Name - Vasu Kalariya
Roll - PE29
Sub - SSC

Lab Assignment - 4

Assignment Title: Design of Pass II of Two Pass Macroprocessor.

Aim: Design of pass II of Two Pass Macroprocessor.

Objective: Design suitable data structure & implement pass 1 of Two Pass. Macroprocessor.

Algorithm for Pass II:

1] Read Instruction from source program output by pass 1 divide its field as label ~~ne~~ mnemonic opcode arguments.

2] Search through MNT to find match for opcode of instruction step 1 with macro name MNT.

3] /* if no macro call found */ if no match ~~f~~ then it indicates that this instruction is not a macro call instruction and then hence:

→ write this instruction to expanded source program

→ check whether opcode of this instruction is END or NOT.

→ If NOT END then it indicates that this is not end of source program & hence go to step 1

→ If Opcode = then this indicates end of source program and hence give the output pass II i.e. expanded source program to assembler

4] /* if macro name found */ if OPCODE of (instruction read in step 1) = any macro name of MNT then it indicate & hence: this instruction is macro call instruction & hence:

→ Obtain corresponding MDT indent and assign to MDTP

→ ~~St~~ Setup ALA (for association of integer indicates and actual parameters.

→ MDTP MDTP + 1

- → Get next instruction from MDT.
- → Substitute actual argument ~~us~~ instead of integer indicate
- → If OPCODE of this instruction is not MEND then write this instruction to expand source program
- → If OPCODE of this instruction is MEND then go to step 1 again.

Input:

1] Program without macro definition.

```
START
MOVER   AREG S1
MOVER   BREG S1
MCALL   D5   D8
INCR    D2  D1
S1      DC   5
D1      DC   2
D2      DC   3
D5      DC   5
D8      DC   1
END
```

2] Macro defination table

```
0    INCR  ARG1    ARG2
1    ADD   AREG    ARG1
2    ADD   BREG    ARG2
3    MEND
4    MCALL   K1     K2
5    SUB     BREG   K1
6    SUB     DREG   K2
7    MEND
```

3] Macro Name Table

0	INCR	0
1	MCALL	4

4] Argument List Array.

0	&ARG1
1	&ARG2
2	&K1
3	&K2

<u>Output</u>: Expanded Code

```
START
MOVER    AREG S1
MOVER    BREG S1
SUB      CREG  D5
SUB      DREG D8
ADD      AREG  D2
ADD      BREG  D1
S1       DC      5
D1       DC      2
D2       DC     3
D5       DC    4
D8       DC    1
END
```

5] Modified ALA

0	&D2
1	&D1
2	&D5
3	&D8

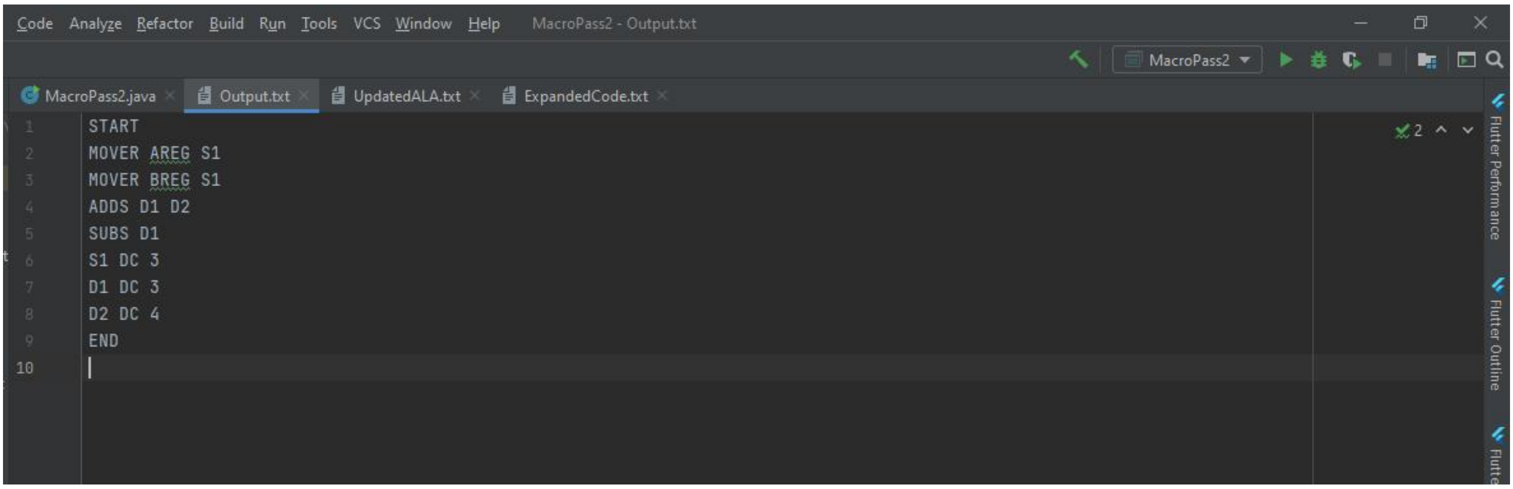
Conclusion: The function of pass II of a 2 pass macroprocessor is ~~stuffed~~ studied along with advanced macro ~~facility.~~ facility.

