



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

**Batch: B3**

**Roll No.: 2021009**

**Experiment / assignment / tutorial No. 10**

**Grade: AA / AB / BB / BC / CC / CD / DD**

**Signature of the Staff In-charge with date**

**TITLE : Design and implement 2 pass Macro Processor**

**AIM :** To learn working of two pass macro preprocessor and databases required for implementation of Macro Processor.

---

**Expected OUTCOME of Experiment:**

**CO 2:** design and implement the system programs like Assembler and Macro Pre-Processor etc.

---

**Books/ Journals/ Websites referred:**

1. Jhon J. Donovan : Systems programming, Tata McGraw Hill

---

**Pre Lab/ Prior Concepts:**

How macros are used in assembly language Program.

**Algorithm for PASS – I**

1. Set MDTC(macro definition table counter) to 1
2. Set MNTC(macro name table counter) to 1
3. Read next statement from source program
4. If this source statement is pseudo-opcode MACRO (start of macro definition) then goto step 5 else goto step 16
5. Read next statement from source program(macro name line)
6. Enter macro name found in step 5 in name field of MNT(macro name table) and Also enter current value of MDTC in MDT index field of MNT
7. Increment MNTC by 1
8. Prepare argument list array
9. Enter macro name into MDT at index MDTC
10. Increment MDTC by one



## **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

11. Read next statement from source program.
12. Create and substitute index notation for arguments in the source statement if any
13. Enter this line into the MDT
14. Increment MDTC by one
15. Check if currently read source statement is pseudo-opcode MEND. If yes then goto step 3 else goto step 11
16. Write source program statement as it is in file(output of pass 1 s.p.)
17. Check if pseudo opcode END(end of source program) is encountered . If yes goto step 18 else goto step 19
18. goto PASS – II
19. Go to step 3
20. end of PASS –I

### **Algorithm for PASS – II**

1. Read next statement from source program(from output s.p. file of pass I)
2. Search in MNT for match with operation code(mnemonic)
3. If macro name found the goto step 4 else goto step 11
4. Retrieve MDT index from MNT and store it in MDTP(macro definition table pointer).
5. Set up argument list array (store actual parameter in ala)
6. Increment MDTP by 1
7. Retrieve line pointer by MDTP from MDT
8. Substitute index notation by actual parameter from argument list array if any
9. Check if currently retrieved line is pseudo opcode MEND, if yes go to step 1 else goto step 10
10. Write the statement formed in step 8 to expanded source file and go to step 6
11. Write source statement directly into expanded source file
12. Check if pseudo opcode END(end of s.p. output of pass I) encountered, if yes goto step 13 else goto step 1
13. End of PASS-II

### **Stepwise-Procedure:**

- a. Read a Source Program as input.
- b. Recognise the Macro Definition and
- c. Create Macro Name Table (MNT), and Macro Definition Table (MDT), and Prepare ALA.
- d. Recognise the Macro Call statement
- e. Update the Macro Definition Table Pointer(MDTP), and Prepare ALA.
- f. Expand the macro call by reading instructions from the MDT.



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

**Sample input and output:**

***Input file:***

```
MACRO
INCR1 &FIRST,&SECOND=DATA9
A    1,&FIRST
L    2,&SECOND
MEND
MACRO
INCR2 &ARG1,&ARG2
L    3,&ARG1
ST   4,&ARG2
MEND
PRG2 START
      USING *,BASE
INCR1 DATA1,DATA12
INCR2 DATA3,DATA4
FOUR DC    F'4'
FIVE  DC    F'5'
BASE  EQU   8
TEMP  DS    '1'F
      DROP  8
      END
```

***Sample Output:***

**Intermediate source file generated by Pass-I**

```
PRG2 START
      USING *,BASE
INCR1 DATA1,DATA12
INCR2 DATA3,DATA4
FOUR DC    F'4'
FIVE  DC    F'5'
BASE  EQU   8
TEMP  DS    '1'F
      DROP  8
      END
```

