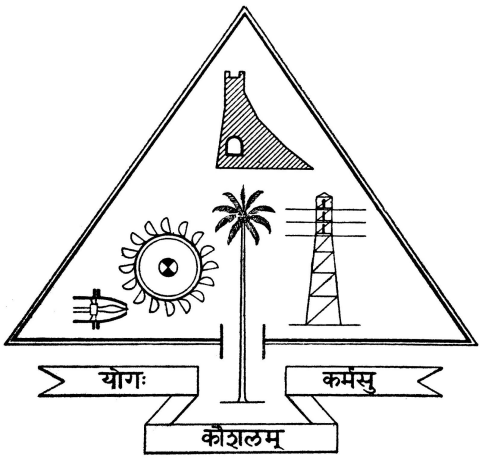

Government Engineering College Thrissur

System Software Lab

Navaneeth D
TCR18CS043
S5, CSE

Experiment 9

2nd Pass of Two-Pass Assembler

AIM

Implement pass two of a two pass assembler. *

THEORY

Assembler is a program for converting instructions written in low-level assembly code into relocatable machine code and generating along information for the loader.

It generates instructions by evaluating the mnemonics (symbols) in the operation field and finding the value of symbols and literals to produce machine code. Now, if an assembler does all this work in one scan then it is called a single pass assembler, otherwise if it does in multiple scans then called multiple pass assembler. Here assembler divide these tasks in two passes:

- **Pass-1:**
    1. Define symbols and literals and remember them in symbol table and literal table respectively.
    2. Keep track of location counter
    3. Process pseudo-operations
- **Pass-2:**
    1. Generate object code by converting symbolic op-code into respective numeric op-code
    2. Generate data for literals and look for values of symbols

RESULT

The 2nd pass of a two-pass assembler was implemented with successful output.

Output Screenshots

INPUT

Test > input.txt

```
1   **  COPY    START   2000
2   2000    **  LDA FIVE
3   2003    **  STA ALPHA
4   2006    **  LDCH    CHARZ
5   2009    **  STCH    C1
6   200C    ALPHA   RESW    1
7   200F    FIVE    WORD    5
8   2012    CHARZ   BYTE    C'Z'
9   2013    C1  RESB    1
10  2014    **  END **
11
```

OUTPUT

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Test$ gcc test.c
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Test$ ./a.out

------OBJECT PROGRAM(RESULT OF 2nd PASS OF 2 PASS FILTER)------

H^COPY^2000^20
T^002000^00200F^23200C^502012^542013^00005^5A^
E^002000
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Test$
```