MICROPROCESSOR ASSIGNMENTS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIVISION : B

ROLL NO 222006

PROBLEM STATEMENT :

Write X86 ALP to find, a) Number of Blank spaces b) Number of lines c) Occurrence of a particular character. Accept the data from the text file. The text file has to be accessed during Program\_1 execution and write FAR PROCEDURES in Program\_2 for the rest of the processing. Use of PUBLIC and EXTERN directives is mandatory.

CODE :

extern proc\_far

global fh, char, buff, length

%macro scall 4

mov rax, %1

mov rdi, %2

mov rsi, %3

mov rdx, %4

syscall

%endmacro

;--------------

section .data

msg db 10, "ENTER FILE NAME >> "

msg\_l equ $-msg

msg1 db 10, "ENTER THE CHARACTER TO BE SEARCHED "

msg1\_l equ $-msg1

err\_msg db 10, "FILE NOT FOUND "

err\_msg\_l equ $-err\_msg

;-----------

section .bss

filename resb 50

buff resb 1024

buff\_len equ $-buff

char resb 2

fh resq 1

length resq 1

;----------

section .txt

global \_start

\_start :

scall 1, 1, msg, msg\_l

scall 0, 0, filename, 50

dec rax

mov byte[filename + rax], 0

scall 1, 1, msg1, msg1\_l

scall 0, 0, char, 2

scall 2, filename, 2, 0777o

cmp rax, -1h

jle error

mov [fh], rax

scall 0, [fh], buff, buff\_len

mov [length], rax

call proc\_far

jmp exit

error : scall 1, 1, err\_msg, err\_msg\_l

jmp exit

exit : mov rax, 60

mov rdi, 0

syscall

%macro scall 4

mov rax, %1

mov rdi, %2

mov rsi, %3

mov rdx, %4

syscall

%endmacro

;----------

section .data

smsg db 10, "no of spaces = "

smsg\_l equ $-smsg

lmsg db 10, "no of lines = "

lmsg\_l equ $-lmsg

cmsg db 10, "number of characters = "

cmsg\_l equ $-cmsg

nline db 10

nline\_l equ $-nline

;----------

section .bss

scnt resq 1

ccnt resq 1

lcnt resq 1

temp resb 2

;-----------

global proc\_far

extern fh, char, buff, length

section .txt

global main

main :

proc\_far :

mov rax, 0

mov rbx, 0

mov rcx, 0

mov rdx, 0

mov rsi, 0

mov rdi, 0

mov bl, [char]

mov rsi, buff

mov rcx, [length]

back : mov al, [rsi]

cmp al, 20h

jne line

inc byte[scnt]

jmp next

line : cmp al, 0Ah

jne character

inc byte[lcnt]

jmp next

character : cmp al, bl

jne next

inc byte[ccnt]

next : inc rsi

dec rcx

jnz back

scall 1, 1, smsg, smsg\_l

mov rbx, [scnt]

call display\_proc

scall 1, 1, temp,16

scall 1, 1, cmsg, cmsg\_l

mov rbx, [ccnt]

call display\_proc

scall 1, 1, temp, 16

scall 1, 1, lmsg, lmsg\_l

mov rbx, [lcnt]

call display\_proc

scall 1, 1, temp, 16

scall 3, [fh], 0, 0

scall 1, 1, nline, nline\_l

ret

display\_proc :

mov rcx, 2

mov al, bl

mov rsi, temp

back2 : rol al, 4

mov dl, al

and dl, 0fh

cmp dl, 09h

jbe c1

add dl, 07h

c1 : add dl, 30h

mov [rsi], dl

inc rsi

dec rcx

jnz back2

ret

OUTPUT :

