ASSIGNMENT NO 8

PROBLEM STATEMENT:

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.

SOURCE CODE:

```
section .data
        menumsg db 10,10,'***Nonoverlap block transfer***',10
                db 10,'1.Block transfer without string '
                db 10,'2.Block transfer with string '
                db 10,'3.exit   '
        menumsg_len equ $-menumsg
        wrmsg db 10,10,'Wrong choice entered',10,10
        wrmsg_len equ $-wrmsg
        bfrmsg db 10,'**Block contents before transfer: '
        bfrmsg_len equ $-bfrmsg
        afrmsg db 10,'**Block contents after transfer:'
        afrmsg_len equ $-afrmsg
        srcmsg db 10,'*_*Source block contents   '
        srcmsg_len equ $-srcmsg
        dstmsg db 10,'*_*Destination block contents   '
        dstmsg_len equ $-dstmsg
        srcblk db 01h,02h,03h,04h,05h
```

```
        dstblk times 5 db 0             ;destination block is defined 5 times
        cnt equ 05
        spacechar db 20h
        lfmsg db 10,10

section .bss
        optionbuff resb 02
        dispbuff resb 02

%macro dispmsg 2
        mov eax,04
        mov ebx,01
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

%macro accept 2
        mov eax,03
        mov ebx,00
        mov ecx,%1
        mov edx,%2
        int 80h
%endmacro

section .text
```

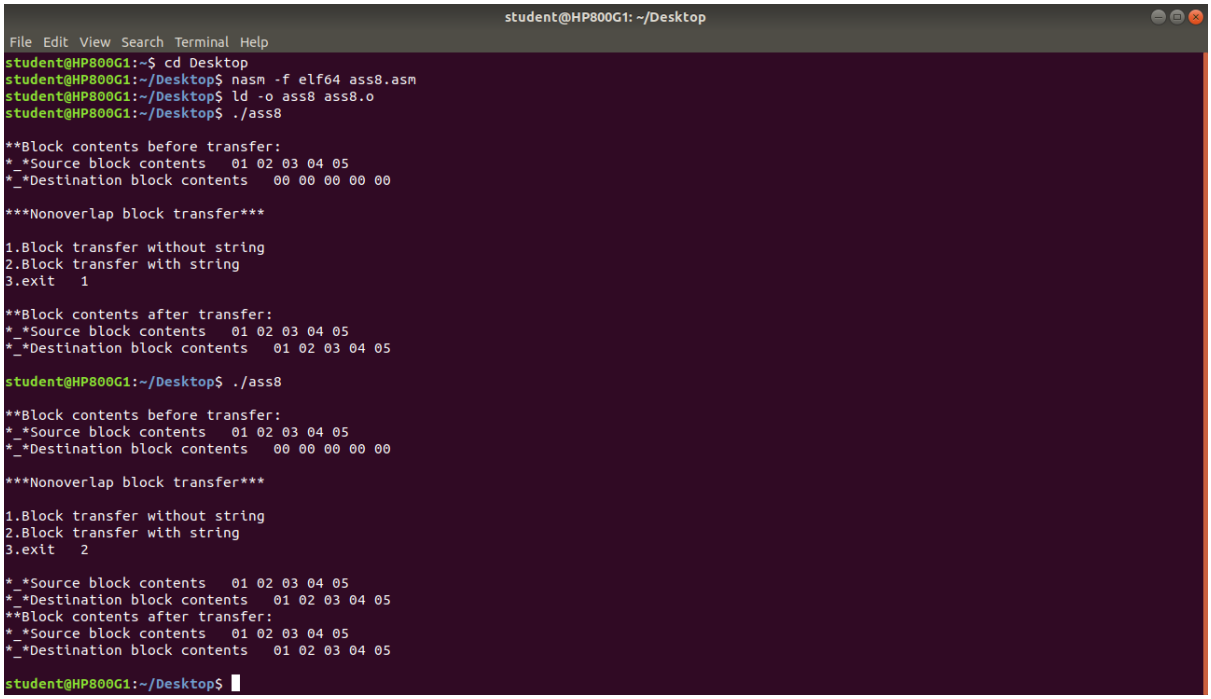
```
global _start
_start:
        dispmsg bfrmsg,bfrmsg_len
        call show
        menu:
                dispmsg menumsg,menumsg_len
                accept optionbuff,02
                cmp byte [optionbuff],'1'
                jne case2
                call wos                        ;wos=With Out String
                jmp exit1

        case2:
                cmp byte [optionbuff],'2'
                jne case3
                call ws                         ;ws=with string
                jmp exit1

        case3:
                cmp byte [optionbuff],'3'
                je exit
                dispmsg wrmsg,wrmsg_len
                jmp menu

        exit1:
                dispmsg afrmsg,afrmsg_len
```

```
        call show
        dispmsg lfmsg,2
exit:
        mov eax,01
        mov ebx,00
        int 80h

dispblk:
        mov rcx,cnt

rdisp:
        push rcx
        mov bl,[esi]
        call disp8
        inc esi
        dispmsg spacechar,1
        pop rcx
        loop rdisp
ret

wos:
        mov esi,srcblk
        mov edi,dstblk
        mov ecx,cnt
        x:
                mov al,[esi]
```

```
        mov [edi],al
        inc esi
        inc edi
        loop x
        ret

ws:
    mov esi,srcblk
    mov edi,dstblk
    mov ecx,cnt
    cld                         ;clear direction flag
    rep movsb

show:
    dispmsg srcmsg,srcmsg_len
    mov esi,srcblk
    call dispblk
    dispmsg dstmsg,dstmsg_len
    mov esi,dstblk
    call dispblk
    ret

disp8:
    mov ecx,02
    mov edi,dispbuff
    dub1:
```

```
        rol bl,4
        mov al,bl
        and al,0fh
        cmp al,09h
        jbe x1
        add al,07
x1:
        add al,30h
        mov [edi],al
        inc edi
        loop dub1
        dispmsg dispbuff,3
ret
```

OUTPUT:


student@HP800G1: ~/Desktop
File Edit View Search Terminal Help
student@HP800G1:~$ cd Desktop
student@HP800G1:~/Desktop$ nasm -f elf64 ass8.asm
student@HP800G1:~/Desktop$ ld -o ass8 ass8.o
student@HP800G1:~/Desktop$ ./ass8
**Block contents before transfer:
*_*Source block contents 01 02 03 04 05
*_*Destination block contents 00 00 00 00 00
Nonoverlap block transfer
1.Block transfer without string
2.Block transfer with string
3.exit 1
**Block contents after transfer:
*_*Source block contents 01 02 03 04 05
*_*Destination block contents 01 02 03 04 05
student@HP800G1:~/Desktop$ ./ass8
**Block contents before transfer:
*_*Source block contents 01 02 03 04 05
*_*Destination block contents 00 00 00 00 00
Nonoverlap block transfer
1.Block transfer without string
2.Block transfer with string
3.exit 2
*_*Source block contents 01 02 03 04 05
*_*Destination block contents 01 02 03 04 05
**Block contents after transfer:
*_*Source block contents 01 02 03 04 05
*_*Destination block contents 01 02 03 04 05
student@HP800G1:~/Desktop$