Assignment No. 6

Att (2)	Perm (5)	Oral (3)	Total (10)	Sign

Aim: Write X86/64 ALP to switch from real mode to protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers.

6.1 Theory:

Real Mode:

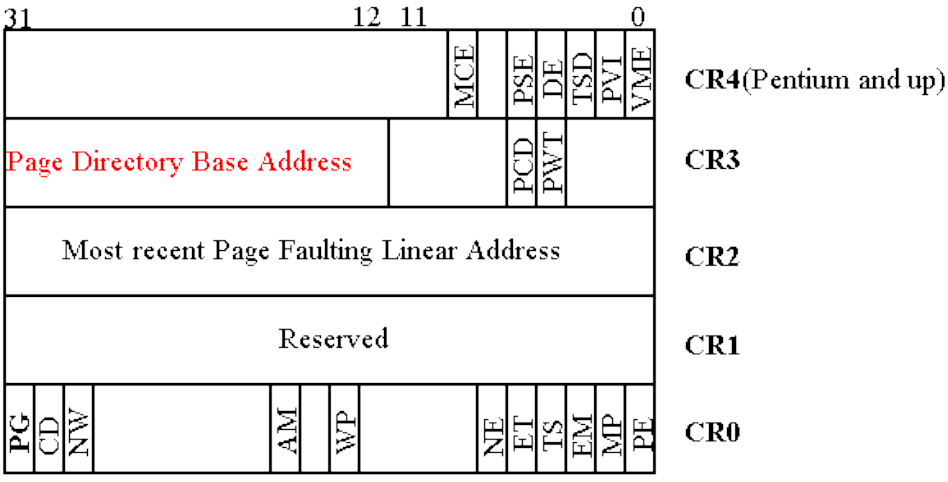
Real mode, also called real address mode, is an operating mode of all x86-compatible CPUs. Real mode is characterized by a 20-bit segmented memory address space (giving exactly 1 MiB of addressable memory) and unlimited direct software access to all addressable memory, I/O addresses and peripheral hardware. Real mode provides no support for memory protection, multitasking, or code privilege levels.

Protected Mode:

In computing, protected mode, also called protected virtual address mode is an operational mode of x86-compatible central processing units (CPUs). It allows system software to use features such as virtual memory, paging and safe multi-tasking designed to increase an operating system's control over application software.

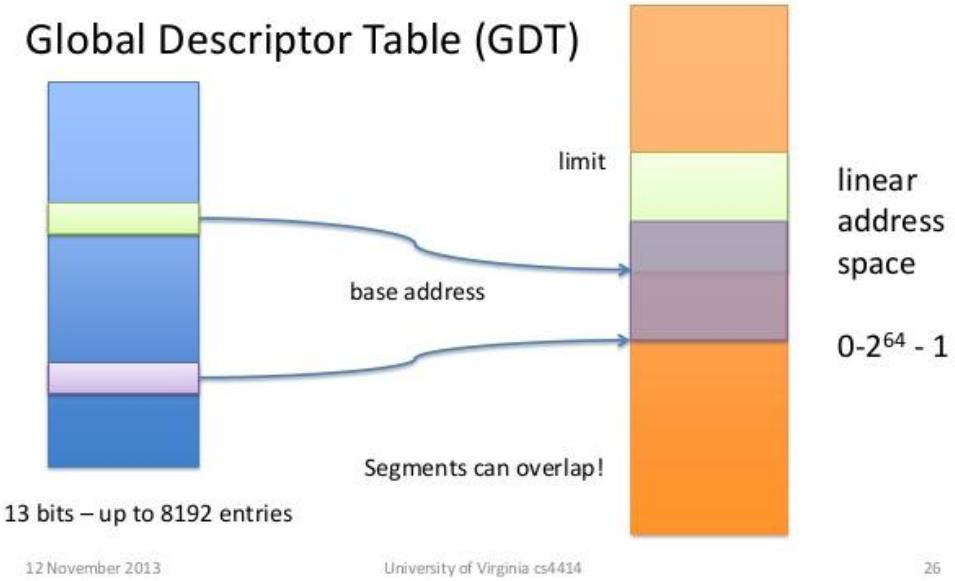
When a processor that supports x86 protected mode is powered on, it begins executing instructions in real mode, in order to maintain backward compatibility with earlier x86 processors. Protected mode may only be entered after the system software sets up several descriptor tables and enables the Protection Enable (PE) bit in the control register 0 (CR0).

