PROTECTED MODE IN MICROPROCESSORS

• In the protected-mode, memory **larger** than **1 MB** can be accessed.**Windows XP** operates in the **protected mode**.

• In addition, **segments** can be of **variable size** (below or above **64 KB**).

• Some **system control instructions** are *only* valid in the protected mode.

• In protected mode, the base:offset logical memory addressing scheme (which is used in real mode) is changed.

• The offset part of the logical memory address is still used. However, when in the protected mode, the processor can work either with 16-bit offsets (the 16-bit instruction mode) or with 32-bit offsets (the 32-bit instructionmode). A 32-bit offset allows segments of up to **4G bytes** in length. Notice that in real-mode the only available instruction mode is the 16-bit mode (during which accessing 32-bit registers requires the prefix 66h).

• However, the segment base address calculation is different in protected mode. Instead of appending a 0 at the end of the segment register contents to create a segment base address (which gives a 20-bit physical address), the segment register contains a **selector** that *selects* a **descriptor** from a descriptor table. The descriptor *describes* the memory segment's location,length, and access rights. This is similar to selecting one card from a deck of cards in one's pocket.

• Because the segment register and offset address still create a logical memory address, protected mode **instructions** are the same as real mode instructions. In fact, most programs written to function in the real mode will function without change in the protected mode.

DESCRIPTORS:

• The selector, located in the segment register, selects one of **8192 descriptors** from one of two tables of descriptors (stored in memory): the global and local descriptor tables. The descriptor describes the **location**, **length** and **access rights** of the memory segment.

• Each descriptor is **8 bytes long** and its format is shown below:

The 8192 descriptor table requires 8 * 8192 = **64K bytes** of memory. The main parts of a descriptor are:

Base (B31 – B0): indicates the starting location **(base address)** of the memory segment. This allows segments to begin at any location in the processor's 4G bytes of memory.

Limit (L19 – L0): contains the **last offset address** found in a segment. Since this field is 20 bits, the segment size could be anywhere between 1 and 1M bytes. However, if the **G bit (granularity bit)** is set, the value of the limit is multiplied by **4K bytes** (i.e., appended with FFFH). In this case, the segment size could be anywhere between 4K and 4G bytes in steps of 4K bytes.

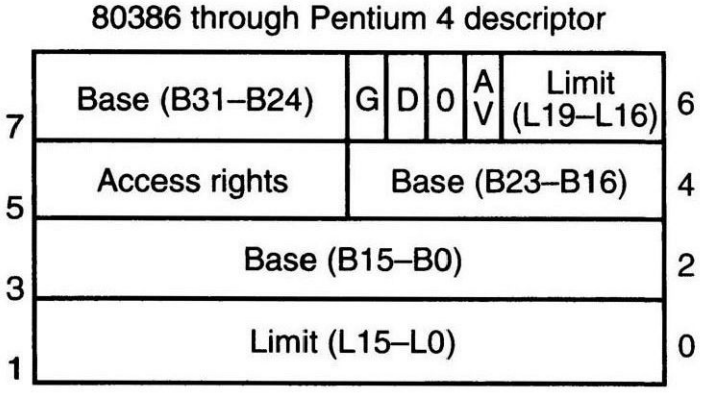
Example,
Base = Start = **10000000h**
Limit = **001FFh** and G = **0**
So, End = Base + Limit = 10000000h + 001FFh = 100001FFh
Segment Size = **512 bytes**
Base = Start = **10000000h**
Limit = **001FFh** and G = **1**
So, End = Base + Limit * 4K = 10000000h + **001FFFFFh** = **101FFFFFh**
Segment Size = **2M bytes**

AV bit: is used by some operating systems to indicate that the segment is available (AV = 1) or not available (AV = 0).

