en segment.)

Sólo hay un bit de derechos de acceso: R/W

- Desde una página de supervisor se puede acceder a todas las páginas.
- Desde una página de usuario sólo se puede acceder a páginas de usuario.

PROTECCIÓN DE LAS INSTRUCCIONES

No todas las instrucciones se pueden ejecutar desde cualquier nivel de privilegio.

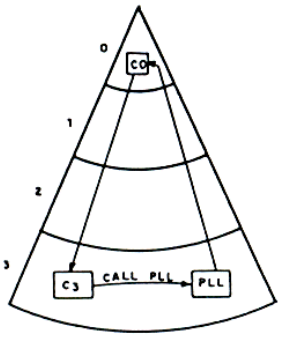
- **INSTRUCCIONES PROTEGIDAS:**
  Dedicadas a las E/S: IN, OUT, INS, OUTS, CLI, STI
  SÓLO SE PUEDEN EJECUTAR DESDE SEGMENTOS DE CÓDIGO CON CPL ≤ IOPL.

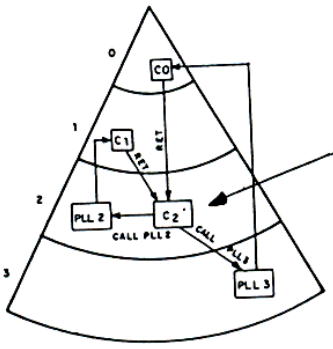
- **INSTRUCCIONES PRIVILEGIADAS:**
  SÓLO SE PUEDEN EJECUTAR DESDE EL NIVEL MÁS PRIVILEGIADO, CPL = 0:
  - Instrucciones que puedan modificar el IOPL: POPF, IRET, ...
  - Instrucciones que escriben los registros que controlan las tablas del sistema: LGDT, LIDT, SGDT, SIDT, LLDT, LTR, ...
  - Instrucciones que afectan al contenido de la palabra de estado: LMSW, SMSW, ...
  - Instrucción de parada HLT.

PUERTAS DE LLAMADA

Permiten acceder desde un segmento de código a otro con mayor nivel de privilegio. Por ejemplo, para llamadas al S.O.

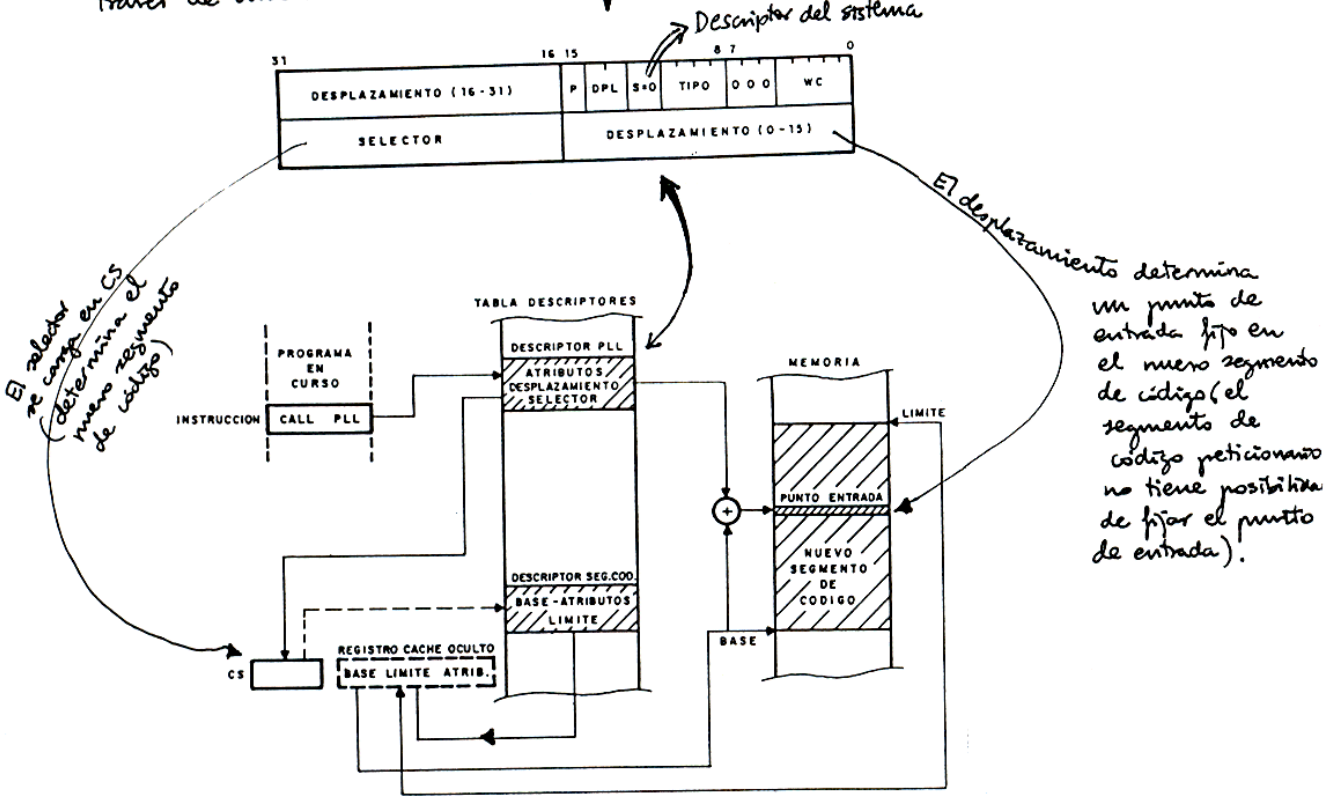
Ejemplos:





