Assignment No. 6.

Title :- To read & display contents pointed by GDTR, LDTR & IDTR.

Problem Statement :-

Write an ALP to read & display the table content pointed by GDTR / LDTR & IDTR.

Objective :-

To understand how to read & display content of GDTR, LDTR & IDTR register.

Outcome :-

we will study different Descriptor tables in system also different register associated with it.

S/W packages / H/W appratus :- editor :- gedit
Assembler :- NASM
Debugger :- gdb.

Theory :-

GDTR & IDTR :-

The register hold 32-bit linear base address & 16-bit limit of GDT & IDT.

When a reference is made to data in memory, a segment selector is used to find a segment descriptor in GDT or LDT.

Local Descriptor Table Register :-

The register holds the 32-bit base address, 16-bit segment limit, and 16-bit segment selector for LDT. The segment which contains LDT has a segment descriptor in the GDT.

Interrupt Descriptor Table register :-

This register hold the 32-bit base address & 16-bit segment limit for interrupt descriptor table. When interrupt occurs, the interrupt vector is used as an index to get a gate descriptor from this table.

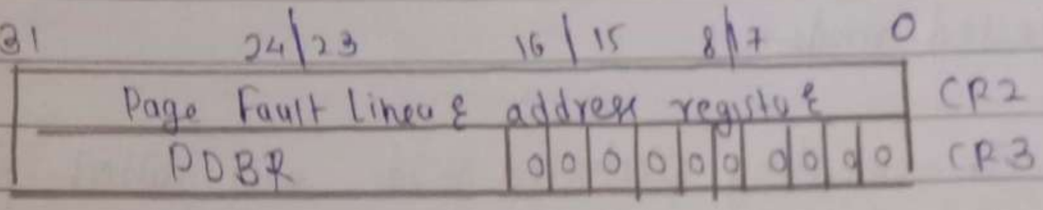

fig. control register CR2 & CR3

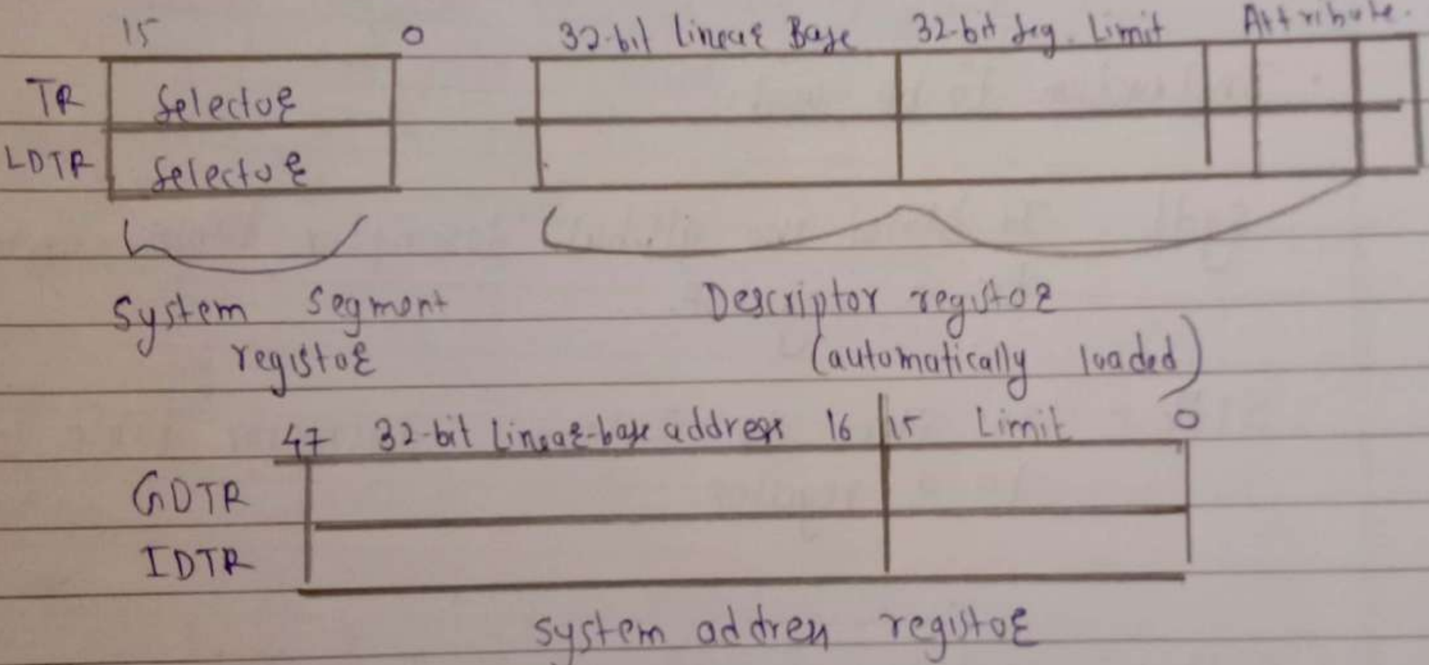

fig. System address & system segment Registers.

- Real Mode :-

Real mode, called real address mode of all x86 compatible CPUs. Real mode characterized by 20-bit Segmented memory address space & unlimited direct software access to addressable memory., I/o & peripheral h/w.

- Protected mode :-

In computing protected mode, also called protected virtual address mode is an operational mode of x86 - compatible CPUs. It allows system s/w to use feature such as virtual memory. paging & safe multi - tasking design.

- Instruction to be used :-

Sgdt :- It stores the global descriptor table content to a registeR.

SIdt :- It Store the Interrupt descriptor table content to a registeR.

Sldt :- It stores the Local descriptoR table content to a registeR.

Str :- It stores the Segment selectoR from task registeR

Algorithm :-

1) Start
2) Display the message using write sys-call
3) Read CR0
4) checking PE bit, if 1 = protected mode
5) Load the number of digits to display.
6) perform required rotation for register content for particular type register (LDT, GDT)
7) convert it to ASCII
8) Display the Number
9) END

Conclusion :-

We learn & implement how to read & display contents pointed by GDTR, LDTR, IDTR using sgdt, sldt, SIdt instructions.