D bit: If D = 0, the instructions are 16-bit instructions, compatible with the 8086-80286 microprocessors. This means that the instructions use 16-bit offset addresses and 16-bit registers by default. This mode is the 16-bit instruction mode or DOS mode. If D = 1, the instructions are 32-bits by default (Windows XP works in this mode). By default, the 32-bit instruction mode assumes that all offset addresses and all registers are 32 bits. Note that the default for register size and offset address can be overridden in both the 16- and 32-bit instruction modes using the 66h and 67h prefixes. In 16-bit protected-mode, descriptors are still used but segments are supposed to be a maximum of 64K bytes.

Access rights byte: allows complete control over the segment. If the segment is a data segment, the direction of growth is specified. If the segment grows beyond its limit, the microprocessor's operating system program is interrupted, indicating a **general protection fault**. You can specify

whether a data segment can be written or is write-protected. The code segment can have reading inhibited to protect software. This is why It is called protected mode. This kind of protection is unavailable in realmode.



.

SELECTORS:

Descriptors are chosen from the descriptor table by the segment register.

There are two descriptor tables:

Global descriptors table: contains segment definitions that apply to **all** programs (also called **system descriptors**).

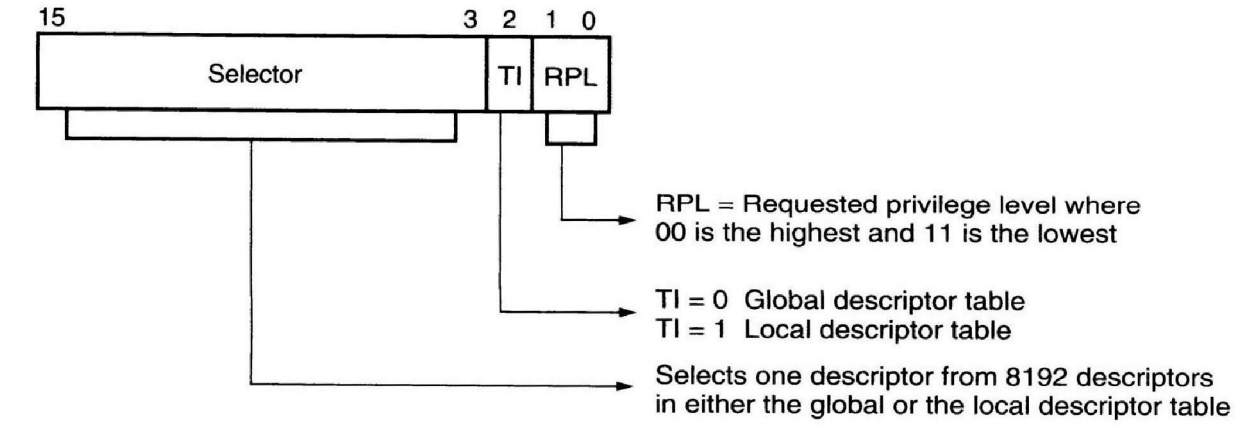
Local descriptors table: usually unique to an application (also called **application descriptors**).

Each descriptor table contains 8192 descriptors, so a total of 16,384 descriptors are available to an application at any time. This allows up to 16,384 memory segments to be described for each application. The Figure below shows the segment register in the protected mode. It contains:

13-bit selector field: chooses one of the 8192 descriptors from the descriptor table (213 = 8192).

Table indicator (TI) bit: selects either the global descriptor table (TI = 0) or the local descriptor table (TI = 1).

