Practical No:7

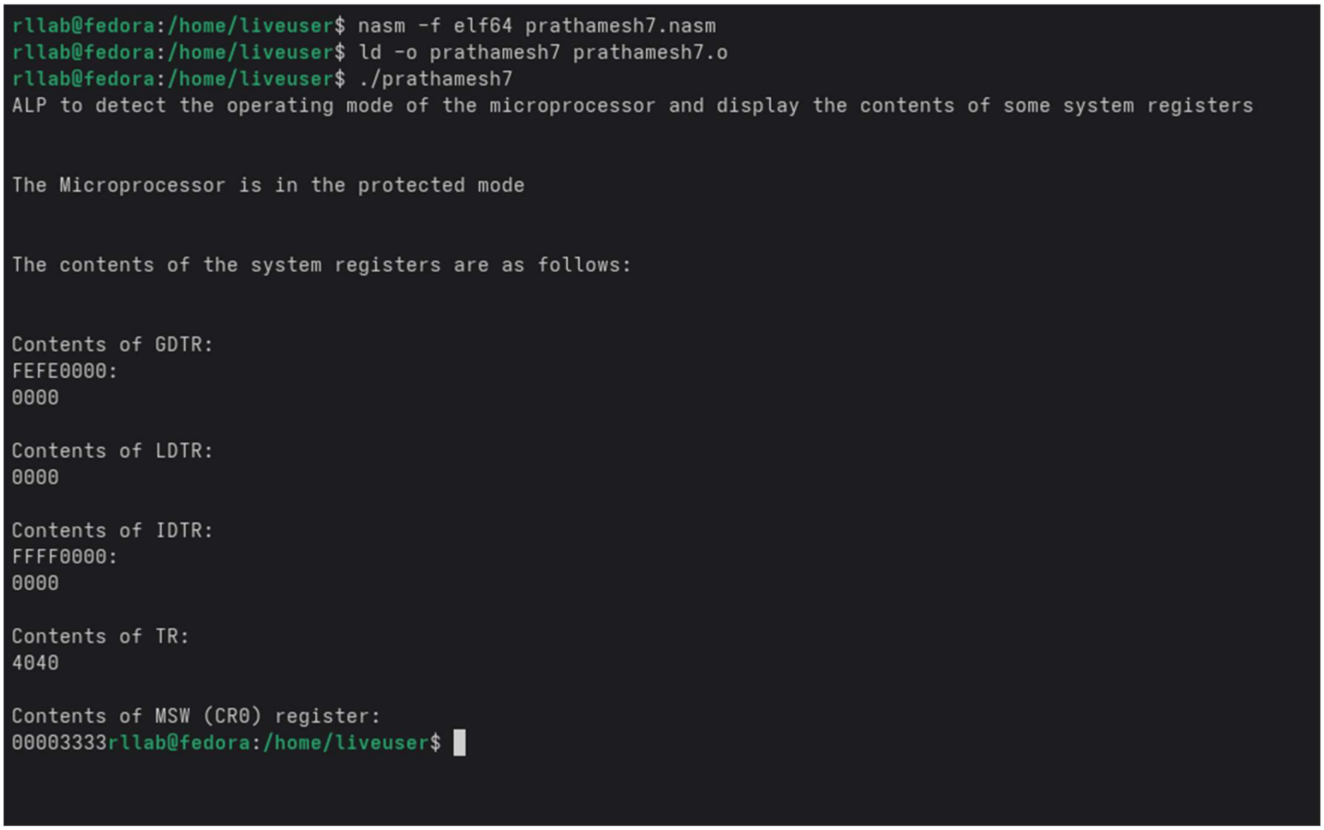
```
section .data
introMsg db "ALP to detect the operating mode of the microprocessor and display the contents
of some system registers", 10
introMsgLen equ $ - introMsg
gdtrMsg db 10, 10, "Contents of GDTR: ", 10
gdtrMsgLen equ $ - gdtrMsg
ldtrMsg db 10, 10, "Contents of LDTR: ", 10
ldtrMsgLen equ $ - ldtrMsg
idtrMsg db 10, 10, "Contents of IDTR: ", 10
idtrMsgLen equ $ - idtrMsg
trMsg db 10, 10, "Contents of TR: ", 10
trMsgLen equ $ - trMsg
mswMsg db 10, 10, "Contents of MSW (CR0) register: ", 10
mswMsgLen equ $ - mswMsg
protectedMsg db 10, 10, "The Microprocessor is in the protected mode", 10
protectedMsgLen equ $ - protectedMsg
proMsg db 10, 10, "The contents of the system registers are as follows:", 10
proMsgLen equ $ - proMsg
realMsg db 10, 10, "The Microprocessor is in the real mode", 10
realMsgLen equ $ - realMsg
colon db ":", 10
colonLen equ $ - colon
section .bss
gdtr resd 1
resw 1
ldtr resw 1
idtr resd 1
resw 1
tr resw 1
```

```
msw resd 1
result resb 4
%macro write 2
mov rax, 1
mov rdi, 1
mov rsi, %1
mov rdx, %2
syscall
%endmacro
section  .text
global _start
_start:
write introMsg, introMsgLen
smsw eax
bt eax, 0
jc protected_mode
write realMsg, realMsgLen
jmp endOfProgram
protected_mode:
write protectedMsg, protectedMsgLen
write proMsg, proMsgLen
sgdt [gdtr]
sldt [ldtr]
str [tr]
smsw [msw]
write gdtrMsg, gdtrMsgLen
mov bx, [gdtr+4]
call disp
mov bx, [gdtr+2]
call disp
```

```
write colon, colonLen
mov bx, [gdtr]
call disp
write ldtrMsg, ldtrMsgLen
mov bx, [ldtr]
call disp
write idtrMsg, idtrMsgLen
mov bx, [idtr+4]
call disp
mov bx, [idtr+2]
call disp
write colon, colonLen
mov bx, [idtr]
call disp
write trMsg, trMsgLen
mov bx, [tr]
call disp
write mswMsg, mswMsgLen
mov bx, [msw+2]
call disp
mov bx, [msw]
call disp
endOfProgram:
mov rax, 60
mov rdi, 0
syscall
disp:
mov rdi, result ;point rdi to result variable
mov cx,04 ;load count of rotation in cl
up1:
```

```
rol bl,04 ;rotate number left by four bits

mov dl,bl ;move lower byte in dl

and dl,0fh ; get only LSB

cmp dl,09h ;compare with 39h

jg add_37 ;if greater than 39h skip add 37

add dl,30h

jmp skip1 ;else add 30

add_37: add dl,37h

skip1: mov [rdi],dl ;store ascii code in result variable

inc rdi ;point to next byte

dec cx ;decrement the count of digits to display

jnz up1 ;if not zero jump to repeat

write result , 4

ret
```

output



```
rllab@fedora:/home/liveuser$ nasm -f elf64 prathamesh7.nasm
rllab@fedora:/home/liveuser$ ld -o prathamesh7 prathamesh7.o
rllab@fedora:/home/liveuser$ ./prathamesh7
ALP to detect the operating mode of the microprocessor and display the contents of some system registers


The Microprocessor is in the protected mode


The contents of the system registers are as follows:


Contents of GDTR:
FEFE0000:
0000

Contents of LDTR:
0000

Contents of IDTR:
FFFF0000:
0000

Contents of TR:
4040

Contents of MSW (CR0) register:
00003333rllab@fedora:/home/liveuser$ 
```