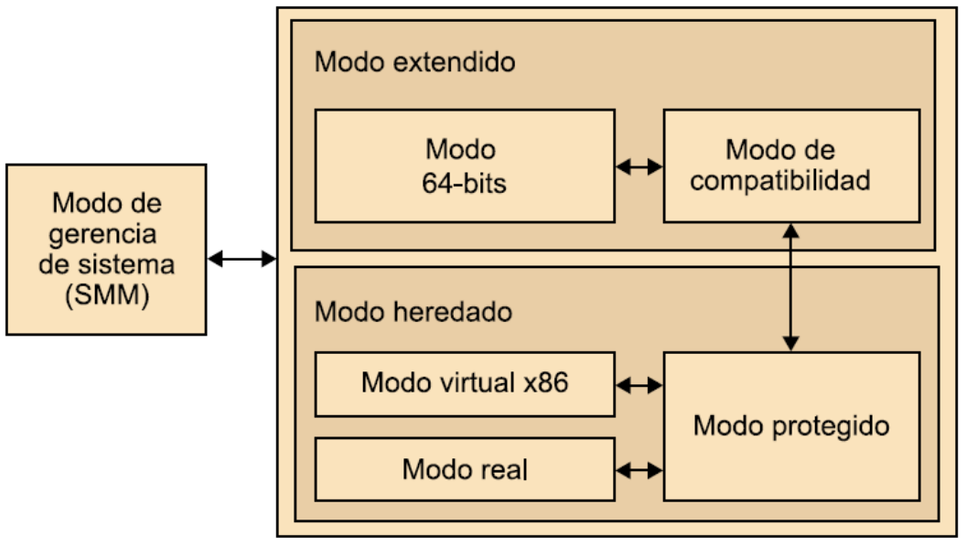
MODOS DE OPERACIÓN DE LAS ARQUITECTURAS

Modos de operación de la arquitectura x86-64



- En un sistema operativo de 64 bits, los programas de 64 bits se ejecutan en modo de 64 bits y las aplicaciones de 16 y 32 bits se ejecutan en modo de compatibilidad.
- Los programas de 16 y 32 bits que se tengan que ejecutar en modo real o virtual x86 no se podrán ejecutar en modo extendido si no son emulados.

MODO EXTENDIDO

a. Modo 64 bits

El modo de 64 bits proporciona acceso a 16 registros de propósito general de 64 bits. En este modo se utilizan direcciones virtuales (o lineales) que por defecto son de 64 bits y se puede acceder a un espacio de memoria lineal de 264 bytes.
El tamaño por defecto de los operandos se mantiene en 32 bits para la mayoría de las instrucciones.
El tamaño por defecto puede ser cambiado individualmente en cada instrucción mediante modificadores. Además, soporta direccionamiento relativo a PC (RIP en esta arquitectura) en el acceso a los datos de cualquier instrucción.

b. Modo de compatibilidad

El modo de compatibilidad permite a un sistema operativo de 64 bits ejecutar directamente aplicaciones de 16 y 32 bits sin necesidad de recompilarlas.
En este modo, las aplicaciones pueden utilizar direcciones de 16 y 32 bits, y pueden acceder a un espacio de memoria de 4 Gbytes. El tamaño de los operandos puede ser de 16 y 32 bits.
Desde el punto de vista de las aplicaciones, se ve como si se estuviera trabajando en el modo protegido dentro del modo heredado.

MODO HEREDADO

a. Modo heredado de 16 y 32 bits

El modo heredado de 16 y 32 bits es utilizado por los sistemas operativos de 16 y 32 bits. Cuando el sistema operativo utiliza los modos de 16 bits o de 32 bits, el procesador actúa como un procesador x86 y solo se puede ejecutar código de 16 o 32 bits. Este modo solo permite utilizar direcciones de 32 bits, de manera que limita el espacio de direcciones virtual a 4 GB.

Dentro de este modo general hay tres modos:

- ***Modo real.*** Implementa el modo de programación del Intel 8086, con algunas extensiones, como la capacidad de poder pasar al modo protegido o al modo de gestión del sistema. El procesador se coloca en modo real al iniciar el sistema y cuando este se reinicia. Es el único modo de operación que permite utilizar un sistema operativo de 16 bits.
  El modo real se caracteriza por disponer de un espacio de memoria segmentado de 1 MB con direcciones de memoria de 20 bits y acceso a las direcciones del hardware (sistema de E/S). No proporciona soporte para la protección de memoria en sistemas multitarea ni de código con diferentes niveles de privilegio.

- ***Modo protegido.*** Este es el modo por defecto del procesador. Permite utilizar características como la memoria virtual, la paginación o la computación multitarea.
  Entre las capacidades de este modo está la posibilidad de ejecutar código en modo real, modo virtual-8086, en cualquier tarea en ejecución.

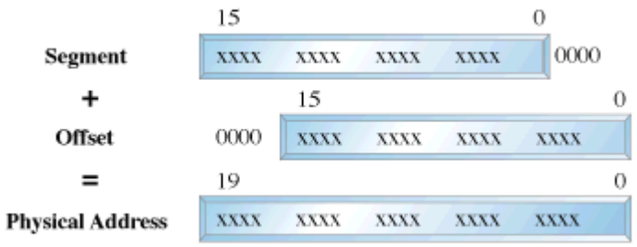
- ***Modo virtual 8086.*** Este modo permite ejecutar programas de 16 bits como tareas dentro del modo protegido.

MODO DE GESTIÓN DE SISTEMA

El modo de gestión de sistema o system management mode (SMM) es un modo de operación transparente del software convencional (sistema operativo y aplicaciones). En este modo se suspende la ejecución normal (incluyendo el sistema operativo) y se ejecuta un software especial de alto privilegio diseñado para controlar el sistema. Tareas habituales de este modo son la gestión de energía, tareas de depuración asistidas por hardware, ejecución de microhardware o un software asistido por hardware. Este modo es utilizado básicamente por la BIOS y por los controladores de dispositivo de bajo nivel.
Accedemos al SMM mediante una interrupción de gestión del sistema (SMI, system management interrupt). Una SMI puede ser generada por un acontecimiento independiente o ser disparada por el software del sistema por el acceso a una dirección de E/S considerada especial por la lógica de control del sistema.

DIRECCIONAMIENTO EN MODO REAL

- Permite direccional solo el 1 MB de la memoria.
- Se utilizaba en los microprocesadores que procesaban hasta 16 bits.
- Acceder a una localidad de memoria se combinan una dirección de segmento y un desplazamiento.
- Dirección de segmento esta almacenada en el registro de segmento respectivo.
- La dirección de segmento define la posición en la memoria desde la cual inicia el segmento de 64KB.
- El desplazamiento selecciona cualquier localidad dentro del segmento.



Ejemplo:

Dirección de segmento es 1000H y su desplazamiento es 22H, la dirección real es la sumatoria de los dos:

10000 H Dirección de Segmento — Se le adiciona un cero a la derecha. Con esto se consigue una
+ 22 H Desplazamiento — dirección de 20 Bits para poder alcanzar el 1 MB de memoria.

_____
10022 H Dirección real

Los segmentos de 64K del modo real se dice que inician en un límite de párrafo ósea 16 bits.
Por ejemplo:
20000H hasta 2FFFFH FFFFH es igual a 65536 que es 2 ^ 16
10000H hasta 1FFFFH

Memoria ALTA

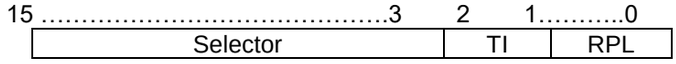
Para la dirección de segmento FFFFH ósea FFFF0H se debe tener en cuenta que el PC soporte la Terminal de dirección A20, y trabaje con el archivo HIMEM.SYS.

