

Department of Computer Engineering

Academic Term : Jan-May 23-24

Class : T.E. (Computer)

Subject Name : System Programming and Compiler Construction

Subject Code : (CPC601)

Practical No:	8
Title:	One Pass and Two Pass Macroprocessor
Date of Performance:	
Date of Submission:	
Roll No:	9601
Name of the Student:	Ivan Dsouza

Evaluation:

Sr. No	Rubric	Grade
1	Time Line (2)	
2	Output(3)	
3	Code optimization (2)	
4	Postlab (3)	

Signature of the Teacher :

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

Experiment No 8

Aim: Write a program to implement two pass Macro Processor.

Learning Objective: To understand how the pre-processor replaces all the macros in the program by its real definition prior to the compilation process of the program.

Algorithm:

Pass1:

1. Set the 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)
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)
7. Increment MNTC by 1.
8. Prepare ALA
9. Enter macro name into MDT at index MDTC
10. Increment MDTC by 1.
11. Read source statement from source program
12. Create and substitute index notation for arguments in the source statement if any.
13. Enter this line into MDT
14. Increment MDTC by 1.
15. Check if currently read source statement is pseudo-opcode MEND. If yes then goto step 3 else goto step 11.

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

16. Write source program statement as it is in the file
17. Check if pseudo-opcode END is encountered. If yes goto step 18 else goto step 19
18. Goto Pass2
19. Go to step 3
20. End of PASS1.

Pass2:

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

Implementation Details

1. Read input file with Macros

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

2. Display output of Pass1 as the output file, MDT, MNT, and ALA tables.

3. Display output of pass2 as the expanded source file, MDT, MNT and ALA tables.

Test Cases :

1. Call macro whose definition is not present
2. Define macro without MEND

Conclusion:

PASS-1

Code:

```
Spcc > exp8.1 > C main.c > ...
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5
6 int main(){
7     char label[10],opcode[10],operand[10];
8     int mntc=1,mdtc=1;
9
10    FILE *fp1,*fp2,*fp3,*fp4;
11    fp1 = fopen("input.txt","r"); //input
12    fp2 = fopen("mnt.txt","w"); //output
13    fp3 = fopen("mdt.txt","w"); //output
14    fp4 = fopen("copyfile.txt","w");//output
15
16    fscanf(fp1,"%s %s %s",label,opcode,operand);
17
```

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

```
17
18     while(strcmp(opcode,"END")!=0){
19         if(strcmp(opcode,"MACRO")==0){
20             fscanf(fp1,"%s %s %s",label,opcode,operand);
21             fprintf(fp2,"%d %s %d\n",mntc,opcode,mdtc);
22             mntc++;
23             while(strcmp(opcode,"MEND")!=0){
24                 fprintf(fp3,"%d %s %s %s\n",mdtc,label,opcode,operand);
25                 mdtc++;
26                 fscanf(fp1,"%s %s %s",label,opcode,operand);
27             }
28             fprintf(fp3,"%d %s %s %s\n",mdtc,label,opcode,operand);
29             mdtc++;
30         }
31     else{
32         fprintf(fp4,"%s %s %s\n",label,opcode,operand);
33     }
34     fscanf(fp1,"%s %s %s",label,opcode,operand);
35 }
36
37     fprintf(fp4,"%s %s %s\n",label,opcode,operand);
38
39     fclose(fp1);
40     fclose(fp2);
41     fclose(fp3);
42     fclose(fp4);
43
44
45     return 0;
46 }
```

Input:

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

```
Spcc > exp8.1 > ≡ input.txt
1 ** MACRO **
2 ** MATH **
3 ** ar 5,3
4 ** sr 5,4
5 ** MEND **
6 ** MACRO **
7 ** MUL **
8 ** mr 5,3
9 ** MEND **
10 pg1 START 0
11 ** USING *,15
12 ** 1 1,FIVE
13 ** MATH **
14 FIVE DC H'5'
15 ** MATH **
16 ** MUL **
17 ** END **
```

Output:

```
Spcc > exp8.1 > ≡ mdt.txt
1 1 ** MATH **
2 2 ** ar 5,3
3 3 ** sr 5,4
4 4 ** MEND **
5 5 ** MUL **
6 6 ** mr 5,3
7 7 ** MEND **
8
```

```
Spcc > exp8.1 > ≡ mnt.txt
1 1 MATH 1
2 2 MUL 5
3
```

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

```
Spcc > exp8.1 >  ≡ copyfile.txt
 1 pg1 START 0
 2 ** USING *,15
 3 ** 1 1,FIVE
 4 ** MATH **
 5 FIVE DC H'5'
 6 ** MATH **
 7 ** MUL **
 8 ** END **
 9
```

