

# ML Assignment

**Name:** Jagruti Ravindra Patil

**Roll No:** 2089

**Batch:** S4 Comp

## Assignment 7:

**Code:**

```
section .data

    msg1: db 'GDTR contents :', 0xa
    len1: equ $ - msg1

    msg2: db 'LDTR contents:', 0xa
    len2: equ $ - msg2

    msg3: db 'IDTR contents :', 0xa
    len3: equ $ - msg3

    msg4: db 'TR contents:', 0xa
    len4: equ $ - msg4

    msg5: db 'MSW contents:', 0xa
    len5: equ $ - msg5

    msg6: db 'We are in protected mode.!!', 0xa
    len6: equ $ - msg6

    msg7: db ' ', 0xa
    len7: equ $ - msg7

    msg8: db 'We are not in protected mode.!!', 0xa
    len8: equ $ - msg8

    msg9: db ' : ', 0xa
    len9: equ $ - msg9

section .bss

    gdt: resd 01
        resw 01

    ldt: resw 01

    idt: resd 01
```

```
    resw 01  
tr: resw 01  
msw: resw 01  
result: resw 01
```

```
section .text  
global _start
```

```
_start:  
    smsw [msw]  
    sgdt [gdt]  
    sldt [ldt]  
    sidt [idt]  
    str [tr]
```

```
    mov ax, [msw]  
    bt ax, 0  
    jc next
```

```
    mov rax, 1  
    mov rdi, 1  
    mov rsi, msg8  
    mov rdx, len8  
    syscall  
    jmp exit
```

```
next:  
    mov rax, 1  
    mov rdi, 1  
    mov rsi, msg6  
    mov rdx, len6  
    syscall
```

```
; GDTR
```

```
mov rax, 1
mov rdi, 1
mov rsi, msg1
mov rdx, len1
syscall
mov bx, word [gdt + 4]
call HtoA
mov bx, word [gdt + 2]
call HtoA
mov rax, 1
mov rdi, 1
mov rsi, msg9
mov rdx, len9
syscall
mov bx, word [gdt]
call HtoA
```

; LDTR

```
mov rax, 1
mov rdi, 1
mov rsi, msg7
mov rdx, len7
syscall
mov rax, 1
mov rdi, 1
mov rsi, msg2
mov rdx, len2
syscall
mov bx, word [ldt]
call HtoA
```

; IDTR

```
mov rax, 1
mov rdi, 1
```

```
mov rsi, msg7
mov rdx, len7
syscall
mov rax, 1
mov rdi, 1
mov rsi, msg3
mov rdx, len3
syscall
mov bx, word [idt + 4]
call HtoA
mov bx, word [idt + 2]
call HtoA
mov rax, 1
mov rdi, 1
mov rsi, msg9
mov rdx, len9
syscall
mov bx, word [idt]
call HtoA
```

```
; TR
```

```
mov rax, 1
mov rdi, 1
mov rsi, msg7
mov rdx, len7
syscall
mov rax, 1
mov rdi, 1
mov rsi, msg4
mov rdx, len4
syscall
mov bx, word [tr]
call HtoA
```

```
; MSW
mov rax, 1
mov rdi, 1
mov rsi, msg7
mov rdx, len7
syscall
mov rax, 1
mov rdi, 1
mov rsi, msg5
mov rdx, len5
syscall
mov bx, word [msw]
call HtoA
```

```
; EXIT
exit:
    mov rax, 60
    mov rdi, 0
    syscall
```

```
HtoA:
    mov rcx, 4
    mov rdi, result
```