Platform: Linux / Windows

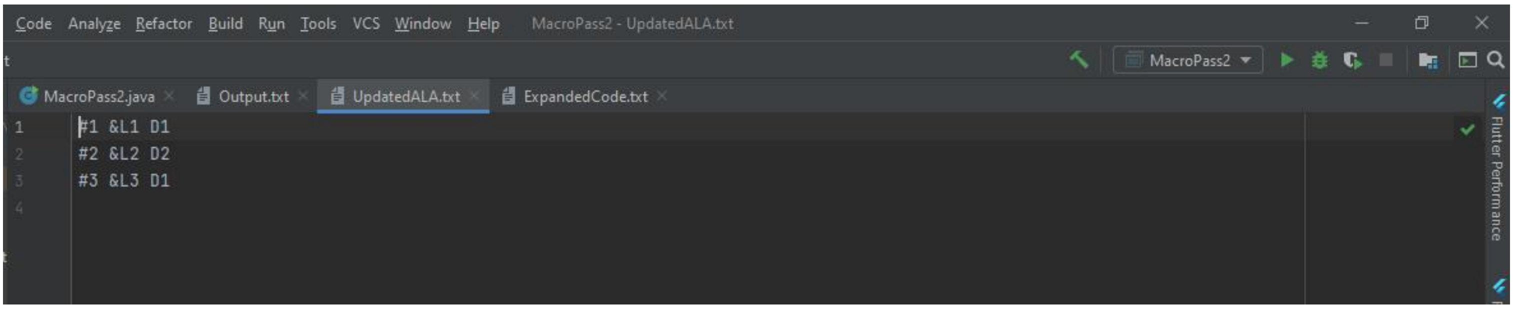
```java
import java.io.*;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Hashtable;

public class MacroPass2 {
    public static void main(String[] args) throws IOException {

        Hashtable<String, String> ala = new Hashtable<String, String>();
        Hashtable<String, String> mnt = new Hashtable<String, String>();

        String inputfilename = "F:\\T9\\SSC\\Assi 4\\src\\Output.txt";       //Input File
        FileReader fr1 = new FileReader(inputfilename);
        BufferedReader br1 = new BufferedReader(fr1);

        String MNT = "F:\\T9\\SSC\\Assi 4\\src\\MNT.txt";       //Input File MNT
        FileReader fr2 = new FileReader(MNT);
        BufferedReader br2 = new BufferedReader(fr2);

        String MDT = "F:\\T9\\SSC\\Assi 4\\src\\MDT.txt";       //Input File MDT
        FileReader fr3 = new FileReader(MDT);
        BufferedReader br3 = new BufferedReader(fr3);

        String ALA = "F:\\T9\\SSC\\Assi 4\\src\\ALA.txt";       //Input File ALA
        FileReader fr4 = new FileReader(ALA);
        BufferedReader br4 = new BufferedReader(fr4);

        String f1 = "F:\\T9\\SSC\\Assi 4\\src\\ExpandedCode.txt";
        FileWriter fw1 = new FileWriter(f1);
        BufferedWriter bw1 = new BufferedWriter(fw1);

        String f2 = "F:\\T9\\SSC\\Assi 4\\src\\UpdatedALA.txt";
        FileWriter fw2 = new FileWriter(f2);
        BufferedWriter bw2 = new BufferedWriter(fw2);

        LineNumberReader lineNumberReader =
                new LineNumberReader(new FileReader("F:\\T9\\SSC\\Assi 4\\src\\MDT.txt"));

        String sCurrentLine,s0,s1,s2;
        boolean macro=false;
        int line_number=0;


        while ((sCurrentLine = br4.readLine()) != null){
            s0 = sCurrentLine.split(" |\\,")[1];
            s1 = sCurrentLine.split(" |\\,")[0];
            ala.put(s0,s1);
        }

        while ((sCurrentLine = br2.readLine()) != null){
            s0 = sCurrentLine.split(" |\\,")[1];
            s1 = sCurrentLine.split(" |\\,")[2];
            mnt.put(s0,s1);
        }

        while ((sCurrentLine = br1.readLine()) != null){
            s0 = sCurrentLine.split(" |\\,")[0];
            s1 = Files.readAllLines(Paths.get("F:\\T9\\SSC\\Assi 4\\src\\MDT.txt")).get(line_number);
            s1 = s1.split(" |\\,")[1];

            if (mnt.containsKey(s0)){
                macro = true;
                line_number = (Integer.parseInt(mnt.get(s0)) - 1);

                for (int i = 0; i < (sCurrentLine.split(" |\\,").length); i++){

                    String mdt_words = Files.readAllLines(Paths.get("F:\\T9\\SSC\\Assi 4\\src\\MDT.txt")).get(line_number);
                    mdt_words = mdt_words.split(" |\\,")[i+1];
```

```java

          String output_word = sCurrentLine.split(" |\\,")[i];

          if(mnt.containsKey(mdt_words) != true){
            String parameter = ala.get(mdt_words);
            ala.remove(mdt_words);
            ala.put(parameter, output_word);
          }

        }
        line_number++;

        while (macro == true){
          String mdt_line = Files.readAllLines(Paths.get("F:\\T9\\SSC\\Assi 4\\src\\MDT.txt")).get(line_number);
          s1 = mdt_line.split(" |\\,")[1];
          if(s1.equals("MEND")){
            macro = false;
            line_number = 0;
          }
          else{
            for (int i = 1; i < (mdt_line.split(" |\\,").length); i++){
              String temp = mdt_line.split(" |\\,")[i];
              if(ala.containsKey(temp)){
                bw1.write(ala.get(temp)+ " ");
              }
              else {
                bw1.write(temp+" ");
              }
            }
            bw1.write("\n");
            line_number++;
          }
        }
      }
```