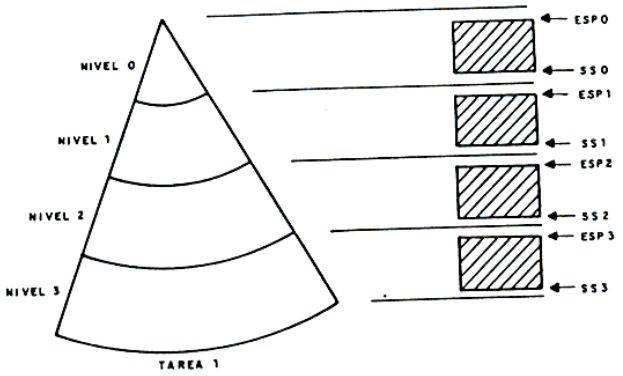
El nivel de privilegio del segmento de código ha de ser mayor o igual que el de la puerta.

Una puerta de llamada es un DESCRIPTOR local o global al que se accede a través de una instrucción CALL



COMPORTAMIENTO DE LA PILA EN LAS TRANSFERENCIAS INTERNIVEL

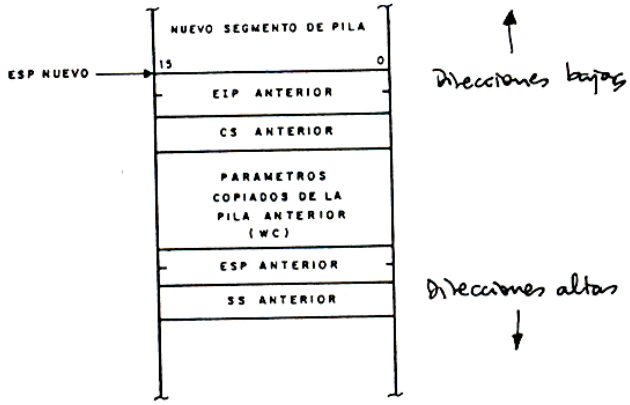
- Cada tarea dispone de un segmento de pila independiente para cada nivel de privilegio.
- Cuando se cambia de nivel de privilegio se cambia automáticamente de pila, modificándose el contenido de SS y ESP, que se encontraban guardados en el TSS.



Cuando se transfiere el control a través de una puerta de llamada, se hace una copia automática de parámetros entre las dos pilas afectadas.