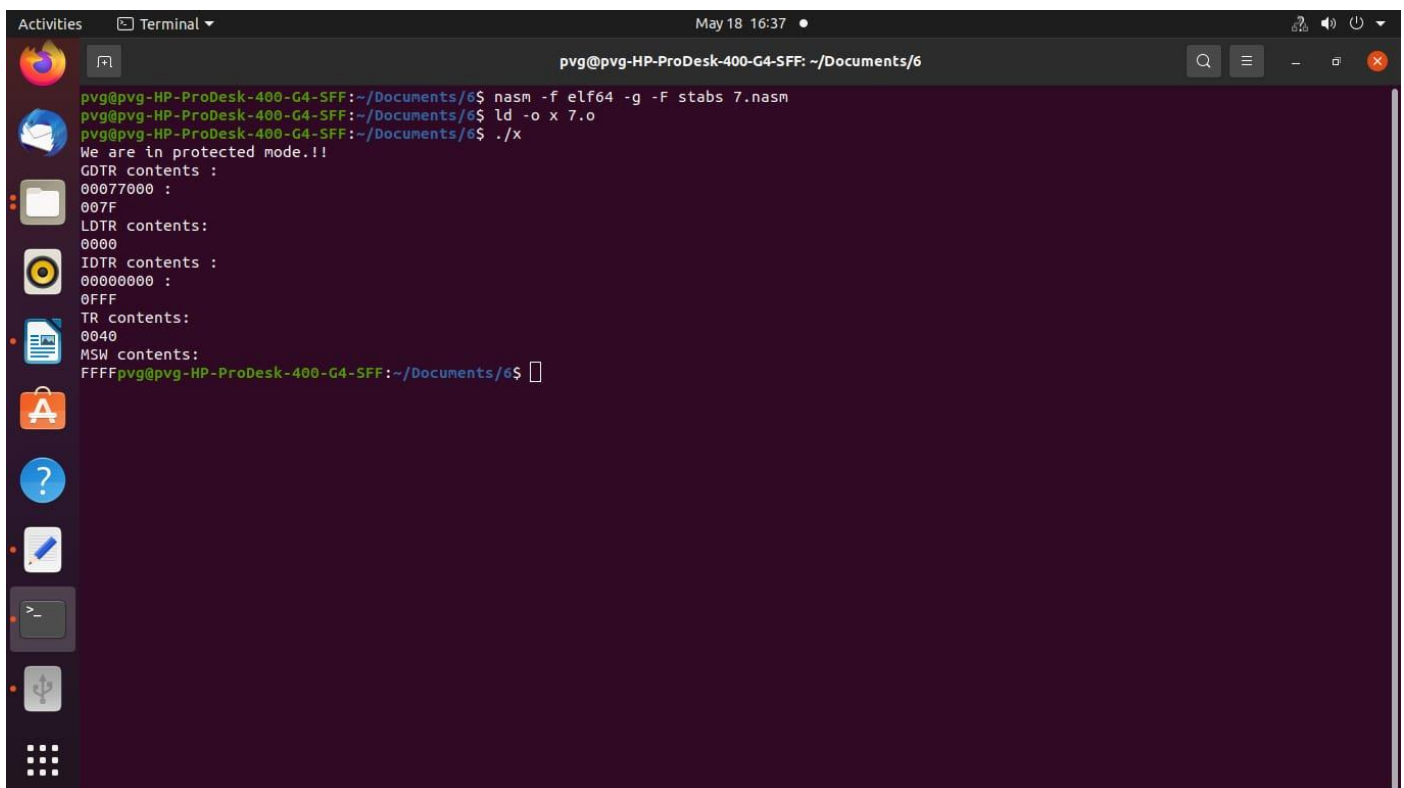
```
dup1:
    rol bx, 4
    mov al, bl
    and al, 0fh
    cmp al, 09h
    jg p3
    add al, 30h
    jmp p4
```

```
p3:
    add al, 37h
```

```
p4:
```

```
mov [rdi], al
inc rdi
loop dup1
mov rax, 1
mov rdi, 1
mov rsi, result
mov rdx, 4
syscall
ret
```

## Output:



The screenshot shows a terminal window titled "Terminal" with the following content:

```
pvg@pvg-HP-ProDesk-400-G4-SFF: ~/Documents/6
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ nasm -f elf64 -g -F stabs 7.nasm
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ ld -o x 7.o
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$ ./x
We are in protected mode.!!
GDTR contents :
00077000 :
007F
LDTR contents:
0000
IDTR contents :
00000000 :
0FFF
TR contents:
0040
MSW contents:
FFFFpvg@pvg-HP-ProDesk-400-G4-SFF:~/Documents/6$
```

The terminal output displays the assembly process using NASM and LD, followed by the execution of the resulting binary. The program prints "We are in protected mode.!!" and then displays the contents of the GDTR, LDTR, IDTR, TR, and MSW registers in hexadecimal format.