SRES's Sanjivani College of Engineering, Kopargaon
(*An Autonomous Institute*)
Department of Computer Engineering

SPOS Lab Manual

Assignment No. 04

Title: Implementation of Pass 2 of Two Pass Macroprocessor

Aim :
Write a Java/Python 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.

Inputs :
1. Assembly language program without macro definitions but with macro calls
2. Macro Definition Table
3. Macro Name Table

Outputs :
1. Assembly Language Program in which macro calls are get replaced by their definitions
2. Argument List Array

Theory :
Pass 2 of macroprocessor perform two important tasks
Recognize macro calls – The macroprocessor must recognize macro calls that appear as operation mnemonics.
Expand calls and substitute arguments – The macroprocessor must substitute for dummy arguments the corresponding arguments from a macro call. The resulting assembly language text is then substituted for the macro call.

Pass 2 data structures
1. **Input Assembly Language Program without macro definitions and with macro calls**
   Input Assembly Language Program contains few assembly instructions and one or macro macro calls

2. **Macro Name Table (MNT), created by pass1**
   used to match the mnemonic opcode of each input line with the macro names stored in MNT, for recognizing the macro calls from input file

3. **Macro Definition Table (MDT), created by pass1**
   used to fetch the macro definitions for expansion of calls

4. **Macro Definition Table Pointer (MDTP)**
   used to indicate the next line of text to be used during macro expansion

5. **Argument List Array (ALA)**
   used to substitute macro call arguments for the index markers in the stored macro definition

6. **Output Assembly Language Program with expansion of macro calls**
   In the output assembly language program macro calls are replaced by their definitions

Sample Example Showing Inputs and Outputs of Pass2 Macroprocessor

Input Assembly Language Program without macro definiton but with macro calls

```
          START
          MOVER    AREG, A
          MOVEM    BREG, B
          INCR     DATA1, DATA2
          DECR     DATA3, DATA4
DATA1     DC       5
DATA2     DC       10
DATA3     DC       15
DATA4     DC       20
```

Macro Name Table (MNT)

MNT Index	Macro Name	MDT Index
1	INCR	1
2	DECR	5

Argument List Array (ALA)

Index	Argument Name
1	DATA1
2	DATA2
3	DATA3
4	DATA4

Macro Definition Table (MDT)

MDT Index	Macro Instructions
1	INCR &ARG1, &ARG2
2	ADD AREG, #1
3	ADD AREG, #2
4	MEND
5	DECR &ARG3, &ARG4
6	SUB BREG, #3
7	SUB BREG, #4
8	MEND

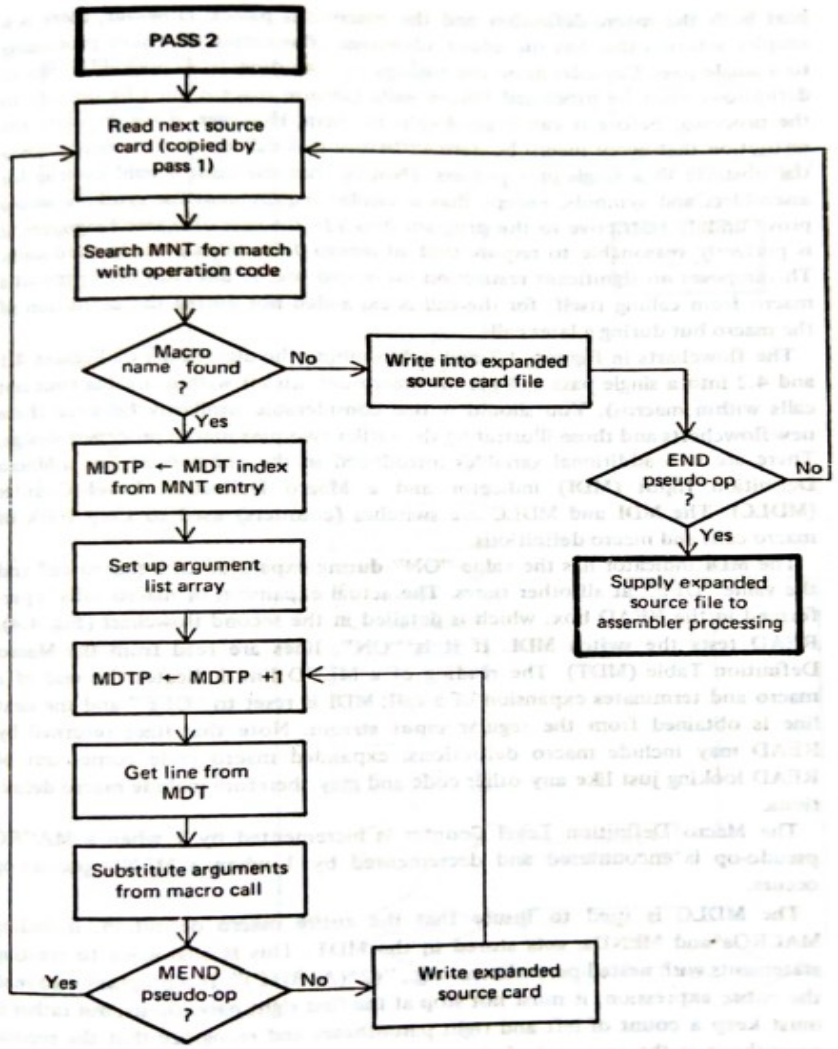
Output Assembly Language Program with expansion of macro calls

```
        START
        MOVER   AREG, A
        MOVEM   BREG, B
        ADD     AREG, DATA1
        ADD     AREG, DATA2
        SUB     BREG, DATA3
        SUB     BREG, DATA4
DATA1   DC      5
DATA2   DC      10
DATA3   DC      15
DATA4   DC      20
        END
```

Algorithm for Pass 2 of a two-pass Macroprocessor

1. Test mnemonic opcode of each instruction from input assebly language program
2. If mnemonic opcode matches with any of the Macro name stored in MNT then it is a macro call else write the input instruction as it is into the output file of pass 2
3. When call is found, the call processor sets the Macro Definition Table Pointer (MDTP), to the corresponding macro definition stored in MDT.
4. The initial value of MDTP is obtained from the “MDT index” field of the MNT entry
5. The macro expander prepares the Argument List Array (ALA) consisting of a table of dummy argument indices and corresponding arguments to the call
6. Each successive macro instruction line is read from MDT and actual arguments from ALA are substituted for the dummy argument indices in the macro definition and write the instruction in the output file of pass 2
7. Reading of the MEND line in the MDT terminates expansion of the macro, and scanning continues from the input file
8. When the END pseudo opcode is encountered, the expanded source program is transferred to the assembler for further processing

Flowchart of Pass2 of Macroprocessor



Conclusion: In this assignment we have implemented pass II of Macroprocessor. Macro calls are recognized from input assembly language program and are replaced by their definitions

References: Systems Programming by John J. Donovan

Prepared by
Prof.N.G.Pardeshi
Subject Teacher

Approved by
Dr. D.B.Kshirsagar
HOD