Por ejemplo: Segmento FFFFH ósea FFFF0H y desplazamiento 3000H
FFFF0 H
+ 3000 H
_____
102FF0 H por lo tanto esta área se conoce como **memoria ALTA** FFFF0H hasta 10FFEFH

DIRECCIONAMIENTO EN MODO PROTEGIDO

- Permite acceder a los datos que están dentro y por encima del 1MB de memoria.
- Se sigue utilizando la dirección de segmento para acceder a una localidad de memoria.
- Los registros de segmento ya no poseen la dirección de segmento sino un selector que apunta a un descriptor.
- El registro de segmento contiene un SELECTOR que elige un descriptor de una tabla.
- La dirección de desplazamiento puede ser un número de 32 bits en vez de utilizar uno de 16 bits como en modo real. Es por esto que puede direccionar hasta 4 Gb de longitud.

Selector:



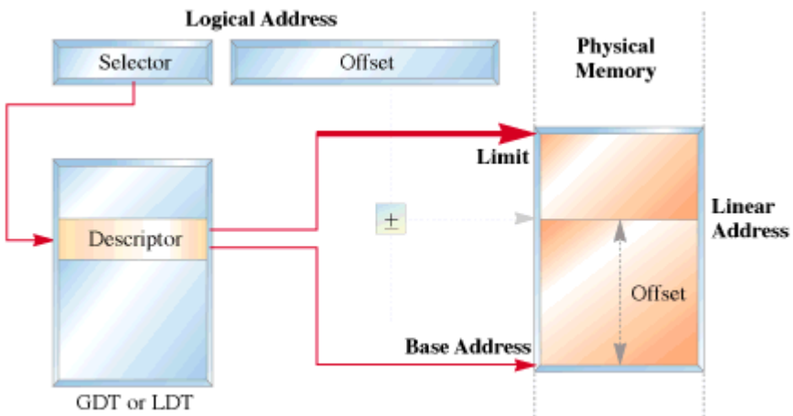
TI = 0 Tabla de descriptor Global ó Sistema
TI = 1 Tabla de descriptor Local ó Aplicación

RPL Nivel de privilegio solicitado 00 Mayor y 11 Menor (Mayor prioridad sobre el de los Derechos Acceso)

Selector es el número del descriptor utilizado.

Descriptor:

- Especifica la ubicación del segmento de memoria, su longitud y sus derechos de acceso.
- Existen dos tablas de descriptores cada una con 8192 descriptores.
  - Descriptores globales o del sistema.
  - Descriptores locales o de aplicación.
- En total una aplicación puede tener 16384 descriptores.



Descriptor del 80286

00000000	00000000
Derechos Acceso	Base (B23 – B16)
Base (B15 – B0)	
Límite (L15 – L0)	

Descriptor del 80386 hasta el 32 bits

Base (B31 – B24)	G	D	0	AV	Límite (L19 – L16)
Derechos Acceso	Base (B23 – B16)				
Base (B15 – B0)					
Límite (L15 – L0)					

La dirección Base es la dirección de inicio del segmento en memoria
La dirección Límite es la dirección final del segmento en memoria.

80286			**80386 y Superiores**		
Base	F00000 H	24 Bits	Base	00F00000 H	32 Bits
Límite	00FF H	16 Bits	Límite	000FF H	20 Bits

Explicación de los Bits especiales de los descriptores 80386 y superiores

Bit G = 1 amplia el limite hasta 4GB de direccionamiento agregando 4 Kb que es igual a colocar 000 adicional.
= 0 especifica que el limite solo va desde 00000 H hasta FFFFF H.

Bit AV = 0 El segmento no esta disponible.
= 1 El segmento si esta disponible.

