Preetam Mogal TE (B)-17

Assignment No. 4.
Pass-2 Macroprocessor.

Aim :- Design of a MACRO PASS-2

Problem Statement :- Write a Java program for pass-II of a two-pass macro-processor. The output of assignment-3 (MNT, MDT and file without any macro defination) should be input for this assignment.

Theory :-

1: Macro processor (Defination)
A macro processor is a program that reads a file (or files) and scans them for certain keywords.

2: Basic tasks performed by macro processor:
a) Recognize macrodefination.
b) Save thedefination.
c) Recognize call.
d) Expanded calls and substitute arguments.

- Pass I Macro Defination
- Pass 2 Macro calls and Expansion.

Pass 1 Macro Defination:
Pass 1 algorithm examines each line of the input data for macro pseudo opcode. Following are steps that are performed during Pass 1 algorithm:
1. Initialize MDTC and MNTC with value one, so that previous value of MDTC & MNTC is set of

2. Read the first input data.
3. If the data contains MACRO pseudo opcode then
   A. Read the next data input.
   B. Enter the name of the macro and current value of MDTC in MNT.
   C. Increase the counter value of MNT by value one
   D. Prepare the argument list array respective to the macro found.
   E. Enter the macro defination into MDT. Increase the counter of MDT By value one.
   F. Read next line of the input data.
   G. Substitute the index notations for dummy arguments passed in MACRO.
   H. Increase the counter of the MDT by value one.
   I. If mend pseudo opcode is encountered then next source of input data Is read.
   J. Else expand data input.

4. If macro pseudo opcode is not encountered in data input then
   A. A copy of input data is created.
   B. If end pseudo code is found then go to Pass 2
   C. Otherwise read next source of input data.

Pass 2 Macro Calls and Expansion
~~Pass~~
1. Read the input data received From Pass-I
2. Examine each operation code for finding respective entry in the MNT.
3. If name of the macro is encountered then
   A. A pointer is set to the MNT entry where

name of macro is found.
This pointer is called Macro Defination Table Pointer (MDTP).

B. Prepare argument list array containing a table of dummy argument.

C. Increase the value of MDTP by value one.

D. Read next line from MDT.

E. Substitute the values from the arguments list of the macro for Dummy arguments.

F. If mend pseudo opcode is found then next source of input data is Read.

G. Else expands datainput.

4. When macro name is not found then create expanded data file

5. If end pseudo opcode is encountered then feed the expanded source file to Assembler for processing.

6. Else read next source of datainput.

Draw Flowchart w.r.t algorithm :

Input:

```
INPUT
MACRO
FNCR1 & FIRST, &SECOND = DATA9
 A      1, & FIRST
 L      2, & SECOND
MEND  MACRO
FNCR2 & ARG1 & ARG1, & ARG2 = DATA 5
 L      3, & ARG1
 ST     4, & ARG2
MEND
PRG2  START
USING     *, BASE
```

```
INCR1 DATA1
INCR2 DATA3, DATA4
FOUR DC F'4'
FIVE DC· F'5'
 BASE EQU 8
TEMP DS 1F
DROP 8
END.
```

Output

```
=== = = = -PASS 1 = = = = =
ALA:
[& FIRST, & SECOND]
[& ARG1, & ARG2]

MNT:
[INCR1 , 0]
[INCR 2 , 4]

MDT:        & FIRST, & SECOND = DAT
 INCR       A9
 1
 A          1, #0
 L          2, #1
MEN
 D          & ARG1 , & ARG2 = DATA5
INCR
 2
 L          3, #0
ST          4, #1
MEN
D
```

```
======= PASS 2 ======
MDT :     & FIRST , & SECOND= DAT
 INCR     19
  1
  A       1,# 0
  L       2,#1
 MEN
  D

PRG 2          STAR
               T              *, BASE
               USIN
                G
               A              1, DATA 1
               L              2, DATA9
               L              3, DATA3
               ST             4, DATA4
FOUR           DC             F'4'
FIVE           DC             F'5'
BASE           EQU            8
TEMP           DS             1F
               DRO            8
               P
               END

ALA:
[DATA 1, DATA9]
[DATA 3, DATA 4]
```

Conclusion:-
Thus Pass II of Macro processor is implemented and ALA file is generated.

Name: Pritam Gotiram Mogal **DivRoll**: TE-B-17

Assignment No. 04 [PASS-2 Macroprocessor]

Problem Satement: Write a Java program for pass-II of a two-pass macro-processor. The output of assignment-3 (MNT, MDT and file without any macro definitions) should be input for this assignment.

