**Title:-Program to analyze the difference between near and far procedure to find number of lines, blank spaces & occurance of character.**

**Assignment Name: - 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.**

**P1.asm**

section .data  
  
global msg6,len6,scount,ncount,chacount,new,new\_len  
  
fname: db 'abc.txt',0  
  
msg: db "File opened successfully",0x0A  
len: equ $-msg  
  
msg1: db "File closed successfully",0x0A  
len1: equ $-msg1  
  
msg2: db "Error in opening file",0x0A  
len2: equ $-msg2  
  
msg3: db "Spaces:",0x0A  
len3: equ $-msg3  
  
msg4: db "NewLines:",0x0A  
len4: equ $-msg4  
  
msg5: db "Enter character",0x0A  
len5: equ $-msg5  
  
msg6: db "No of occurances:",0x0A  
len6: equ $-msg6  
  
new: db "",0x0A  
new\_len: equ $-new  
  
scount: db 0  
ncount: db 0  
ccount: db 0  
chacount: db 0  
  
section .bss  
  
global cnt,cnt2,cnt3,buffer  
  
fd: resb 17  
buffer: resb 200  
buf\_len: resb 17  
cnt: resb 2  
cnt2: resb 2  
cnt3: resb 2  
cha: resb 2  
  
%macro scall 4  
mov rax,%1  
mov rdi,%2  
mov rsi,%3  
mov rdx,%4  
syscall  
%endmacro  
  
section .text  
global \_start  
\_start:  
  
extern spaces,enters,occ  
  
mov rax,2  
mov rdi,fname  
mov rsi,2  
mov rdx,0777  
syscall  
  
mov qword[fd],rax  
BT rax,63  
jc next  
scall 1,1,msg,len  
jmp next2  
next:  
scall 1,1,msg2,len2  
  
  
next2:  
scall 0,[fd],buffer, 200  
mov qword[buf\_len],rax  
mov qword[cnt],rax  
mov qword[cnt2],rax  
mov qword[cnt3],rax  
  
  
  
scall 1,1,msg3,len3  
call spaces  
  
scall 1,1,msg4,len4  
call enters  
  
scall 1,1,msg5,len5  
scall 0,1,cha,2  
mov bl, byte[cha]  
call occ  
jmp exit  
  
  
  
exit:  
mov rax,60  
mov rdi,0  
syscall

**P2.asm**

section .data  
  
  
extern msg6,len6,scount,ncount,chacount,new,new\_len  
  
  
  
section .bss  
  
  
extern cnt,cnt2,cnt3,scall,buffer  
  
  
  
%macro scall 4  
  
mov rax,%1  
  
mov rdi,%2  
  
mov rsi,%3  
  
mov rdx,%4  
  
syscall  
  
%endmacro  
  
  
  
section .text  
  
global main2  
  
main2:  
  
  
global spaces,enters,occ  
  
  
  
spaces:  
  
mov rsi,buffer  
  
  
up:  
  
mov al, byte[rsi]  
  
cmp al,20H  
  
je next3  
  
inc rsi  
  
dec byte[cnt]  
  
jnz up  
  
  
jmp next4  
  
  
next3:  
  
inc rsi  
  
inc byte[scount]  
  
dec byte[cnt]  
  
jnz up  
  
  
next4:  
  
add byte[scount], 30h  
  
scall 1,1,scount, 2  
  
scall 1,1,new,new\_lenret  
  
  
  
enters:  
  
mov rsi,buffer  
  
up2:  
  
mov al, byte[rsi]  
c  
mp al,0AH  
  
je next5  
  
inc rsi  
  
dec byte[cnt2]  
  
jnz up2  
  
  
jmp next6  
  
next5:  
  
inc rsi  
  
inc byte[ncount]  
  
dec byte[cnt2]  
  
jnz up2  
  
  
next6:  
  
add byte[ncount], 30h  
  
scall 1,1,ncount, 2  
  
scall 1,1,new,new\_len  
ret  
  
  
  
occ:  
  
mov rsi,buffer  
  
up3:  
  
mov al, byte[rsi]  
  
cmp al,bl  
  
  
je next7  
  
inc rsi  
  
dec byte[cnt3]  
  
jnz up3  
  
jmp next8  
  
  
next7:  
  
inc rsi  
  
inc byte[chacount]  
  
dec byte[cnt3]  
  
jnz up3  
  
  
next8:  
  
add byte[chacount], 30h  
  
scall 1,1,msg6,len6  
  
scall 1,1,chacount, 1  
  
scall 1,1,new,new\_len  
ret

**abc.txt**

Hello  
Welcome to Pune  
This is sumit  
  
  
  
  
;\*\*\*\*\*\*\*output\*\*\*\*\*\*\*  
; nasm -f elf64 p1 p1.asm  
; nasm -f elf64 p2 p2.asm  
; ld -o p p1.o p2.o  
; ./p  
;File opened successfully  
;Spaces:  
;6  
;NewLines:  
;3  
;Enter character  
;s  
;No of occurances:  
;3