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.

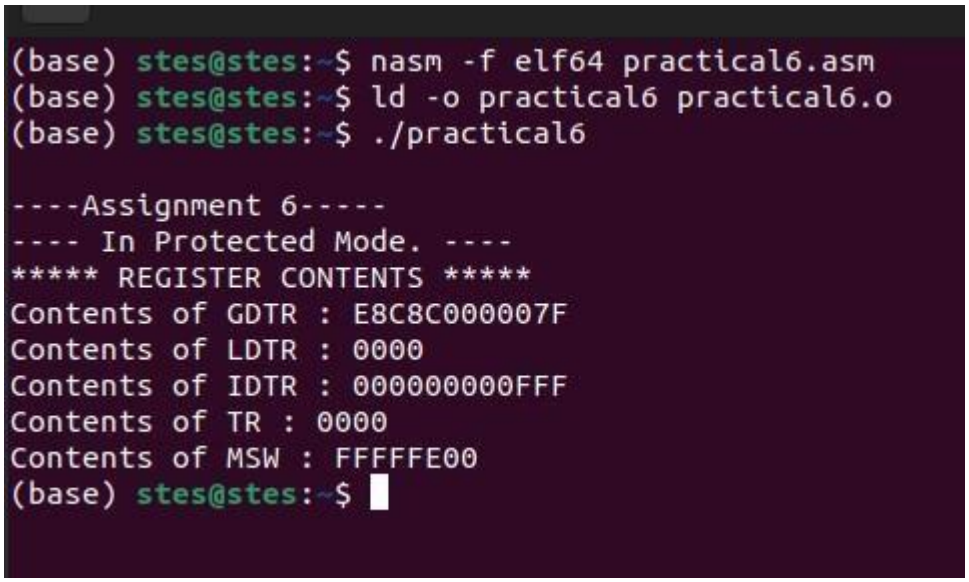
Program –

```
%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


(base) stes@stes:~$ nasm -f elf64 practical6.asm
(base) stes@stes:~$ ld -o practical6 practical6.o
(base) stes@stes:~$ ./practical6
----Assignment 6-----
---- In Protected Mode. ----
***** REGISTER CONTENTS *****
Contents of GDTR : E8C8C000007F
Contents of LDTR : 0000
Contents of IDTR : 000000000FFF
Contents of TR : 0000
Contents of MSW : FFFFFE00
(base) stes@stes:~$