1. Pass 2 Macro Code:

```
import java.io.*;
import java.util.HashMap;
import java.util.Vector;

public class macroPass2 {
        public static void main(String[] Args) throws IOException{
                BufferedReader b1 = new BufferedReader(new FileReader("intermediate.txt"));
                BufferedReader b2 = new BufferedReader(new FileReader("mnt.txt"));
                BufferedReader b3 = new BufferedReader(new FileReader("mdt.txt"));
                BufferedReader b4 = new BufferedReader(new FileReader("kpdt.txt"));
                FileWriter f1 = new FileWriter("Pass2.txt");
                HashMap<Integer,String> aptab=new HashMap<Integer,String>();
                HashMap<String,Integer> aptabInverse=new HashMap<String,Integer>();
                HashMap<String,Integer> mdtpHash=new HashMap<String,Integer>();
                HashMap<String,Integer> kpdtpHash=new HashMap<String,Integer>();
                HashMap<String,Integer> kpHash=new HashMap<String,Integer>();
                HashMap<String,Integer> macroNameHash=new HashMap<String,Integer>();
                Vector<String>mdt=new Vector<String>();
                Vector<String>kpdt=new Vector<String>();
                String s,s1;
                int i,pp,kp,kpdtp,mdtp,paramNo;
                while((s=b3.readLine())!=null)
                        mdt.addElement(s);
                while((s=b4.readLine())!=null)
                        kpdt.addElement(s);
                while((s=b2.readLine())!=null){
                        String word[]=s.split("\t");
                        s1=word[0]+word[1];
                        macroNameHash.put(word[0],1);
                        kpHash.put(s1,Integer.parseInt(word[2]));
                        mdtpHash.put(s1,Integer.parseInt(word[3]));
                        kpdtpHash.put(s1,Integer.parseInt(word[4]));
                }
                while((s=b1.readLine())!=null){
                        String b1Split[]=s.split("\\s");
                        if(macroNameHash.containsKey(b1Split[0])){
                                pp= b1Split[1].split(",").length-b1Split[1].split("=").length+1;
                                kp=kpHash.get(b1Split[0]+Integer.toString(pp));
                                mdtp=mdtpHash.get(b1Split[0]+Integer.toString(pp));
                                kpdtp=kpdtpHash.get(b1Split[0]+Integer.toString(pp));
                                String actualParams[]=b1Split[1].split(",");
                                paramNo=1;
                                for(int j=0;j<pp;j++){
                                        aptab.put(paramNo, actualParams[paramNo-1]);
                                        aptabInverse.put(actualParams[paramNo-1],paramNo);
                                        paramNo++;
                                }
                                i=kpdtp-1;
                                for(int j=0;j<kp;j++){
```

```
						String temp[]=kpdt.get(i).split("\t");
						aptab.put(paramNo,temp[1]);
						aptabInverse.put(temp[0],paramNo);
						i++;
						paramNo++;
					}
					i=pp+1;
					while(i<=actualParams.length){
						String initializedParams[]=actualParams[i-1].split("=");

	aptab.put(aptabInverse.get(initializedParams[0].substring(1,initializedParams[0].length())),initializedParams[1
].substring(0,initializedParams[1].length()));
						i++;
					}
					i=mdtp-1;
					while(mdt.get(i).compareToIgnoreCase("MEND")!=0){
						f1.write("+ ");
						for(int j=0;j<mdt.get(i).length();j++){
							if(mdt.get(i).charAt(j)=='#')
								f1.write(aptab.get(Integer.parseInt("" +
mdt.get(i).charAt(++j))));
							else
								f1.write(mdt.get(i).charAt(j));
						}
						f1.write("\n");
						i++;
					}
					aptab.clear();
					aptabInverse.clear();
				}
				else
					f1.write("+ "+s+"\n");
			}
			b1.close();
			b2.close();
			b3.close();
			b4.close();
			f1.close();
	}
}
```

/*
OUTPUT:

```
OUTPUT:
Pritam-spos@Pritam-HP:~/SPOSL$ javac macroPass2.java
Pritam-spos@Pritam-HP:~/SPOSL$ java macroPass2
Pritam-spos@Pritam-HP:~/SPOSL$ cat Pass2.txt

Intermediate - -
M1 10,20,&b=CREG
M2 100,200,&u=&AREG,&v=&BREG



Kpdt--
a	AREG
b	-
u	CREG
v	DREG
```

```
pass2 --
+ MOVE AREG,10
+ ADD AREG,='1'
+ MOVER AREG,20
+ ADD AREG,='5'
+ MOVER &AREG,100
+ MOVER &BREG,200
+ ADD &AREG,='15'
+ ADD &BREG,='10'


MNT --
M1      2       2       1       1
M2      2       2       6       3


MDT --
MOVE #3,#1
ADD #3,='1'
MOVER #3,#2
ADD #3,='5'
MEND
MOVER #3,#1
MOVER #4,#2
ADD #3,='15'
ADD #4,='10'
MEND
```