PASS-2

Code:

```
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6
7
8 int main(){
9     char label[10],opcode[10],operand[10],mntc[10],macroname[10],mdtc[10],mdtmdtc[10],ignore[10],mdtlabel[10],mdtvalue[10];
10    int flag = 0;
11
12    FILE *fp1,*fp2,*fp3,*fp4;
13    fp1 = fopen("copyfile.txt","r");           //input
14    fp2 = fopen("mnt.txt","r");                //input
15    fp3 = fopen("mdt.txt","r");                //input
16    fp4 = fopen("expandedSource.txt","w");      //output
17
18    fscanf(fp1,"%s %s %s",label,opcode,operand);
19
20    while(strcmp(opcode,"END")!=0){
21        if(strcmp(opcode,"START")==0 || strcmp(opcode,"USING")==0 || strcmp(opcode,"DC")==0 || strcmp(opcode,"DS")==0 ){
22            fprintf(fp4,"%s %s %s\n",label,opcode,operand);
23        }
24        else{
25            fscanf(fp2,"%s %s %s",mntc,macroname,mdtc);
26            while(!feof(fp2)){
27                if(strcmp(opcode,macroname)==0){
28                    flag = 1;
29                    break;
30                }
31            }
32            fscanf(fp2,"%s %s %s",mntc,macroname,mdtc);
```

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

```
31     fscanf(fp2,"%s %s %s",mntc,macroname,mdtc);
32 }
33 if(strcmp(opcode,macroname)==0){
34     flag = 1;
35 }
36 if(flag==1){
37     flag = 0;
38     fscanf(fp3,"%s %s %s %s",mdtmdtc,ignore,mdtlable,mdtvalue);
39     while(strcmp(mdtc,mdtmdtc)!=0){
40         fscanf(fp3,"%s %s %s %s",mdtmdtc,ignore,mdtlable,mdtvalue);
41     }
42     fscanf(fp3,"%s %s %s %s",mdtmdtc,ignore,mdtlable,mdtvalue);
43     while(strcmp(mdtlable,"MEND")!=0){
44         fprintf(fp4,"%s %s %s\n",ignore,mdtlable,mdtvalue);
45         fscanf(fp3,"%s %s %s %s",mdtmdtc,ignore,mdtlable,mdtvalue);
46     }
47     rewind(fp3);
48 }
49 else{
50     fprintf(fp4,"%s %s %s\n",label,opcode,operand);
51 }
52 rewind(fp2);
53 }
54 fscanf(fp1,"%s %s %s",label,opcode,operand);
```

```
56 |
57     fprintf(fp4,"%s %s %s\n",label,opcode,operand);
58
59     fclose(fp1);
60     fclose(fp2);
61     fclose(fp3);
62     fclose(fp4);
63
64
65     return 0;
66 }
```

Input:

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

Spcc > exp8.1 > \equiv input.txt	1 ** MACRO ** 2 ** MATH ** 3 ** ar 5,3 4 ** sr 5,4 5 ** MEND ** 6 ** MACRO ** 7 ** MUL ** 8 ** mr 5,3 9 ** MEND **
Spcc > exp8.2 > \equiv copyfile.txt	10 pg1 START 0 11 ** USING *,15 12 ** 1 1,FIVE 13 ** MATH ** 14 FIVE DC H'5' 15 ** MATH ** 16 ** MUL ** 17 ** END **

Spcc > exp8.2 > \equiv mdt.txt	1 1 ** MATH ** 2 2 ** ar 5,3 3 3 ** sr 5,4 4 4 ** MEND ** 5 5 ** MUL ** 6 6 ** mr 5,3 7 7 ** MEND ** 8
Spcc > exp8.2 > \equiv mnt.txt	1 1 MATH 1 2 2 MUL 5 3

Output:

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

System Programming and Compiler Construction

VI Semester (Computer)

Academic Year: 2023- 24

```
Spcc > exp8.2 >  ≡ expandedSource.txt
1    pg1 START 0
2    ** USING *,15
3    ** 1 1,FIVE
4    ** ar 5,3
5    ** sr 5,4
6    FIVE DC H'5'
7    ** ar 5,3
8    ** sr 5,4
9    ** mr 5,3
10   ** END **
11
```

Post Lab Questions:

1. **What is meant by macro processor?**
2. **What are the features of macro processor?**

Ans 1] A Macro processor is a program that performs text manipulation tasks on a source program, typically expanding macros into their corresponding code sequences before the actual compiler or assembly process.

Ans 2] Features of Macro Processor:

- i) Macro Definition: Allows users to define macros which are symbolic name representing a sequence of instructions.
- ii) Macro Expansion: Replaces instances of macro calls with their corresponding code sequences.
- iii) Parameterized Macros: Supports macros with parameters enabling customization and reuse.
- iv) Nesting: Allows macros to be nested within each other.
- v) Provides error detection and reporting mechanisms for macro related errors.