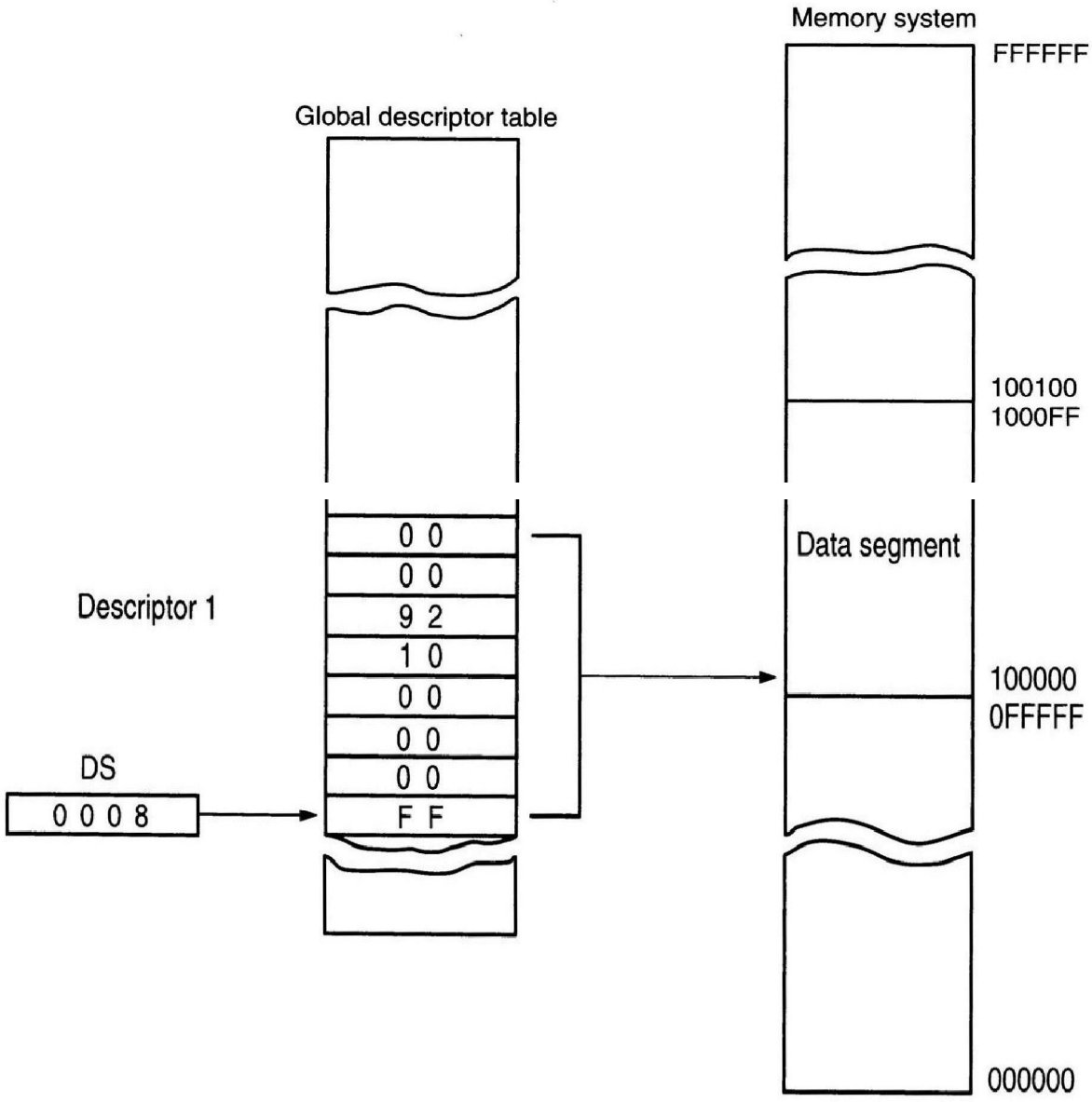
Requested privilege level (RPL) field: requests the access privilege level of a memory segment. The highest privilege level is 00 and the lowest is 11.If the requested privilege level matches or is higher in priority than the privilege level set by the access rights byte, access is granted. Windows uses privilege level 00 (ring 0) for the kernel and driver programs and level 11 (ring 3) for applications. Windows does not use levels 01 or 10. If privilege levels are violated, the system normally indicates a **privilege level violation**.



Example:

Real Mode: DS = 0008h, then the data segment begins at location **00080h** and its length is **64K bytes**.

Protected Mode: DS = 0008h = 0000 0000 0000 **1000**, then the selector selects **Descriptor 1** in the **global** descriptor table using a requested privilege level of **00**. The global descriptor table is stored in memory as shown below.



Descriptor number 1 contains a descriptor that defines the base address as 00100000h with a segment limit of 000FFh. This refers to memory locations **00100000h – 001000FFh** for the data segment.