Assignment A-2

Title: Implementation of pass II of a two pass assembler

Problem statement: Implement Pass-2 of two pass assembler for Psuedo-machine in Java using object oriented features. The output of assignment-1 (intermediate file and Symbol table) should be the input for this assignment.

objective:
- Understand the internals of language translators
- Handle tools like LEX and YACC
- Understand the operating system internals and functionalities with implementation point of view.

S/w Packages and Hardware Packages Apparatus used: 64-bit Open Source Linux, Eclipse IDE, JAVA '13 and i5 machines.

Theory:

Assembler is a program which converts assembly language instructions into machine language form. A two pass assembler takes two Scans of Source code to produce the machine code from Assembly language program.

Assembly process consists of following Activities:

- Convert mneomonics to their machine language opcode equivalents
- Convert symbolic operands to their machine address
- Translate data constants into internal machine representation
- Output the object program and provide other information required for linker and loader.

Pass II Tasks:

- Assemble instructions (generate opcode and look up addresses)
- Generate data values defined by BYTE, WORD
- Perform processing of Assembler directives (not done in pass I)
- Write the object program and the assembly listing

Description using set theory:

Let 's' be set which represents a System

$S = \{I, O, T, D, Succ, Fail\}$

Where I = Input
O = output
T = Type
D = Data structure.

$I = \{Ic, st, Lt\}$

Where, Ic = Intermediate Code file
st = Symbol Table
Lt = Literal Table

$st = \{N, A\}$

Where,
N = Name of Symbol
A = Address of Symbol

$Lt = \{N, A\}$

where,
N = Name of Literal
A = Addres of Literal

Conclusion :

Hence, we successfully learn to Parse and tokenize the intermediate code file, perfo the LC procesing, generate the target code file, Demonstrate the use of symbol table, literal table, pooltable.