Aumkar Gadekar

Rno. 21

Batch B

**EXPERIMENT 3**

**Aim :** To implement a two-pass macro processor in C for SIC code.

**Theory :**

A macro name is an abbreviation, which stands for some related lines of code. Macros are useful for the following purposes:

· To simplify and reduce the amount of repetitive coding

· To reduce errors caused by repetitive coding

· To make an assembly program more readable.

A macro consists of name, set of formal parameters and body of code. The use of macro name with set of actual parameters is replaced by some code generated by its body. This is called macro expansion.

Macros allow a programmer to define pseudo operations, typically operations that are generally desirable, are not implemented as part of the processor instruction, and can be implemented as a sequence of instructions. Each use of a macro generates new program instructions, the macro has the effect of automating writing of the program.

Two-pass macro processor :

* All macro definitions are processed during the first pass.
* All macro invocation statements are expanded during the second pass.
* Two-pass macro processor would not allow the body of one macro instruction to contain definitions of other macros.
* The macro names are entered into NAMTAB, NAMTAB contains two pointers to the beginning and the end of the definition in DEFTAB
* The third data structure is an argument table ARGTAB, which is used during the expansion of macro invocations.
* The arguments are stored in ARGTAB according to their position in the argument list.

**Input :**

Input file - input.txt

COPY START 0

RDBUFF MACRO &INDEV,&BUFADR,&RECLTH

CLEAR X

CLEAR A

CLEAR S

+LDT #4096

TD =X'&INDEV'

JEQ \*-3

RD =X'&INDEV'

COMPR A,S

JEQ \*+11

STCH &BUFADR,X

TIXR T

JLT \*-19

STX &RECLTH

MEND

WRBUFF MACRO &OUTDEV,&BUFADR,&RECLTH

CLEAR X

LDT &RECLTH

LDCH &BUFADR,X

TD =X'&OUTDEV'

JEQ \*-3

WD =X'&OUTDEV'

TIXR T

JLT \*-14

MEND

FIRST STL RETADR

CLOOP RDBUFF F1,BUFFER,LENGTH

LDA LENGTH

COMP #0

JEQ ENDFIL

WRBUFF 05,BUFFER,LENGTH

J CLOOP LOOP

ENDFIL WRBUFF 05,EOF,THREE

J RETADR

EOF BYTE C'EOF'

THREE WORD 3

RETADR RESW 1

LENGTH RESW 1

BUFFER RESB 4096

END FIRST

**Code :**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stdbool.h>

typedef struct inst

{

char label[20];

char operand[20];

char opcode[20];

}inst;

typedef struct symbol

{

char label[20];

int loc;

int arg\_count;

char args[10][10];

}symbol;

char \*replaceWord(const char \*s, const char \*oldW, const char \*newW)

{

char \*result;

int i, cnt = 0;

int newWlen = strlen(newW);

int oldWlen = strlen(oldW);

for (i = 0; s[i] != '\0'; i++)

{

if (strstr(&s[i], oldW) == &s[i])

{

cnt++;

i += oldWlen - 1;

}

}

result = (char \*)malloc(i + cnt \* (newWlen - oldWlen) + 1);

i = 0;

while (\*s)

{

if (strstr(s, oldW) == s)

{

strcpy(&result[i], newW);

i += newWlen;

s += oldWlen;

}

else

result[i++] = \*s++;

}

result[i] = '\0';

return result;

}

int main()

{

int i\_count=0,m\_count=0;

FILE \*fp,\*fp1,\*fp2,\*fp3,\*fp4,\*fp5,\*fp6,\*fp7,\*fp8,\*fp9;

char line[256];

fp = fopen("Input.txt", "r"); // read mode

fp1= fopen("Input.txt", "r"); // read mode

fp2= fopen("NAMETAB.txt","w");

fp3= fopen("DEFTAB.txt","w");

fp5= fopen("MacroOutPut.txt","w");

fp6= fopen("output1.txt","w");

fp7= fopen("output1.txt","r");

fp4= fopen("output1.txt", "r"); // read mode

fp8= fopen("OUTPUTFINAL.txt","w");

char label[200],opcode[200],operand[200];

inst DEFTAB[300];

symbol NAMETAB[50];

bool mstart=false;

while (fgets(line, sizeof(line), fp))

{

int i=0;

int t\_count=1;

while(i<strlen(line))

{

if(line[i]==' ')

t\_count+=1;

i++;

}

if(t\_count==3)

{

fscanf(fp1, "%s %s %s", &label, &opcode, &operand);

