Practical – 7

Problem Statement: Write X86/64 ALP to detect protected mode and display the values of DTR, LDTR, IDTR, TR and MSW Registers also identify CPU type using CPUID instruction.

Program:

```
section .data
nline db 10,10
nline_len: equ $-nline
colon db ":"
rmsg db 10,'Processor is in Real Mode...'
rmsg_len: equ $-rmsg
pmsg db 10,'Processor is in Protected Mode...'
pmsg_len: equ $-pmsg
gmsg db 10,"GDTR (Global Descriptor Table Register)
: "
gmsg_len: equ $-gmsg
imsg db 10,"IDTR (Interrupt Descriptor Table Register)
: "
imsg_len: equ $-imsg
lmsg db 10,"LDTR (Local Descriptor Table Register) : "
lmsg_len: equ $-lmsg
tmsg db 10,"TR (Task Register) : "
tmsg_len: equ $-tmsg
mmsg db 10,"MSW (Machine Status Word) : "
mmsg_len: equ $-mmsg
;---------------------------------------------------------------------
```

```
Section .bss
GDTR resw 3 ; 48 bits, so 3 words
IDTR resw 3
LDTR resw 1 ; 16 bits, so 1 word
TR resw 1
MSW resw 1
char_sum resb 4 ; 16-bits, so 4 digits
;------------------------------------------------------------------
;you can change the macros as per 64-bit convensions
%macro print 2
mov rax, 1
mov rdi, 1
mov rsi, %1
mov rdx, %2
syscall
%endmacro
%macro exit 0
mov rax, 60
mov rdi, 0
syscall
%endmacro
;------------------------------------------------------------------
; If U ARE MODIFYING 32-BIT PROGRAM then
; Check line by line and make all 'e' as 'r' and other modifications
; for 64-bit numbers
section .text
```

```
global _start
_start:
SMSW [MSW]
mov rax,[MSW]
ror rax,1 ; Check PE bit, if 1=Protected
Mode, else Real Mode
jc p_mode
print rmsg,rmsg_len
jmp next
p_mode:
print pmsg,pmsg_len
next:
SGDT [GDTR]
SIDT [IDTR]
SLDT [LDTR]
STR [TR]
; SMSW [MSW]
print gmsg, gmsg_len ;GDTR (Global Descriptor
Table Register)
; LITTLE ENDIAN SO
TAKE LAST WORD FIRST
mov ax,[GDTR+4] ; load value of GDTR[4,5] in ax
call disp16_proc ; display GDTR contents
mov ax,[GDTR+2] ; load value of GDTR[2,3] in ax
call disp16_proc ; display GDTR contents
print colon,1
```

```
mov ax,[GDTR+0] ; load value of GDTR[0,1] in ax
call disp16_proc ; display GDTR contents
print imsg, imsg_len ;IDTR (Interrupt Descriptor Table
Register)
mov ax,[IDTR+4]
call disp16_proc
mov ax,[IDTR+2]
call disp16_proc
print colon,1
mov ax,[IDTR+0]
call disp16_proc
print lmsg, lmsg_len ;LDTR (Local Descriptor Table
Register)
mov ax,[LDTR]
call disp16_proc
print tmsg, tmsg_len ;TR (Task Register)
mov ax,[TR]
call disp16_proc
print mmsg, mmsg_len ;MSW (Machine Status
Word)
mov ax,[MSW]
call disp16_proc
print nline, nline_len
exit
;---------------------------------------------------------------
disp16_proc:
```

```
mov rsi,char_sum+3 ; load last byte address of char_sum
buffer in rsi
mov rcx,4 ; number of digits
cnt: mov rdx,0 ; make rdx=0 (as in div instruction
rdx:rax/rbx)
mov rbx,16 ; divisor=16 for hex
div rbx
cmp dl, 09h ; check for remainder in RDX
jbe add30
add dl, 07h
add30:
add dl,30h ; calculate ASCII code
mov [rsi],dl ; store it in buffer
dec rsi ; point to one byte back
dec rcx ; decrement count
jnz cnt ; if not zero repeat
print char_sum,4 ; display result on screen
ret
;---------------------------------------------------------
```

Output:

```
atharva@atharva:~$ gedit lab7.asm
atharva@atharva:~$ nasm -f elf64 lab7.asm
atharva@atharva:~$ ld -o lab7 lab7.o
atharva@atharva:~$ ./lab7
Processor is in Protected Mode...
GDTR (Global Descriptor Table Register) : 00082000:007F
```

IDTR (Interrupt Descriptor Table Register) : 00000000:0FFF

LDTR (Local Descriptor Table Register) : 0000

TR (Task Register) : 0040

MSW (Machine Status Word) : 0033