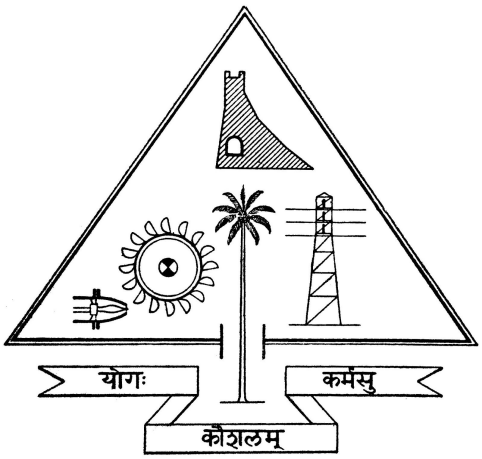

Government Engineering College Thrissur

System Software Lab

Navaneeth D
TCR18CS043
S5, CSE

Experiment 12

Two Pass Macro Assembler

AIM

Implement a two pass macro assembler.

THEORY

A two pass macro assembler is used for identifying the macro name and performing expansion.

Features of macro processor:

- Recognized the macro definition
- Save macro definition
- Recognized the macro call
- Perform macro expansion

Forward reference Problem:

The assembler specifies that the macro definition should occur anywhere in the program. So there can be chances of macro call before it's definition witch gives rise to the forwards reference problem of macro due to which macro is divided into two passes

PASS 1-

- Recognize macro definition
- Save macro definition

PASS 2-

- Recognize macro call
- Perform macro expansion

RESULT

A two pass macro assembler was implemented with successful output.

Output Screenshots

Terminal Output:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

navaneeth@navaneeth-lap:~/Documents/NAV/Cprog$ cd Exp12
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp12$ gcc TCR18CS043_Pass1.c
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp12$ ./a.out
[info] Pass 1 complete. Please verify dtab2.txt
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp12$ gcc TCR18CS043_Pass2.c
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp12$ ./a.out
[info] Pass 2 complete. Please verify atab2.txt and op2.txt
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp12$
```

Files Generated:

Pass 1:

Exp12 > dtab2.txt

```
1  EX1 &A,&B
2  LDA &A
3  STA &B
4  MEND
```

Exp12 > atab2.txt

```
1  N1
2  N2
```

Pass 2:

Exp12 > op2.txt

```
1  SAMPLE  START   1000
2  .   EX1 N1,N2
3  -   LDA N1
4  -   STA N2
5  N1  RESW    1
6  N2  RESW    1
7  -   END -
8
```