}

else if(t\_count==2)

{

fscanf(fp1, "%s %s", &opcode, &operand);

strcpy(label,"");

}

else

{

fscanf(fp1, "%s", &opcode);

strcpy(label,"");

strcpy(operand,"");

}

if(mstart==false)

{

if(strcmp(opcode,"MACRO")==0)

{

strcpy(NAMETAB[m\_count].label,label);

NAMETAB[m\_count].loc=i\_count;

strcpy(DEFTAB[i\_count].label,label);

strcpy(DEFTAB[i\_count].opcode,opcode);

strcpy(DEFTAB[i\_count].operand,operand);

int count=0,pos=0;

for(int t=0;t<strlen(operand);t++)

{

if(operand[t]=='&')

{

count+=1;

pos=0;

NAMETAB[m\_count].args[count-1][pos]=operand[t];

pos+=1;

}

else

{

if(operand[t]!=',')

{

NAMETAB[m\_count].args[count-1][pos]=operand[t];

pos+=1;

}

}

}

NAMETAB[m\_count].arg\_count=count;

m\_count+=1;

i\_count+=1;

mstart=true;

fprintf(fp2,"%s\t%d\n",label,i\_count);

fprintf(fp3,"%s",line);

}

else

{

fprintf(fp6,"%s",line);

}

}

else

{

if(strcmp(opcode,"MEND")==0)

{

mstart=false;

}

fprintf(fp3,"%s",line);

strcpy(DEFTAB[i\_count].label,label);

strcpy(DEFTAB[i\_count].opcode,opcode);

strcpy(DEFTAB[i\_count].operand,operand);

i\_count+=1;

}

}

printf("PASS ONE COMPLETED\n");

printf("NAME TABLE AND DEFINITION TABLE GENERATED\n");

for(int k=0;k<m\_count;k++)

{

printf("MACRO NAME: %s\t MACRO POSITION :%d\t NUMBER OF ARGUMENTS: %d\n ARGUMENTS:\t",NAMETAB[k].label,NAMETAB[k].loc+1,NAMETAB[k].arg\_count);

for(int p=0;p<NAMETAB[k].arg\_count;p++)

{

for(int i=1;i<strlen(NAMETAB[k].args[p]);i++)

printf("%c",NAMETAB[k].args[p][i]);

printf("\t");

}

printf("\n");

}

fclose(fp6);

fclose(fp1);

fclose(fp2);

fclose(fp3);

fclose(fp5);

printf("\n\nPASS TWO\n");

bool started=0;

int ct=-1;

int prev=0;

while (fgets(line, sizeof(line), fp7))

{

int i=0;

int t\_count=1;

while(i<strlen(line))

{

if(line[i]==' ')

t\_count+=1;

i++;

}

if(t\_count==3)

{

fscanf(fp4, "%s %s %s", &label, &opcode, &operand);

}

else if(t\_count==2)

{

fscanf(fp4, "%s %s", &opcode, &operand);

}

else

{

fscanf(fp4, "%s", &opcode);

}

int flag=0;

int z;

for(z=0;z<m\_count;z++)

{

if(strcmp(NAMETAB[z].label,opcode)==0)

{

printf("MACRO CALL DETECTED\n");

if(t\_count==3)

fprintf(fp8,"%s ",label);

flag=1;

int w=NAMETAB[z].loc+1;

int c=NAMETAB[z].arg\_count;

char arg[20][20];

int pos=0;

prev=ct+1;

ct=ct+1;

for(int s=0;s<strlen(operand);s++)

{

if(operand[s]!=',')

{

arg[ct][pos]=operand[s];

pos+=1;

}

else

{

ct+=1;

pos=0;

}

}

fp9= fopen("DEFTAB.txt","r");

int u=0;

char l[50];

while(u<NAMETAB[z].loc+1)

{

fgets(l, sizeof(l), fp9);

u+=1;

}

while(strcmp(DEFTAB[w].opcode,"MEND")!=0 && fgets(l, sizeof(l), fp9))

{

char \*result[10];

for(int q=0;q<=c;q++)

result[q]=NULL;

result[0]=l;

int y;

for(y=0;y<c;y++)

{

result[y+1]=replaceWord(result[y],NAMETAB[z].args[y],arg[prev+y]);

}

fprintf(fp8,"%s",result[y]);

for(int q=0;q<=c;q++)

free(result[q]);

w+=1;

}

fclose(fp9);

break;

}

}

if(flag==0)