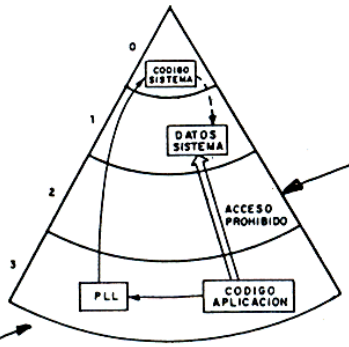
En el campo WC (Contador de palabras) de la puerta de llamada se especifica el número de palabras que se transfieren desde la cima de la pila en curso a la nueva, situada en otro nivel de privilegio.

Estructura del contenido de la pila nueva, después de la llamada →



PROBLEMA DE PROTECCIÓN QUE PODRÍA TENER EL USO DE LAS PUERTAS DE LLAMADA

(Problema del caballo de Troya)



Hay que evitar que desde un segmento de código se pueda acceder a un segmento de datos de mayor privilegio.

Pero un astuto programador, o un error de programa, podría enviar por medio de la transferencia de parámetros a la pila (opción WC) el selector del segmento de datos del sistema y algún dato para escribir en él!, supuesto que existen las instrucciones correspondientes en el código del sistema.

Solución: la comparación del DPL del segmento de datos a acceder no se hace con el CPL del segmento de código del sistema, sino con EPL = máx (CPL, RPL)

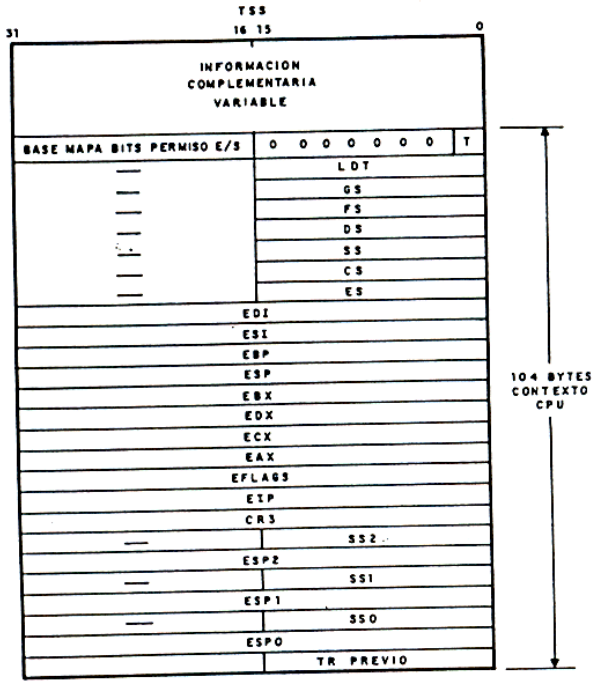
RPL = 3 (en los dos lsbits de CS = CPL del código aplicación)
CPL = 0 (en el registro cache de CS)
DPL = 1 (en el descriptor del segmento de datos del sistema)

CPL ≥ DPL ⇒ acceso permitido a los datos !!
EPL = máx(CPL, RPL) = máx(0,3) = 3 < DPL ⇒ acceso prohibido

