SUBJECT: MICROPROCESSOR LAB (MPL)	
NAME:	
CLASS: SE COMP	**ROLL NO.:**
SEMESTER: SEM-II	**YEAR: 2023-24**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED: Prof. G. B. Aochar	

Assignment No-06

Title:- Program to switch from real mode to protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers.

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

Objective:-

- To understand the assembly language program
- To understand 64 bit interrupt.
- To study GDTR, LDTR and IDTR and MSW

Outcome:-

- Students will be able to understand different assembly language instruction.
- Students will be able to write code for how to display the values of GDTR, LDTR, IDTR, TR and MSW Registers
- Students will be able to switch Processor Mode.

Prerequisite -

System call of Unix for Assembly language Program.

Hardware Requirement-

Desktop PC

Software Requirement-

Ubuntu 14.04,

Assembler: NASM

version 2.10.07
Linker: ld

Introduction:-

Four registers of the 80386 locate the data structures that control segmented memory management called as memory management registers:

1. GDTR : Global Descriptor Table Register

   These register point to the segment descriptor tables GDT. Before any segment register is changed in protected mode, the GDT register must point to a valid GDT. Initialization of the GDT and GDTR may be done in real-address mode. The GDT (as well as LDTs) should reside in RAM, because the processor modifies the accessed bit of descriptors. The instructions LGDT and SGDT give access to the GDTR.

2. LDTR : Local Descriptor Table Register

   These register point to the segment descriptor tables LDT. The LLDT instruction loads a linear base address and limit value from a six-byte data operand in memory into the LDTR. The SLDT instruction always store into all 48 bits of the six-byte data operand.

3. IDTR Interrupt Descriptor Table Register

   This register points to a table of entry points for interrupt handlers (the IDT). The LIDT instruction loads a linear base address and limit value from a six-byte data operand in memory into the IDTR. The SIDT instruction always store into all 48 bits of the six-byte data operand.

4. TR Task Register

   This register points to the information needed by the processor to define the current task. These registers store the base addresses of the descriptor tables (A descriptor table is simply a memory array of 8-byte entries that contain Descriptors and descriptor stores all the information about segment) in the linear address space and store the segment limits.

SLDT: Store Local Descriptor Table Register

Operation: DEST ← 48-bit BASE/LIMIT register contents;

Description: SLDT stores the Local Descriptor Table Register (LDTR) in the two-byte register or memory location indicated by the effective address operand. This register is a selector that points into the Global Descriptor Table. SLDT is used only in operating system software. It is not used in application programs.

Flags Affected: None

SGDT: Store Global Descriptor Table Register Operation:

DEST ← 48-bit BASE/LIMIT register contents;

Description: SGDT copies the contents of the descriptor table register the six bytes of memory indicated by the operand. The LIMIT field of the register is assigned to the first word at the effective address. If the operand-size attribute is 32 bits, the next three bytes are assigned the BASE field of the register, and the fourth byte is written with zero. The last byte is undefined. Otherwise, if the operand-size attribute is 16 bits, the next 4 bytes are assigned the 32-bit BASE field of the register. SGDT and SIDT are used only in operating system software; they are not used in application programs.

Flags Affected: None

SIDT: Store Interrupt Descriptor Table Register Operation:

DEST ← 48-bit BASE/LIMIT register contents;

Description: SIDT copies the contents of the descriptor table register the six bytes of memory indicated by the operand. The LIMIT field of the register is assigned to the first word at the effective address. If the operand-size attribute is 32 bits, the next three bytes are assigned the BASE field of the register, and the fourth byte is written with zero. The last byte is undefined. Otherwise, if the operand-size attribute is 16 bits, the next 4 bytes are assigned the 32-bit BASE field of the register. SGDT and SIDT are used only in operating system software; they are not used in application programs.

Flags Affected: None

Algorithm:

1. Display welcome message on terminal using macro disp.
2. Store most significant bit of CR0 in eax register.
3. Check the PE bit of CR0.
4. If PE=1 then display message "Processor is in Protected mode".
5. And if PE=0 then display message "Processor is in Real mode".

6. Then copies/stores the contents of GDT, IDT, LDT using sgdt, sidt, sldt instruction.
7. Display their contents using macro

Questions-

1. Explain System Address registers
2. Explain Segment selectors- LDTR and TR
3. Explain CRO register