

Practical no: 4

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 definitions) should be input for this assignment.

Name: Aditi Dinesh Mulay

Class: T.E. Computer

Subject: SPOS

Div: A

Roll no: 02

PRN No. 71918146B

SPOS

Assignment A-4

Aditi Dinesh Mulay
T.E. Comp. Div: A
Roll no: 02

Aim: Design of a MACRO PASS-II.

Problem Statement: Write a Java program for pass-II of a two-pass macroprocessor. The o/p of assignment-III should be i/p for this practical.

Theory:

Pass-I Macro Definition.

- 1] Initialize MDT and MNT with value one, so that previous value of MDT & MNT is set to value one.
- 2] Read the first input data.
- 3] If this data contains MACRO pseudo opcode then
 1. Read the next data i/p.
 2. Enter the name of macro & current value of MDT in MNT.
 3. Increase the counter value of MNT by value one.
 4. Prepare that argument list array respective to the macro found.
 5. Enter the macro definition into MDT. Increase the counter of MDT by value one.
 6. Read next line of the input data.
 7. Substitute the index notations for dummy arguments passed in Macro.
 8. Increase the counter of MDT by value one.
 9. If send pseudo opcode is encountered then next source of input data is read.
 10. Else expands data i/p.
- 4] If macro pseudo opcode is not encountered in data input then,

1. A copy of i/p data is created.
2. If end pseudo opcode is found then go to pass-II.
3. Otherwise read next source of i/p. data.

Pass-II Macro Calls & Expansion.

- 1] Read the i/p data received from Pass-I.
- 2] Examine each operation code for finding resp. entry in the MNT.
- 3] If the name of macro is encountered then,
 1. If pointer is set to MNT entry where name of macro is found. This pointer is called Macro Definition Table Pointer (MDTP).
 2. Prepare argument list array containing a table of dummy arguments.
 3. Increase the value of MDTP by value one.
 4. Read next line from MDT.
 5. Substitute the values from the arguments list of macro for dummy arguments.
 6. If end pseudo opcode is found then next source of i/p data is read.
 7. Else expands data input.
- 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 data input.

```

Input: 1000 00000000 10000000 10000000
INPUT
MACRO
INCR1 &FIRST, &SECOND = DATA9
A 1, &FIRST
L 2, &SECOND
MEND MACRO
INCR2 &ARG1, &ARG2 = DATA5
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 IF
DROP 8
END

OUTPUT
—Pass I—
ALA:
[&FIRST, &SECOND]
[&ARG1, &ARG2]

MNT:
[INCR1, 0]
[INCR2, 4]

```



```

MODT:    $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 -

MODT:    $FIRST, $SECOND = DAT
INCR     A9
1
A        1, #0
L        2, #1
MEN
D

PRG2     STAR
          T        *, BASE
          USING
          @A        1, DATA1
          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:
[DATA1, DATA9]
[DATA3, DATA4]

Conclusion:
Thus, pass II of Macro processor is implemented &
ALA file is generated.

```