Bit D = 0 Indica 16 Bits el cual es el modo en que operan las instrucciones en el microprocesador
= 1 Indica 32 Bits el cual es el modo en que operan las instrucciones en el microprocesador

Byte Derechos de Acceso

7	**6** **5**	**4** **3**	**2**	**1** **0**	
P	DPL	S	E	ED/C	R/W	A

A = 0 El segmento no ha sido utilizado
= 1 El segmento ya ha sido utilizado
E = 0 El descriptor es de un segmento de pila
ED = 0 El segmento se expande hacia arriba (Segmento de Datos)
ED = 1 El segmento se expande hacia abajo (Segmento de Pila)
W = 0 Los datos no puede ser escritos.
W = 1 Los datos pueden ser escritor.
E = 1 Es descriptor es de un segmento de Código
C = 0 Ignora el nivel de privilegio del descriptor.
C = 1 Honra el nivel de privilegio del descriptor.
R = 0 El segmento de código no puede leerse.
R = 1 El segmento de código puede leerse.
S = 0 Descriptor del sistema o Global.
= 1 Descriptor de aplicación o Local.

DPL = Establece el nivel de privilegio del descriptor 00 Mayor y 11 Menor
P = 0 Descriptor sin definir
= 1 El segmento contiene una base y un límite validos.

TABLAS DE DESCRIPTORES

Las tablas de descriptores globales y locales se encuentran en el sistema de memoria.

Microprocesador contiene registros invisibles para el programa. NO SON DIRECCIONADOS POR SOFTWARE, motivo por el cual reciben este nombre (TR, LDTR,GDTR,IDTR)

La parte invisible de estos registros recibe el nombre de caché (no se debe confundir con los niveles 1 o 2 encontradas en el microprocesador). Cada vez que se cambia el número en el registro de segmento, la parte del registro de segmentos invisible para el programa es cargada en la dirección base., el límite y los derechos de acceso. Cuando se escribe un nuevo número en un registro, el microprocesador accede a la tabla de descriptores y carga el descriptor en la caché invisible para el programa que forma parte del registro del segmento. El descriptor se mantiene ahí y es usado para acceder al segmento de memoria hasta que el número de segmento es cambiado nuevamente. Esto permite al microprocesador acceder repetidamente a un segmento de memoria sin consultar en cada ocasión la tabla de descriptores (de ahí el término caché).

GDTR (registro de tabla de descriptores globales)
LDTR (registro de la tabla de descriptores locales)
IDTR (registro de tabla de descriptores de interrupción) contiene la dirección base y límite de la tabla de descriptores.

TR (registro de tarea) contiene un selector, el cual accede a un descriptor que define una tarea. Una tarea es con frecuencia un procedimiento o un programa de aplicación. El descriptor para el procedimiento o programa es almacenado en la tabla de descriptores globales, de forma que el acceso pueda ser controlado por medio de los niveles de privilegio.

Registro	Modo Real		Modo Protegido		Modo virtual 8086	
	Escr	Lect	Escr	Lect	Escr	Lect
Registros generales	Sí	Sí	Sí	Sí	Sí	Sí
Registros de segmento	Sí	Sí	Sí	Sí	Sí	Sí
Indicadores	Sí	Sí	Sí	Sí	IOPL	IOPL
Registros de control	Sí	Sí	CPL=0	CPL=0	No	Sí
GDTR	Sí	Sí	CPL=0	Sí	No	Sí
IDTR	Sí	Sí	CPL=0	Sí	No	Sí
LDTR	No	No	CPL=0	Sí	No	No
TR	No	No	CPL=0	Sí	No	No
Registros de depuración	Sí	Sí	CPL=0	CPL=0	No	No
Registros de test	Sí	Sí	CPL=0	CPL=0	No	No

Características de los dos modos principales de operación en la arquitectura x86-64

Modo de operación		Sistema operativo	Las aplicaciones necesitan recompilación	Por defecto		Tamaño de los registros de propósito general
				Tamaño (en bits) de las direcciones	Tamaño (en bits) de los operandos	
Modo extendido	Modo de 64 bits	Sistema operativo de 64 bits	sí	64	32	64
	Modo compatibilidad		no	32		32
				16	16	16
Modo heredado	Modo protegido	Sistema operativo de 32 bits	no	32	32	32
				16	16	
	Modo virtual-8086			16	16	16
	Modo real	Sistema operativo de 16 bits