**Macro Name Table (MNT) created by Pass-1**



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

INDEX	MACRO NAME	MDT INDEX
1	INCR1	1
2	INCR2	5

**Argument List Array (ALA) created by Pass-1 for INCR1 Macro**

INDEX	ARGUMENT
1	&FIRST
2	&SECOND

**Argument List Array (ALA) created by Pass-1 for INCR2 Macro**

INDEX	ARGUMENT
1	&ARG1
2	&ARG2

**Macro Definition table (MDT) created by Pass-1**

INDEX	DEFINITION
1	INCR1 &FIRST,&SECOND
2	A 1,#1
3	L 2,#2
4	MEND
5	INCR2 &ARG1,&ARG2
6	L 3,#1
7	ST 4,#2
8	MEND

**Argument List Array (ALA) created by Pass-1 for INCR1 Macro**

INDEX	ARGUMENT
1	&FIRST
2	&SECOND

**Output generated by Pass-II**

**Argument List Array (ALA) created for INCR2 Macro call**

INDEX	ARGUMENT
-------	----------



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

-----  
1            DATA1  
2            DATA12

**Argument List Array (ALA) created for INCR2 Macro call**

**INDEX            ARGUMENT**  
-----

1            DATA3  
2            DATA4

**Expanded Source code generated by Pass-2**

```
PRG2  START
      USING *,BASE
      A    1,DATA1
      L    2,DATA12
      L    3,DATA3
      ST   4,DATA4
FOUR  DC   F'4'
FIVE  DC   F'5'
BASE  EQU  8
TEMP  DS   '1F
      DROP 8
      END
```

**Implementation details:**

```
macro_count = 0
extra_count = 0
mnt = []
mnt_counter = 0
mdt = []
mdt_counter = 0
ala1 = []
ala1_counter = 0
ala2 = []
ala2_counter = 0
final_pass = []

file_hand = open('text3.txt', 'r')
file_1 = open('inter.txt', 'w')
```



## K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

```
for line in file_hand:
    line = line.strip()
    if 'MACRO' in line:
        macro_count = 1
        extra_count = 1
        continue
    if extra_count == 1:
        temp = []
        words = line.split()
        temp.append(mnt_counter)
        temp.append(words[0])
        temp.append(mdt_counter)
        mnt_counter += 1
        mnt.append(temp)
        words = line.split()
        args = words[1].split(',')
        for i in range(len(args)):
            temp = []
            temp.append(ala1_counter)
            if '=' in args[i]:
                pos = args[i].find('=')
                args[i] = args[i][0:pos]
            temp.append(args[i])
            ala1_counter += 1
            ala1.append(temp)
        extra_count = 0
    if macro_count == 1:
        if 'MEND' in line:
            temp = []
            temp.append(mdt_counter)
            temp.append(line)
            mdt_counter += 1
            mdt.append(temp)
            macro_count = 0
        if 'MEND' not in line:
            temp = []
            temp.append(mdt_counter)
            temp.append(line)
            mdt_counter += 1
            mdt.append(temp)
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

```
        else:
            file_1.write(line + '\n')
file_1.close()

print("FIRST PASS")
print()
print("ALA")
for i in range(len(ala1)):
    for j in range(len(ala1[i])):
        print(ala1[i][j], end = '\t\t')
    print()
print()
print("MNT")
for i in range(len(mnt)):
    for j in range(len(mnt[i])):
        print(mnt[i][j], end = '\t\t')
    print()
print()
print("MDT")
for i in range(len(mdt)):
    for j in range(len(mdt[i])):
        print(mdt[i][j], end='\t\t')
    print()
print()

file_2 = open('inter.txt', 'r')
for line in file_2:
    line = line.strip()
    for i in range(len(mnt)):
        if mnt[i][1] in line:
            loc = mnt[i][2]
            words = line.split()
            args2 = words[1].split(',')
            for j in range(len(args2)):
                temp = []
                temp.append(ala2_counter)
                temp.append(args2[j])
                ala2_counter += 1
            ala2.append(temp)
file_2.close()
```



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

```
print("SECOND PASS")
print()
print("ALA")
for i in range(len(ala2)):
    for j in range(len(ala2[i])):
        print(ala2[i][j], end='\t\t')
    print()
print()
f1 = open('inter.txt', 'r')
f2 = open('final.txt', 'w')
for line in f1:
    lc = None
    add_line = ''
    for i in range(len(mnt)):
        if mnt[i][1] in line:
            lc = mnt[i][2]
            break
    if lc == None:
        f2.write(line)
    if lc != None:
        while add_line != 'MEND':
            add_line = str(mnt[lc][1])
            new_line = add_line + '\n'
            f2.write(new_line)
            lc += 1
f1.close()
f2.close()

f3 = open('final.txt', 'r')
f4 = open('macros.txt', 'w')
for line in f3:
    if 'MEND' in line:
        continue
    for i in range(len(ala1)):
        if ala1[i][1] in line:
            line = line.replace(ala1[i][1], ala2[i][1])
f4.write(line)
```





**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

**Output:**

