Name: Aishwarya Shukla

Roll no.: 324002

Gr no. 22010492

Batch: D1

**LPCC Assignment no. 02**

Problem Statement: Design suitable data structures &amp; implement pass-I of a two-pass Macro processor.

Objective:

1. Students will be able to design data structure for pass-1 of two pass assembler

Theory:

1. Assembler is a program for converting instructions written in low-level assembly code into relocatable machine code and generating along information for the loader.
2. It generates instructions by evaluating the mnemonics (symbols) in operation field and find the value of symbol and literals to produce machine code. Now, if assembler do all this work in one scan then it is called single pass assembler, otherwise if it does in multiple scans then called multiple pass assembler. Here assembler divide these tasks in two passes:
3. Pass-1: Define symbols and literals and remember them in symbol table and literal table respectively. Keep track of location counter Process pseudo-operations
4. Pass-2: Generate object code by converting symbolic op-code into respective numeric op code Generate data for literals and look for values of symbols

Program:

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Iterator;

import java.util.LinkedHashMap;

public class macrop1 {

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

        BufferedReader br = new BufferedReader(new FileReader("input.asm"));

        FileWriter mnt = new FileWriter("mnt.txt");

        FileWriter mdt = new FileWriter("mdt.txt");

        FileWriter fvsppl = new FileWriter("fvsppl.txt");

        FileWriter avsppl = new FileWriter("avsppl.txt");

        FileWriter pnt = new FileWriter("pntab.txt");

        FileWriter ir = new FileWriter("intermediate.txt");

        LinkedHashMap<String, Integer> pntab = new LinkedHashMap<>();

        String line;

        String Macroname = null;

        int mdtp = 1, kpdtp = 0, paramNo = 1, pp = 0, kp = 0, flag = 0;

        while ((line = br.readLine()) != null) {

            String parts[] = line.split("\\s+");

            if (parts[0].equalsIgnoreCase("MACRO")) {

                flag = 1;

                line = br.readLine();

                parts = line.split("\\s+");

                Macroname = parts[0];

                if (parts.length <= 1) {

                    mnt.write(parts[0] + "\t" + pp + "\t" + "\t" + mdtp + "\n");

                    continue;

                }

                for (int i = 1; i < parts.length; i++) // processing of parameters

                {

                    parts[i] = parts[i].replaceAll("[&,]", "");

                    fvsppl.write(parts[0] + "\n" + "------------------" + "\n" + "\n");

                    avsppl.write(parts[0] + "\n" + "------------------" + "\n" + "\n");

                    if (parts[i].contains("=")) {

                        ++kp;

                        String keywordParam[] = parts[i].split("=");

                        pntab.put(keywordParam[0], paramNo++);

                    } else {

                        pntab.put(parts[i], paramNo++);

                        pp++;

                        fvsppl.write(parts[i] + "\t\t" + "#" + i + "\n" + "\n" + "\n");

                        avsppl.write("\t\t" + "#" + i + "\n" + "\n" + "\n");

                    }

                }

                mnt.write(parts[0] + "\t" + pp + "\t" + "\t" + mdtp + "\t" + "\n");

                // fvsppl.write(parts[0]+"\n"+"\n"+"\n"+"\n"+"\n");

                kpdtp = kpdtp + kp;

                // System.out.println("KP="+kp);

            } else if (parts[0].equalsIgnoreCase("MEND")) {

                mdt.write(line + "\n");

                flag = kp = pp = 0;

                mdtp++;

                paramNo = 1;

                pnt.write(Macroname + ":\t");

                Iterator<String> itr = pntab.keySet().iterator();

                while (itr.hasNext()) {

                    pnt.write(itr.next() + "\t");

                }

                pnt.write("\n");

                pntab.clear();

            } else if (flag == 1) {

                for (int i = 0; i < parts.length; i++) {

                    if (parts[i].contains("&")) {

                        parts[i] = parts[i].replaceAll("[&,]", "");

                        mdt.write("(P," + pntab.get(parts[i]) + ")\t");

                        fvsppl.write(parts[i] + "\t\t" + "#" + i + "\n" + "\n" + "\n");

                    } else {

                        mdt.write(parts[i] + "\t");

                    }

                }

                mdt.write("\n");

                mdtp++;

            } else {

                ir.write(line + "\n");

            }

        }

        br.close();

        mdt.close();

        mnt.close();

        ir.close();

        pnt.close();

        fvsppl.close();

        avsppl.close();

        System.out.println("MAcro PAss1 Processing done."

                + "Check all outputs, "

                + "--------thnak you------ :)");

    }

}

Output:

