**TITLE: Implement pass-II of a two-pass assembler**

**PROBLEM STATEMENT:** Implement pass-II of a two-pass assembler. The output of assignment-1 (intermediate file and symbol table) should be input for this assignment.

**THEORY:**

* Assembler: An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations. Some people call these instructions assembler language and others use the term assembly language.
* Pass-II of two pass assembler: Pass-2 of assembler generates machine code by converting symbolic machine-opcodes into their respective bit configuration(machine understandable form). It stores all machine-opcodes in MOT table (op-code table) with symbolic code, their length and their bit configuration.
* Symbol table: Symbol Table is an important data structure created and maintained by the compiler in order to keep track of semantics of variables i.e. it stores information about the scope and binding information about names, information about instances of various entities such as variable and function names, classes, objects, etc.

**PROGRAM:** import

java.io.BufferedReader; import

java.io.FileReader; import java.io.FileWriter; import java.io.IOException; import java.util.HashMap;

public class pass2 {

public static void main(String[] Args) throws IOException{

BufferedReader b1 = new BufferedReader(new FileReader("intermediate.txt"));

BufferedReader b2 = new BufferedReader(new FileReader("symtab.txt"));

BufferedReader b3 = new BufferedReader(new FileReader("littab.txt"));

FileWriter f1 = new FileWriter("Pass2.txt");

HashMap<Integer, String> symSymbol = new HashMap<Integer, String>();

HashMap<Integer, String> litSymbol = new HashMap<Integer, String>(); HashMap<Integer, String> litAddr = new HashMap<Integer,

String>(); String s; int symtabPointer=1,littabPointer=1,offset; while((s=b2.readLine())!=null){ String word[]=s.split("\t\t\t"); symSymbol.put(symtabPointer++,word[1]);

} while((s=b3.readLine())!=null){ String word[]=s.split("\t\t"); litSymbol.put(littabPointer,word[0]);

litAddr.put(littabPointer++,word[1]);

} while((s=b1.readLine())!=null){

if(s.substring(1,6).compareToIgnoreCase("IS,00")==0){ f1.write("+ 00 0 000\n");

} else if(s.substring(1,3).compareToIgnoreCase("IS")==0){

f1.write("+ "+s.substring(4,6)+" "); if(s.charAt(9)==')'){

f1.write(s.charAt(8)+"

"); offset=3; } else{ f1.write("0 ");

offset=0; }

if(s.charAt(8+offset)=='S')

f1.write(symSymbol.get(Integer.parseInt(s.substring(10+offset,s.length()-1)))+"\n"); else

f1.write(litAddr.get(Integer.parseInt(s.substring(10+offset,s.length()-1)))+"\n");

}

else

if(s.substring(1,6).compareToIgnoreCase("DL,01")==0){ String s1=s.substring(10,s.length()-1),s2=""; for(int i=0;i<3-s1.length();i++)

s2+="0"; s2+=s1;

f1.write("+ 00 0 "+s2+"\n");

} else{ f1.write("\n");

} } f1.close(); b1.close(); b2.close();

b3.close();

}

}

INPUT FILES:

Intermediate.txt:

(AD,01)(C,200)

(IS,04)(1)(L,1)

(IS,05)(1)(S,1)

(IS,04)(1)(S,1)

(IS,04)(3)(S,3)

(IS,01)(3)(L,2)

(IS,07)(6)(S,4)

(DL,01)(C,5)

(DL,01)(C,1)

(IS,02)(1)(L,3)

(IS,07)(1)(S,5)

(IS,00)

(AD,03)(S,2)+2

(IS,03)(3)(S,3)

(AD,03)(S,6)+1

(DL,02)(C,1)

(DL,02)(C,1)

(AD,02)

(DL,01)(C,1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| symtab.txt: |  |  |  |  |
| A | 211 |  |  | 1 |
| LOOP | 202 |  |  | 1 |
| B | 212 |  |  | 1 |
| NEXT | 208 |  |  | 1 |
| BACK | 202 |  |  | 1 |
| LAST | 210 |  |  | 1 |

littab.txt:

5 206

1 207

1 213

OUTPUT:

+ 04 1 206

+ 05 1 211

+ 04 1 211

+ 04 3 212

+ 01 3 207

+ 07 6 208

+ 00 0 005

+ 00 0 001

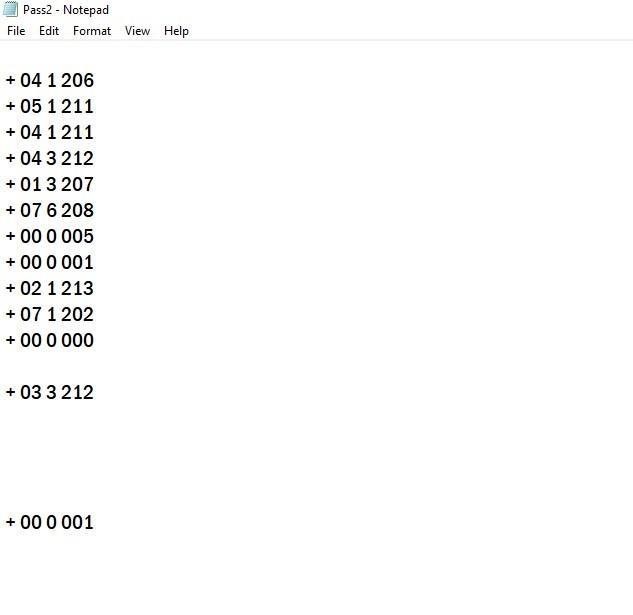
+ 02 1 213

+ 07 1 202

+ 00 0 000

+ 03 3 212

+ 00 0 001



**CONCLUSION:**

We have successfully learnt the implementation and the theory of pass-II of a two pass assembler.