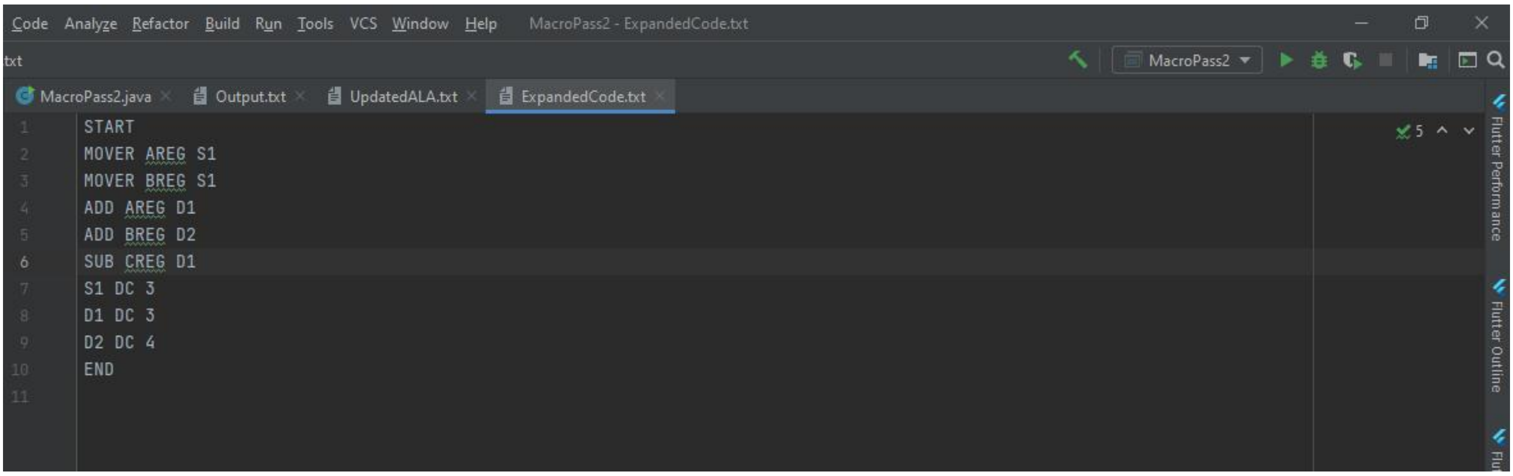
```

            else {
                bw1.write(sCurrentLine+ "\n");
            }

        }

        String ALA2 = "F:\\T9\\SSC\\Assi 4\\src\\ALA.txt";
        FileReader fr5 = new FileReader(ALA2);
        BufferedReader br5 = new BufferedReader(fr5);

        while ((sCurrentLine = br5.readLine()) != null){
            s0 = sCurrentLine.split(" |\\,")[0];
            s1 = sCurrentLine.split(" |\\,")[1];

            fw2.write(s0 + " " + s1 + " " + ala.get(s0) + "\n");
        }

        bw1.close();
        bw2.close();
    }
}

```



```
START
MOVER AREG S1
MOVER BREG S1
ADDS D1 D2
SUBS D1
S1 DC 3
D1 DC 3
D2 DC 4
END
```



```
#1 &L1 D1
#2 &L2 D2
#3 &L3 D1
```



```
START
MOVER AREG S1
MOVER BREG S1
ADD AREG D1
ADD BREG D2
SUB CREG D1
S1 DC 3
D1 DC 3
D2 DC 4
END
```