

SYSTEM PROGRAMING LAB WORK (Credits: Sarvat Mir)

PROGRAM 1

*/*Display symbol table of a sample program stored in file named spp.txt.The symbol table is stored in the form of text file named symtab.txt */*

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<iostream.h>
#include<string.h>
FILE *f1,*f3;//f1 is pointer for spp.txt,f3 is for symbol table
int lc,sa,i=0,j=0,k,len1=1; //lc is location counter,len1 keeps track of instr length
char name[20],opnd[30],la[20],mne[20],s1[20];
//these hold various parts of inst

void addsymbol()
{ //if(strcmp(la,"BEGIN")==0) //for first instru
    if(strcmp(mne,"BALR")==0) //if mnemonic is balr length of instruction is 2
        len1=2;
    else if(strcmp(mne,"START")==0)
        len1=1;
    else
        len1=4;

    if(strcmp(la,"X")!=0 && strcmp(mne,"EQU")!=0) //if label is present in the instruction
        //and its not a constant then its relative
        fprintf(f3,"%s\t\t\t%d\t\t\t%s\n",la,len1,"R");
    else if(strcmp(la,"X")!=0 && strcmp(mne,"EQU")==0) //if label is present in the instruction
        //and its a constant then its absolute
```

```
fprintf(f3,"%s\t\t\t%d\t\t\t%s\n",la,len1,"A");
```

```
}
```

```
void readf(FILE *fp)
```

```
{
```

```
rewind(fp);
```

```
char ch=fgetc(fp);
```

```
while(ch!=EOF)
```

```
{
```

```
if(ch!='X')
```

```
cout<<ch; ch=fgetc(fp);
```

```
}
```

```
getch();
```

```
}
```

```
void main()
```

```
{
```

```
clrscr();
```

```
f1=fopen("spp.txt","r"); //open the text file containing program in read mode
```

```
// f2=fopen("optab.txt","r");
```

```
f3=fopen("symtab.txt","w+"); //table files to be created are opened in write mode
```

```
char ch;
```

```
fscanf(f1,"%s%s",la,mne); //takes ist instr,stores its label in la, opcode mnemonic in mne
```

```
sa=atoi(opnd); //converts string to int and stores value in sa
```

```
strcpy(name,la); //copies label to name
```

```
lc=sa;//sets location counter to 0 when ist instruction is encountered ,sets location counter
```

```
fprintf(f3,"%s\t\t\t%s\t\t\t%s\n\n","symbol","length","Relocation");
```

```
fprintf(f3,"%s\t\t\t%s\t\t\t%s\n",la,"1","R"); //add ist label to symbol table
```

```
//*****now dealing with next instructions
```

```
//while(strcmp(mne,"END")!=0)
```

```

for(j=0;j<15;j++)
{
fscanf(f1, "%s%s%s",la,mne,opnd);
addsymbol();//add label and length of instruction to symbol table

lc+=1; //move to location counter+1
} //endwhile

//READ FILES ONE BY ONE                2
cout<<"****READING THE INPUT FILE:...loading.."<<endl<<endl;
readf(f1); //display original program
fclose(f3);
f3=fopen("symtab.txt","r");
getch();
clrscr();
cout<<"****READING THE SYMBOL TABLE FILE:...loading.."<<endl<<endl;
readf(f3); //display symbol table
getch();
fclose(f3);
getch();
}

```

OUTPUT : (ON LEFT-> INPUT PROGRAM, ON RIGHT->SYMBOL TABLE OF INPUT PROGRAM)

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

****READING THE INPUT FILE:...loading..

```
TEST START
BEGIN BALR 15,0
USING BEGIN+2,15
SR 4,4
L 3,TEN
LOOP L 2,DATA(4)
A 2,FORTY9
ST 2,DATA(4)
A 4,FOUR
BCT 3,LOOP
BCR 15,14
TEN DC F'10'
FOUR DC F'4'
FORTY9 DC F'49'
DATA DC F'1,3,3,3,3,4,5,8,9,0'
END
```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

****READING THE SYMBOL TABLE FILE:...loading..

symbol	length	Relocation
TEST	1	R
BEGIN	2	R
LOOP	4	R
TEN	4	R
FOUR	4	R
FORTY9	4	R
DATA	4	R

PROGRAM 2

/*Display symbol table and psuedo op table for a given program

stored in a text file named spp.txt.The symbol table is stored in the from of text file named

symtab.txt, psuedo op table is stored in pot.txt and machine op table is stored as mot.txt */

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

```
#include<conio.h>
```

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

```
#include<iostream.h>
```

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

```
FILE *f1,*f2,*f3,*f4,*f5;//f1 is pointer for spp.txt,f2 is for mot, f3 is for symbol table
```

```
//f4 is for psuedo op table
```

```
int lc,sa,i=0,j=0,k,len1=1; //lc is location counter,len1 keeps track of instr length
```

