Experiment 7 **Surya Narayanan**

Pass one of two pass assembler

s5 csb 48

Step 1 : START.
Step 2 : Include necessary header files.
Step 3 : Declare character arrays to read from input, optab.
Step 4 : Goto to function pass one ().
Step 5 : Stop.

Pass one ()

Step 1: START
Step 2: Declare necessary variables and file pointer.
Step 3: Open input and optab files in read only mode
Step 4: Symbol output / intermediate and length files in write mode
Step 5: Read first line from input file.
Step 6: If opcode = START (comparison by strcmp
  Step 6.1 - Convert operand to integer value and assign it to variable start.
  Step 6.2 - Set location counter as start
  Step 6.3 - Write the label, opcode and opcode to the output file and read next line
Step 7. else set location counter as 0

Step 8- Repeat the following till opcode – END

Step 8.1 - write the label, location counter, opcode and operand to output file.

Step 8.2 – If '**' is not encountered, write the corresponding labels and location counter values to output file.

Step 8.3 – Read the mnemonic and opcode from optat

Step 8.4 – Repeat the following till END of optat is obtained

Step 8.4.1 – if the opcode matches with optat code, increment location counter by 3 and go to step 8.4.

Step 8.5 – if opcode = WORD

Step 8.5.1 - increment location counter by 3.

Step 8.6 – if opcode = RESW

Step 8.6.1 : increment location counter by 3 times that of operand.

Step 8.7 – if opcode = BYTE

Step 8.7.1 increment location counter with the operand.

Step 8.6 - if opcode = RESB
Step 8.6.1 - increment location counter with the operand
Step 8.9 - Read the next input line.

Step 9. - write the last line to output file.
Step 10 - close all the files.
Step 11 - Go to function display (.)
Step 12 - Calculate length as location counter start and write the result to length file. Also display the result
Step 13 - close length file.
Step 14 - STOP.

Display

Step 1 - start
Step 2 - Declare necessary variables.
Step 3 - Open input file in read mode.
repeat the following till EOF.
Step 3.1 - Display the character
Step 3.2 - Get the next character.
Step 4 - close the input file
Step 5. Open the output file in read mode.

Step 6: Repeat the following till file end is reached.

Step 6.1 - Display character from file pointer of output file

Step 6.2 - Get the next character

Step 7: close output file

Step 8: Open syntax in read mode Repeat the following till file end is reached

Step 8.1: Display character from file pointer of syntax file

Step 8.2: Get next character

Step 9: close the syntax file

Step 10: STOP

Program

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void passOne(char label[10], char opcode[10], char operand[10], char
code[10], char mnemonic[3]);
void display();
int main()
{
    char label[10], opcode[10], operand[10];
    char code[10], mnemonic[3];
    passOne(label, opcode, operand, code, mnemonic);
    return 0;
}
void passOne(char label[10], char opcode[10], char operand[10], char
code[10], char mnemonic[3])
{
    int locctr, start, length;
    FILE *fp1, *fp2, *fp3, *fp4, *fp5;
    fp1 = fopen("INPUT.TXT", "r");
    fp2 = fopen("OPTAB.TXT", "r");
    fp3 = fopen("SYMTAB.TXT", "w");
    fp4 = fopen("INTERMEDIATE.TXT", "w");
    fp5 = fopen("LENGTH.TXT", "w");
    fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
    if (strcmp(opcode, "START") == 0)
    {
        start = atoi(operand);
        locctr = start;
        fprintf(fp4, "\t%s\t%s\t%s\n", label, opcode, operand);
        fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
    }
    else
    {

        locctr = 0;
    }
    while (strcmp(opcode, "END") != 0)
    {
```

```
        fprintf(fp4, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);
        if (strcmp(label, "**") != 0)
        {
            fprintf(fp3, "%s\t%d\n", label, locctr);
        }
        fscanf(fp2, "%s\t%s", code, mnemonic);
        while (strcmp(code, "END") != 0)
        {
            if (strcmp(opcode, code) == 0)
            {
                locctr += 3;
                break;
            }
            fscanf(fp2, "%s\t%s", code, mnemonic);
        }
        if (strcmp(opcode, "WORD") == 0)
        {
            locctr += 3;
        }
        else if (strcmp(opcode, "RESW") == 0)
        {
            locctr += (3 * (atoi(operand)));
        }
        else if (strcmp(opcode, "BYTE") == 0)
        {
            ++locctr;
        }
        else if (strcmp(opcode, "RESB") == 0)
        {
            locctr += atoi(operand);
        }

        fscanf(fp1, "%s\t%s\t%s", label, opcode, operand);
    }
    fprintf(fp4, "%d\t%s\t%s\t%s\n", locctr, label, opcode, operand);
    fclose(fp4);
    fclose(fp3);
    fclose(fp2);
    fclose(fp1);
    display();
```

```
    length = locctr - start;
    fprintf(fp5, "%d", length);
    fclose(fp5);
    printf("\nThe length of the code : %d\n", length);
}
void display()
{
    char str;
    FILE *fp1, *fp2, *fp3;
    printf("\nThe contents of Input Table :\n\n");
    fp1 = fopen("INPUT.TXT", "r");
    str = fgetc(fp1);
    while (str != EOF)
    {
        printf("%c", str);
        str = fgetc(fp1);
    }
    fclose(fp1);
    printf("\n\nThe contents of Output Table :\n\n");
    fp2 = fopen("INTERMEDIATE.TXT", "r");
    str = fgetc(fp2);

    while (str != EOF)
    {
        printf("%c", str);
        str = fgetc(fp2);
    }
    fclose(fp2);
    printf("\n\nThe contents of Symbol Table :\n\n");
    fp3 = fopen("SYMTAB.TXT", "r");
    str = fgetc(fp3);
    while (str != EOF)
    {
        printf("%c", str);
        str = fgetc(fp3);
    }
    fclose(fp3);
}
```

Input

INPUT.TXT

```
**  START   1000
**  LDA FIVE
**  STA ALPHA
**  LDCH    CHARZ
**  STCH    C1
ALPHA   RESW    1
FIVE    WORD    5
CHARZ   BYTE    C'Z'
C1  RESB    1
**  END **
```

OPTAB.TXT

```
LDA 100
STA 23
LDCH 01
STCH 05
END
```

Output

```
/mnt/e/school/ss/7 07:12 PM  > ./a.out

The contents of Input Table :

**  START   1000
**  LDA FIVE
**  STA ALPHA
**  LDCH    CHARZ
**  STCH    C1
ALPHA   RESW    1
FIVE    WORD    5
CHARZ   BYTE    C'Z'
C1  RESB    1
**  END **

The contents of Output Table :

        **      START   1000
1000    **      LDA     FIVE
1003    **      STA     ALPHA
1006    **      LDCH    CHARZ
1009    **      STCH    C1
1012    ALPHA   RESW    1
1015    FIVE    WORD    5
1018    CHARZ   BYTE    C'Z'
1019    C1      RESB    1
1020    **      END     **


The contents of Symbol Table :

ALPHA   1012
FIVE    1015
CHARZ   1018
C1      1019

The length of the code : 20
/mnt/e/school/ss/7 07:14 PM  >
```