Control Register :



Global Descriptor Table Register

This register holds the 32-bit base address and 16-bit segment limit for the global descriptor table (GDT). When a reference is made to data in memory, a segment selector is used to find a segment descriptor in the GDT or LDT. A segment descriptor contains the base address for a segment.

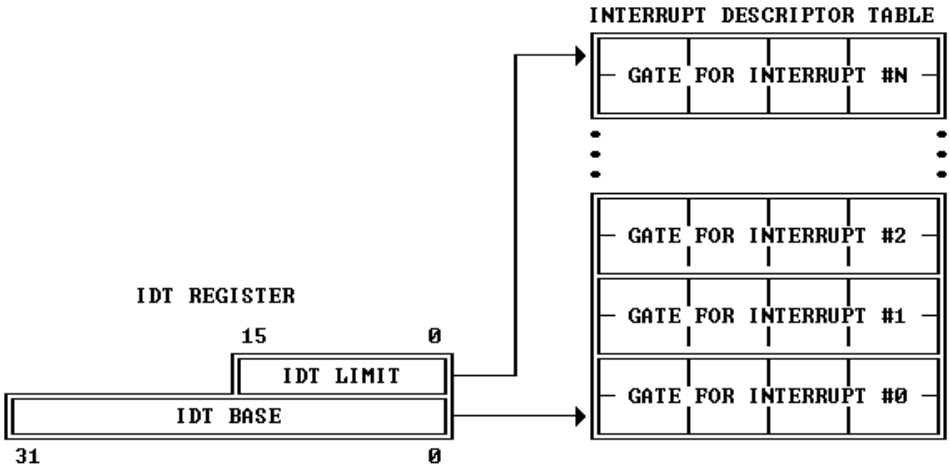


Local Descriptor Table Register

This register holds the 32-bit base address, 16-bit segment limit, and 16-bit segment selector for the local descriptor table (LDT). The segment which contains the LDT has a segment descriptor in the GDT. There is no segment descriptor for the GDT. When a reference is made to data in memory, a segment selector is used to find a segment descriptor in the GDT or LDT. A segment descriptor contains the base address for a segment

Interrupt Descriptor Table Register

This register holds the 32-bit base address and 16-bit segment limit for the interrupt descriptor table (IDT). When an interrupt occurs, the interrupt vector is used as an index to get a gate descriptor from this table. The gate descriptor contains a far pointer used to start up the interrupt handler.



6.2 Algorithm :

1. Start
2. Display the message using sys_write call
3. Read CR0
4. Checking PE bit, if 1=Protected Mode
5. Load number of digits to display
6. Rotate number left by four bits
7. Convert the number in ASCII
8. Display the number from buffer
9. Exit using sys_exit call

6.3 Explain Instruction Set:

1. SMSW:

   _____

   _____

2. SGDT: _____

   _____

3. SLDT: _____

   _____

4. STR: _____

   _____

5. SIDT: _____
   _____

6. _____
   _____

6.4 Conclusion: Hence we performed an ALP to program to use GDTR, LDTR and IDTR in Real Mode

6.5 Assignment Question:

1. What is GDT and GDTR?
2. What is LDT and LDTR?
3. What is IDT and IDTR?
4. What is selector?
5. Function of Descriptor (GDT ,LDT ,IDT) ?
6. What is mean by Interrupt Handler?
7. Explain Difference betweet Real Mode & Protected Mode?
8. Explain CR0 in Detail:
9. Explain POST Sequece?
10. Explain Flowchart of Real to Protected Mode Switch?