```
char name[20],opnd[30],la[20],mne[20],s1[20];
```

```
//these hold various parts of inst
```

```
void addsymbol()
```



```

{
    if(ch!='X')
        cout<<ch;    ch=fgetc(fp);
}
getch();
}
void main()
{

    clrscr();

    f1=fopen("spp.txt","r"); //open the text file containing program in read mode
    // f2=fopen("optab.txt","r");

    f3=fopen("symtab.txt","w+"); //table files to be created are opened in write mode
    f2=fopen("mot.txt","w+");
    f4=fopen("pot.txt","w+");

    char ch;

    fscanf(f1,"%s%s",la,mne); //takes ist instr,stores its label in la, opcode mnemonic in mne
    sa=atoi(opnd);          //converts string to int and stores value in sa
    strcpy(name,la);         //copies label to name

    lc=sa;//sets location counter to 0 when ist instruction is encountered ,sets location counter

    fprintf(f3,"%s\t\t\t%s\t\t\t%s\n\n","symbol","length","Relocation");
    fprintf(f3,"%s\t\t\t%s\t\t\t%s\n",la,"1","R"); //add ist label to symbol table
    fprintf(f4,"%s\t\t\t\t\t%s\n\n","PSEUDO-OP","PROCEDURE");
    fprintf(f2,"%s\t\t%s\t\t%s\n\n","MACHINE OP","INSTRUCTION LENGTH","INSTRUCTION FORMAT");

    //*****now dealing with next instructions
    //while(strcmp(mne,"END")!=0)

    for(j=0;j<15;j++)
    {
        fscanf(f1,"%s%s%s",la,mne,opnd);

        addsymbol();//add label and length of instruction to symbol table

        if(strcmp(mne,"START")==0 | strcmp(mne,"END")==0 | strcmp(mne,"USING")==0 | strcmp(mne,"DC")==0 | strcmp(mne,"EQU")==0)

```

```

addpot(); //add psuedo ops to pot.txt

else //if not a psuedo op add to machine op table

addmot();

lc+=1; //move to location counter+1

// cout<<"la "<<la<<" mne "<<mne<<" opnd "<<opnd;

// cout<<" lc"<<lc<<endl;

} //endwhile

//READ FILES ONE BY ONE                2

cout<<"****READING THE INPUT FILE:...loading.."<<endl<<endl;

readf(f1); //display original program

fclose(f3);

f3=fopen("symtab.txt","r");

getch();

clrscr();

cout<<"****READING THE SYMBOL TABLE FILE:...loading.."<<endl<<endl;

readf(f3); //display symbol table

getch();

fclose(f3);

clrscr();

cout<<"****READING THE PSUEDO-OP TABLE FILE:...loading.."<<endl<<endl;

fclose(f4);

f4=fopen("pot.txt","r");

readf(f4); //display psuedo op table

fclose(f4);

fclose(f2);

clrscr();

cout<<"****READING THE MACHINE-OP TABLE FILE:...loading.."<<endl<<endl;

f2=fopen("mot.txt","r");

readf(f2);

getch();

fclose(f2);

getch();

```

OUTPUT: SHOWS INPUT FILE, SYMBOL TABLE ,MACHINE OPERATION TABLE AND PSUEDO OPERATION TABLE

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0
****READING THE INPUT FILE:...loading..
TEST START
BEGIN BALR 15,0
USING BEGIN+2,15
SR 4,4
L 3,TEN
LOOP L 2,DATA(4)
A 2,FORTY9
ST 2,DATA(4)
A 4,FOUR
BCT 3,LOOP
BCR 15,14
TEN DC F'10'
FOUR DC F'4'
FORTY9 DC F'49'
DATA DC F'1,3,3,3,3,4,5,8,9,0'
END
  
```

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
****READING THE SYMBOL TABLE FILE:...loading..
symbol          length          Relocation
TEST            1                R
BEGIN           2                R
LOOP            4                R
TEN             4                R
FOUR            4                R
FORTY9          4                R
DATA            4                R
  
```

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
****READING THE MACHINE-OP TABLE FILE:...loading..
MACHINE OP      INSTRUCTION LENGTH  INSTRUCTION FORMAT
SR              2                    000
L               4                    001
L               4                    001
A               4                    001
A               4                    001
BCT             4                    001
  
```

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
****READING THE PSUEDO-OP TABLE FILE:...loading..
PSEUDO-OP      PROCEDURE
USING          P1USING
DC             P1DC
DC             P1DC
DC             P1DC
DC             P1DC
DC             P1DC
END            P1END
  
```

TEXT FILE USED(SPP.TXT):

```

TEST START
BEGIN BALR 15,0
X USING BEGIN+2,15
X SR 4,4
X L 3,TEN
LOOP L 2,DATA(4)
  
```



```
X A 2,FORTY9
X ST 2,DATA(4)
X A 4,FOUR
X BCT 3,LOOP
X BCR 15,14
TEN DC F'10'
FOUR DC F'4'
FORTY9 DC F'49'
DATA DC F'1,3,3,3,3,4,5,8,9,0'
X END
```