

SUBJECT: MICROPROCESSOR LAB (MPL)	
NAME:	
CLASS: SE COMP	ROLL NO.:
SEMESTER: SEM-II	YEAR: 2023-24
DATE OF PERFORMANCE:	DATE OF SUBMISSION:
EXAMINED: Prof. G. B. Aochar	

Assignment No-09

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

Objective-

- To understand the assembly language program
- To understand 64 bit interrupt.
- To understand FAR PROCEDURES and PUBLIC and EXTERN directives.

Outcome-

- Students will be able to understand different assembly language instruction.
- Students will be able to analyze the difference between near and far procedure to find number of lines, blank spaces & occurrence of character using nasm.

Prerequisite -

System call of Unix for Assembly language Program.

Hardware Requirement-

Desktop PC

Software Requirement-

Ubuntu 14.04,

Assembler: NASM version 2.10.07

Linker: ld

Introduction:-

EXTERN

Informs the assembler that the names, procedures, labels declared after this directive have already been defined in some other assembly language module while in the other module where the names, procedures & labels actually appear, they must be declared Global using GLOBAL directive

GLOBAL

The labels, variables, constants or procedures declared GLOBAL may be used by other modules of program. Once the variable is declared GLOBAL, it can be used by any module in the program.

FAR

Used to declare the procedure far from the segment from where we are calling it.

PUBLIC

Used to declare procedure publically

When executing a far call, the processor performs these actions:

1. Pushes current value of the CS register on the stack.
2. Pushes the current value of the IP register on the stack.
3. Loads the base address of the segment that contains the called procedure in the CS register.
4. Loads the offset of the called procedure in the IP register.
5. Begins execution of the called procedure.

When executing a far return, the processor does the following:

1. Pops the top-of-stack value (the return instruction pointer) into the IP register.
2. Pops the top-of-stack value (the segment selector for the code segment being returned to) into the CS register.
3. (If the RET instruction has an optional n argument.) Increments the stack pointer by the number of bytes specified with the n operand to release parameters from the stack.
4. Resumes execution of the calling procedure.

Algorithms:-

Algorithm to Read a file

- 1) Call file in the program
- 2) put a pointer to the stack
- 3) Get file name from the stack.

- 4) Call interrupt to open the file.
- 5) The file descriptor is now available in eax. Test that descriptor.
- 6) If file descriptor is not valid then close the file and exit from program.
- 7) If valid file descriptor then copy the file contents into buffer.
- 8) It will display the no of blank spaces in the file.
- 9) It will display the no of lines in the file
- 10) It will display the occurrences of character (o).
- 11) Again testfile descriptor. If it returns null then display the contents from the buffer.
- 12) Close the file.
- 10) Exit from the program

1: Algorithm for Number of Blank spaces in the file

- i. Start
- ii. Initialize RSI to start of text file and RDI to end of text file,
- iii. Start checking for the blank space if space is detected count will increase.
- iv. Display Number of blank spaces
- v. Exit

2: Algorithm for Number of lines in the file

- i. Start
- ii. Initialize RSI to start of text file and RDI to end of text file,
- iii. Start checking the line if enter is detected count of line will increase
- iv. Display Number of lines
- v. Ret

3: Algorithm to count occurrences of character

- i. Start
- ii. Initialize RSI to start of text file and RDI to end of text file,
- iii. Start checking the occurrences of character (o) and count it,
- iv. Display the count value if character (o) is present.
- v. If character (o) is not present Display the message that character (o) is not present.
- vi. Stop

Conclusion:- Hence we studied and implement ALP using NEAR and FAR Procedure.

Questions:-

1. What is Procedure ? Compare NEAR & FAR procedure?
2. Explain PUBLIC and EXTERN directives?