EXPERIMENT NO: 12

1. Title:

Implement Pass-II of two pass assembler for pseudo-machine in Java using object oriented features. The output of assignment-1 (intermediate file and symbol table) should be input for this assignment.

2. Objectives:

- To understand data structures to be used in pass II of an assembler.
- To implement pass I of an assembler

3. Problem Statement:

Write a program to create pass-II Assembler

4. Outcomes:

After completion of this assignment students will be able to:

- Understand the concept of Pass-II Assembler
- Understand the Programming language of Java

5. Software Requirements:

- Linux OS, JDK1.7

6. Hardware Requirement:

- 4GB RAM ,500GB HDD

7. Theory Concepts:

Design of a Two Pass Assembler: -

Tasks performed by the passes of two-pass assembler are as follows:

Pass I: -

Separate the symbol, mnemonic opcode and operand fields.

Determine the storage-required foe every assembly language statement and update the location counter.

Build the symbol table and the literal table.

Construct the intermediate code for every assembly language statement.

Pass II: -

Synthesize the target code by processing the intermediate code generated during pass1

Data Structure used by Pass II:

1. OPTAB: A table of mnemonic opcodes and related information.

2. SYMTAB: The symbol table

3. POOL_TAB and LITTAB: A table of literals used in the program

4. Intermediate code generated by Pass I

5. Output file containing Target code / error listing.

8. Algorithms(procedure) :
Algorithm :

1.code_area_address=address of code area;

Pooltab_ptr:=1;

loc_cntr=0;

2.While next statement is not an END statement

a) clear the machine_code_buffer

b) if an LTORG statement

I) process literals in LITTAB[POOLTAB[pooltab_ptr]]…

LITTAB[POOLTAB[pooltab_ptr+1]]-1 similar to processing of constants in a dc statement.

II) size=size of memory area required for literals

III) pooltab_ptr=pooltab_ptr+1

c) if a START or ORIGIN statement then

I) loc_cntr = value specified in operand field

II) size=0;

d) if a declaration statement

I) if a DC statement then assemble the constant in machine_code_buffer

II) size=size of memory area required by DC or DS:

e) if an imperative statement then

I) get operand address from SYMTAB or LITTAB

II) Assemble instruction in machine code buffer.

III) size=size of instruction;

f) if size # 0 then

I) move contents of machine_code_buffer to the address code_area_address+loc_cntr ;

II) loc_cntr=loc_cntr+size;

3. (Processing of END statement)

9. Conclusion:

Thus, I have studied visual programming and implemented dynamic link library application for arithmetic operation

References:

J. J. Donovan, "Systems Programming", McGraw Hill.[chapter 3]

Oral Questions: [Write short answer]

1. Explain various types of errors that are handled in passé-II.
2. Write algorithm of passé-II.
3. Draw flowchart of passé-II.
4. State various tables used and their significance in the design of two pass Assembler.
5. How LTORG statement is handled in pass II of assembler?
6. How Declarative statement is handled in pass II of assembler?
7. What is the significance of pool table?
8. Which data structures of pass I are used in pass II of assembler?
9. Explain the handling of imperative statement.
10. What feature of assembly language makes it mandatory to design a two pass assembler?
11. How are literals handled in an assembler?
12. How assembler directives are handled in pass I of assembler?