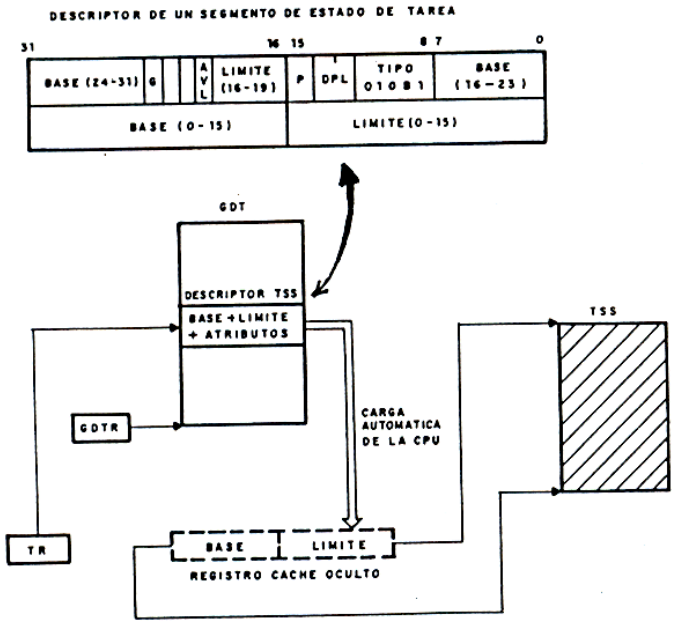
CONMUTACIÓN DE TAREAS

- Una conmutación de tarea consiste en abandonar el procesamiento de la tarea en curso para inicializarlo con otra tarea.
- Para reanudar una tarea en cualquier momento hay que disponer de algún objeto que almacene el estado completo del procesador cuando abandonó la tarea anteriormente.
- Cada tarea tiene guardado el contexto de la CPU en un segmento llamado:

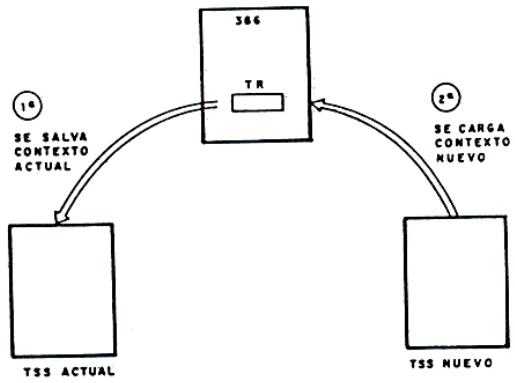
Segmento de Estado de la Tarea (TSS)



El registro de tarea (TR) actúa como un selector de la GDT, seleccionando un descriptor de un TSS.



La conmutación de tarea se realiza por medio de una instrucción CALL o JMP, que provoca el cambio de TR, y que se guarde el contexto actual y se cargue el nuevo.

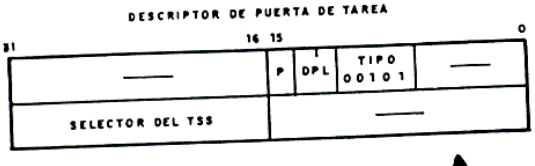


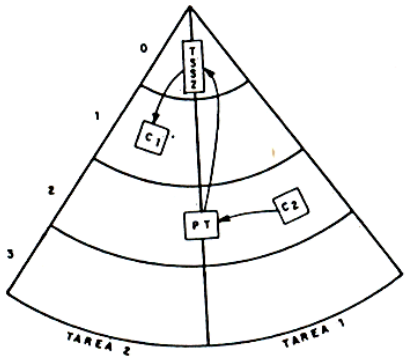
Un TSS tiene siempre DPL = 0 y exige ser llamado desde un segmento de código con CPL = 0.

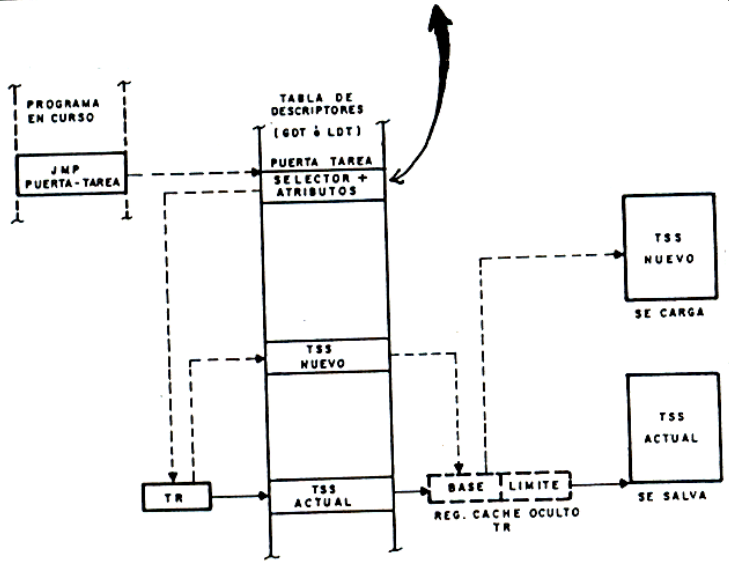
Para poder acceder a un TSS desde un segmento de código con menor privilegio se usan las puertas de tarea.