```
FIRST PASS

ALA
0      &FIRST
1      &SECOND
2      &ARG1
3      &ARG2

MNT
0      INCR1      0
1      INCR2      4

MDT
0      INCR1 &FIRST,&SECOND=DATA9
1      A 1,&FIRST
2      L 2,&SECOND
3      MEND
4      INCR2 &ARG1,&ARG2
5      L 3,&ARG1
6      ST 4,&ARG2
7      MEND

SECOND PASS

ALA
0      DATA1
1      DATA12
2      DATA3
3      DATA4
```

**Files:**

1) **text3.txt**

```
MACRO
INCR1 &FIRST,&SECOND=DATA9
```



## **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

```
A 1,&FIRST  
L 2,&SECOND  
MEND  
MACRO  
INCR2 &ARG1,&ARG2  
L 3,&ARG1  
ST 4,&ARG2  
MEND  
PRG2 START  
USING *,BASE  
INCR1 DATA1,DATA12  
INCR2 DATA3,DATA4  
FOUR DC F'4';  
FIVE DC F'5';  
BASE EQU 8  
TEMP DS '1'F  
DROP 8  
END
```

### **2) macros.txt**

```
PRG2 START  
USING *,BASE  
INCR1 DATA1,DATA12=DATA9  
A 1,DATA1  
L 2,DATA12  
INCR2 DATA3,DATA4  
L 3,DATA3  
ST 4,DATA4  
FOUR DC F'4';  
FIVE DC F'5';  
BASE EQU 8  
TEMP DS '1'F  
DROP 8  
END
```

### **3) inter.txt**

```
PRG2 START  
USING *,BASE  
INCR1 DATA1,DATA12  
INCR2 DATA3,DATA4  
FOUR DC F'4';  
FIVE DC F'5';  
BASE EQU 8  
TEMP DS '1'F  
DROP 8
```



## K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

END

#### 4) final.txt

```
PRG2 START
USING *,BASE
INCR1 &FIRST,&SECOND=DATA9
A 1,&FIRST
L 2,&SECOND
MEND
INCR2 &ARG1,&ARG2
L 3,&ARG1
ST 4,&ARG2
MEND
FOUR DC F'4';
FIVE DC F'5';
BASE EQU 8
TEMP DS '1'F
DROP 8
END
```

**Conclusion:** Thus in this experiment understood the concept and importance of a 2 pass Macro Processor and also implemented the same.

### Post Lab Subjective Questions

1. What are the tasks performed by macro processor?

A Macro instruction is the notational convenience for the programmer. For every occurrence of macro the whole macro body or macro block of statements gets expanded in the main source code. Thus Macro instructions makes writing code more convenient.

#### **Silent features of Macro Processor:**

- **Macro** represents a group of commonly used statements in the source programming language.
- Macro Processor replace each macro instruction with the corresponding group of source language statements. This is known as expansion of macros.
- Using Macro instructions programmer can leave the mechanical details to be handled by the macro processor.
- Macro Processor designs are not directly related to the computer architecture on which it runs.
- Macro Processor involves definition, invocation and expansion.

2. What is the purpose of pass1 macro processor?



## **K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

In Pass 1, the macro definitions are searched and stored in the Macro Definition Table and entry is made in the Macro Name Table

3. What is the purpose of pass2 macro processor?

In Pass 2, the macro calls are identified and placed in appropriate place and Macro Calls are replaced by Macro Definition

4. What is the purpose of MDT table?

Store the body of Macro Definition

5. What is the purpose of MNT table?

Store the names of Macros

6. What is the use of MDI & MDLC?

Additional data structures like macro definition indicator (MDI) and macro definition level counter (MDLC) are the switches used to keep track of macro calls and macro definition. The MDI has status "ON" during the expansion of macro call and the value "OFF" all the other times. When its value is "ON" the cards are read from the MDT and when it is "OFF" the cards are read from the input source card. The use of MDLC is used keep track of the level of macros while defining the macros. Initially it is zero and it is incremented each time a MACRO code is found within a macro. The reverse process happens in case of MEND i.e. the value of MDLC is decremented by one each time it encounters a MEND and the process continues till the MDLC is zero i.e. the completion of macro definition.

### **Post Lab objective Questions**

1. In which way a macro processor for assembly language can be implemented?
- Independent two pass processor
  - Independent One pass Processor
  - Processor incorporated into pass I of 2-pass Assembler
  - All of these above

**ANS: D**

2. Advantage of incorporating the Macro Processor in Pass I of assembler is that
- Many functions have to be implemented twice
  - Functions are combined and it is not necessary to create intermediate file as output from the macro processor and input to the assembler.
  - More Flexibility is provided to the programmer in that he may use all the features of the assembler in conjunction with macros.



**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)

- d. All of these above.

**ANS: D**