Experiments No:12

Title: To find the number of lines,blank spaces.

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.

Objective:

- To understand assembly language programming instruction set
  To understand different assembler directives with example
- To apply instruction set for implementing X86/64 bit assembly language programs

Outcomes: On completion of this practical ,students will be able to

C218.1: Understand and apply various addressing modes and instruction set to implement assembly language programs

Hardware Requirement: NA

Software Requirement: OS: Ubuntu Assembler: NASM version 2.10.07 Linker: ld

Theory Contents :Near Procedure Far Procedure

Explanation:

Open given text file. Read the content of file and store it in a buffer. Call far procedure which will calculate the number of blank spaces, lines and occurrence of a particular character from the buffer.

Assembler Directives Used: (Explain it by your own)

- **Extern:**

- **Global:**

Instructions:

Input: Text File Output: Display of-

1. Number of Blank spaces
2. Number of lines
3. Occurrence of a particular character.

Main Algorithm:

A1: Algorithm for program_1

i. Start

ii. Initialize all the sections needed in programming

iii. Display “Enter file name” message using Print macro expansion

iv. Accept file name using Accept macro and store in filename buffer

v. Display “Enter character to search” message with the expansion of Print macro

vi. Read character using Accept macro expansion

vii. Open file using fopen macro

viii. Compare RAX with -1H if equal then display error message “Error in Opening File” with Print macro expansion else go to step ix

ix. Read content of opened file in buffer

x. Store file length in abuf_len

xi. Call far_procedure

xii. xii. Stop

Macros: Macro 1

i. Name : Print

ii. Purpose: to display the messages by replacing the whole code by simple macro declaration

iii. I/P: sys_write call Number i.e eax=___, File descriptor (for Standard output ebx=1), Buffer Address in rsi, and length of Buffer in rdx. Then Call int 80h.

Macro 2

1. Name : Accept
2. Purpose: to accept input from the user by replacing the whole code by simple macro Declaration

3. I/P: sys_read call Number i.e eax=____, File descriptor (for Standard input rdi=0), Buffer Address in ecx, and length of Buffer in edx then call int 80h

Macro 3

1. Name : fopen
2. Purpose: to open a file in given mode
3. I/P: sys_write call Number i.e eax= File name in ebx, Mode of file in ecx (R=0,W=1,RW=2), and file permission in edx then call int 80h.

Macro 4

1. Name: fread

2. Purpose: to read the content of file

4. I/P: sys_read call Number i.e eax=3, File descriptor in ebx , Buffer Address in ecx, and Length of Buffer in edx. Then Call int 80h

Macro 5

1. Name: fclose

2. Purpose: to close opened file

3. I/P: sys_read call Number i.e eax=6, File handler in ebx. Then Call int 80h.

Procedure: 1

1. Name: far_procedure
2. Purpose: to count 1. Number of Blank spaces 2. Number of lines 3. Occurrence of a particular character.
3. I/P : Content stored in buffer

4. Algorithm for Procedures

i. Start

ii. Load effective address of buffer in RSI

iii. Load content of abuf_len in ECX

iv. Load content of char in BL

v. Move value of RSI in AL

vi. Compare AL with 20H (ASCII value of space) if not equal then go to step

vii else increment content of scount vii. Compare AL with 10H (ASCII value of line)

if not equal then go to step

viii else increment content of ncount viii. Compare AL with BL if not equal then go to step ix else increment content of ccount

ix. Increment RSI

x. Repeat from step

vi if RCX is not equal to zero

xi. Display "Number of space" message with the expansion of Print macro.

xii. Move content of scount in EBX

xiii. Call display8num procedure

xiv. Display "Number of lines" message with the expansion of Print macro.

xv. Move content of ncount in EBX

xvi. Call display8num procedure

xvii. Display "Number of Occurrence of Character" message with the expansion of Print macro. xviii. Move content of ccount in EBX

xix. Call display8num procedure

xx. Ret

xxi. Stop

Procedure: 2

1. Name: display8num

2. Purpose: Convert

2 digit hex number into 2 ASCII character to display Positive and Negative Number count on Standard output (stdout).

3. I/P : bl=pcnt/ncnt

4. Algorithm for Procedures

a. Move RSI with effective address of dispbuff.

b. Initialize rcx by 2

c. Rotate the contents of bl to the left side by 4 bits.

d. Move the contents of bl into al

e. And the contents of al with 0fH

f. Compare al with 09h

i. If al is below or equal then add 30H in al

ii. Else add 37H in al

g. Move the content of al into memory pointed by ESI

h. Increment ESI i. Repeat from step c to h until rcx is not equal to 0

Conclusions:

Assembly Level Program to find,

a) Number of Blank spaces

b) Number of lines

c) Occurrence of a particular character is assembled and executed successfully.

Assignment Questions

Q1 Explain 'EXTERN' and 'EXTRN' directive.

Q2 Explain 'GLOBAL' and 'PUBLIC' directive.

Q3 How far procedure is called in masm?

Q4 How you counted the occurrences of character in the given string? Explain logic.

Q5 How you assembled and linked the source files?

Q6 Explain FAR call and return ?

Q7 Explain difference between "near" and "far" procedure.

Q8 Write Down the ASCII of Space, Enter ?

MPL Practical Oral Question Bank

Sr No	BL	Questions	Oral 1	Oral 2 (improvement)	Remark
1	1	What is'GLOBAL' and 'PUBLIC' directive. ?			
2	1	What is difference between Far and Near Procedure?			
3	1	What is use Far Procedure?			

Sign of Student