PUERTAS DE TAREA (son descriptores especiales del sistema)

Se comportan de forma similar a las puertas de llamada:

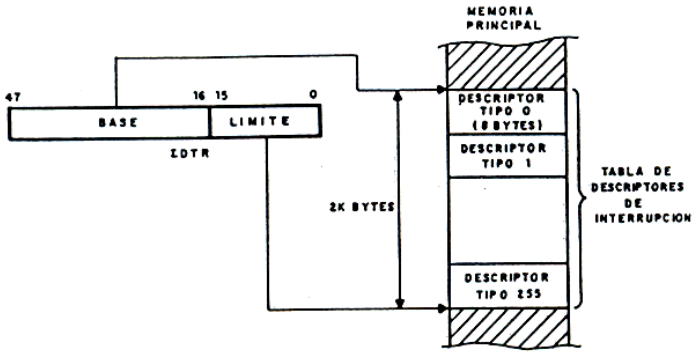






INTERRUPCIONES EN EL MODO PROTEGIDO

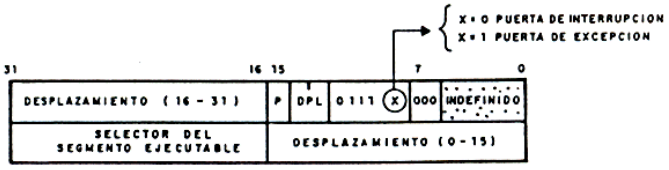
Se dispone de una tabla para el tratamiento de 256 posibles interrupciones (IDT), que puede estar en cualquier zona de memoria. Cada entrada es un descriptor de interrupción:

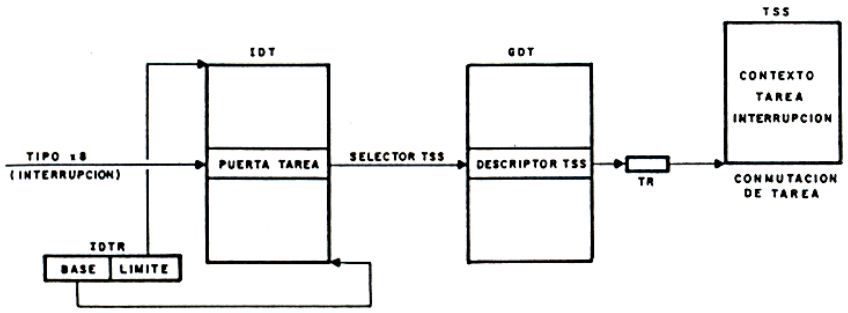


Un descriptor de interrupción puede ser:

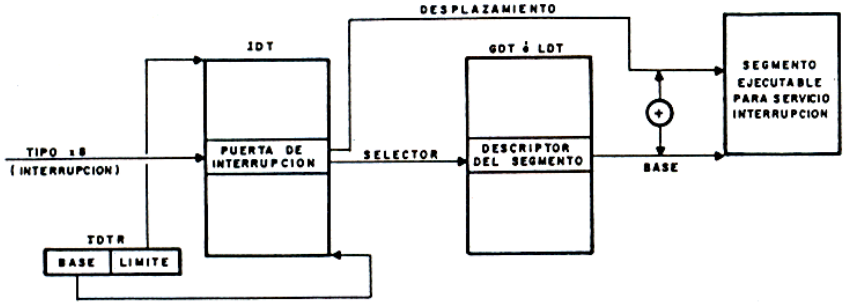
- una puerta de tarea
- una puerta de interrupción
- una puerta de excepción

PUERTA DE TAREA





Mecanismo de funcionamiento del servicio de una interrupción, a través de la activación de una puerta de tarea.



Mecanismo de funcionamiento de una puerta de interrupción.