**Name: Vaishnavi Ranjan**

**Roll no: 205C058**

**Batch: D**

# Assignment 6

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

Program –

%macro scall 4 mov rax,%1 mov rdi,%2 mov rsi,%3 mov rdx,%4

syscall

%endmacro

Section .data

"\*\*\*\*\* REGISTER CONTENTS \*\*\*\*\*" regmsg\_len: equ $-regmsg gmsg: db 0x0A,"Contents of GDTR : " gmsg\_len: equ $-gmsg lmsg: db 0x0A,"Contents of LDTR : " lmsg\_len: equ $-lmsg imsg: db 0x0A,"Contents of IDTR : " imsg\_len: equ $-imsg tmsg: db 0x0A,"Contents of TR : " tmsg\_len: equ $-tmsg mmsg: db 0x0A,"Contents of MSW : " mmsg\_len: equ $-mmsg realmsg: db "---- In Real mode. ----" realmsg\_len: equ $-realmsg protmsg: db "---- In Protected Mode. ----" protmsg\_len: equ $-protmsg cnt2:db 04H

newline: db 0x0A

Section .bss g: resd 1 resw 1 l: resw 1 idtr: resd 1 resw 1 msw: resd 1 tr: resw 1 value :resb 4 Section .text global \_start \_start: scall 1,1,title,title\_len

smsw [msw] mov

eax,dword[msw]

bt eax,0 jc next scall 1,1,realmsg,realmsg\_len jmp EXIT next: scall 1,1,protmsg,protmsg\_len scall 1,1, regmsg,regmsg\_len ;printing register contents scall 1,1,gmsg,gmsg\_len SGDT [g] mov bx, word[g+4] call HtoA mov bx,word[g+2] call HtoA mov bx, word[g] call HtoA

;--- LDTR CONTENTS---- find valid values for all labels after 1001 passes, giving up. scall 1,1, lmsg,lmsg\_len SLDT [l] mov bx,word[l] call HtoA

;---- IDTR Contents ------- scall 1,1,imsg,imsg\_len SIDT [idtr] mov bx, word[idtr+4] call HtoA mov bx,word[idtr+2] call HtoA mov bx, word[idtr] call HtoA

;---- Task Register Contents -0----- scall 1,1, tmsg,tmsg\_len mov bx,word[tr] call HtoA ;------- Content of MSW --------- scall 1,1,mmsg,mmsg\_len mov bx, word[msw+2] call HtoA mov bx, word[msw] call HtoA scall 1,1,newline,1 EXIT: mov rax,60 mov rdi,0 syscall

;------HEX TO ASCII CONVERSION METHOD ----------------

HtoA: ;hex\_no to be converted is in bx //result is stored in rdi/user defined variable mov rdi,value mov byte[cnt2],4H aup: rol bx,04 mov cl,bl and cl,0FH cmp cl,09H jbe

ANEXT ADD cl,07H ANEXT: add cl, 30H mov byte[rdi],cl INC rdi dec byte[cnt2] JNZ aup scall 1,1,value,4

ret

**Output**