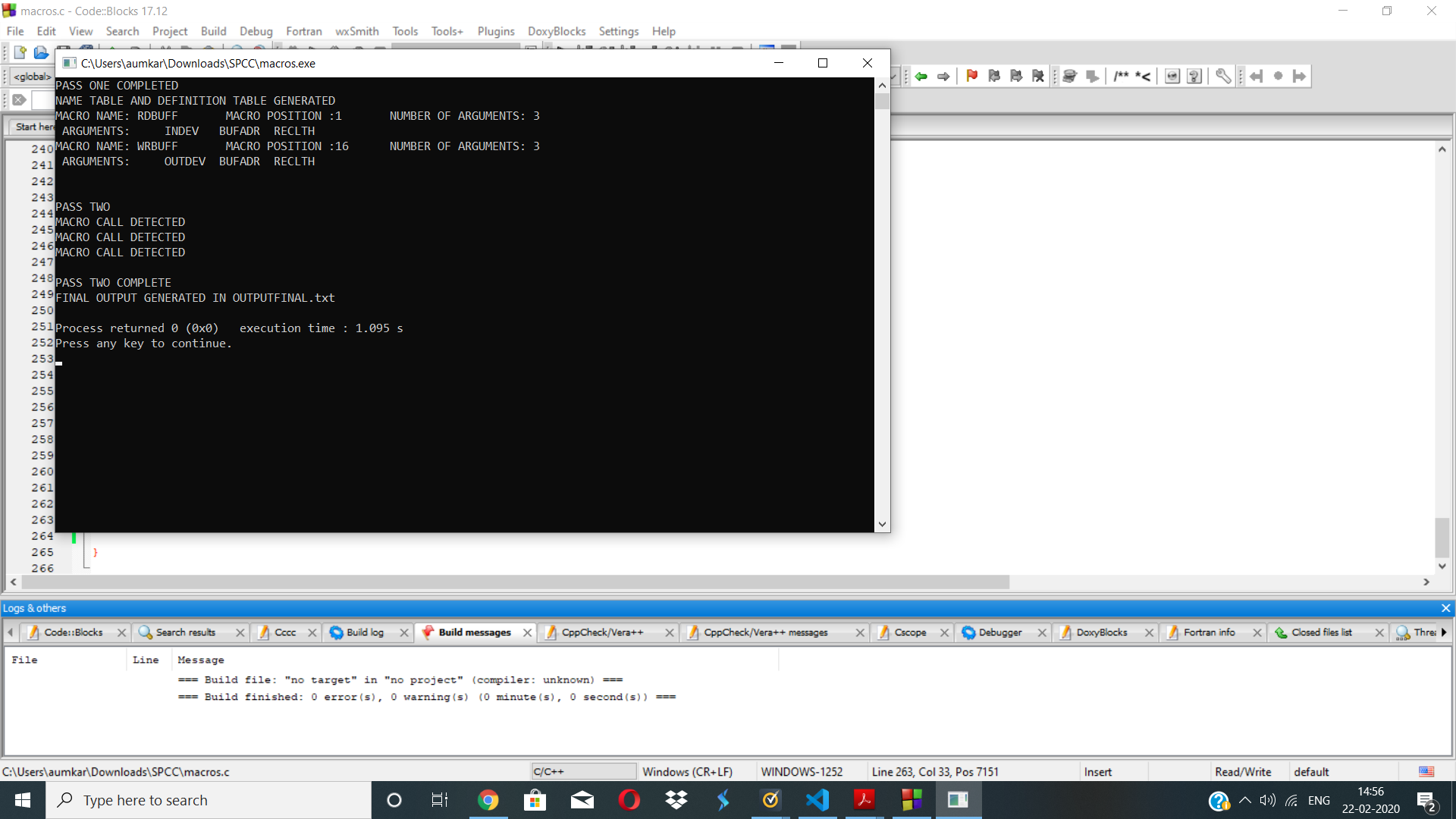
fprintf(fp8,"%s",line);

}

printf("\nPASS TWO COMPLETE\nFINAL OUTPUT GENERATED IN OUTPUTFINAL.txt\n");

}

**Output :**

****

Final Output file (OUTPUTFINAL.txt ):

COPY START 0

FIRST STL RETADR

CLOOP CLEAR X

CLEAR A

CLEAR S

+LDT #4096

TD =X'F1'

JEQ \*-3

RD =X'F1'

COMPR A,S

JEQ \*+11

STCH BUFFER,X

TIXR T

JLT \*-19

STX LENGTH

LDA LENGTH

COMP #0

JEQ ENDFIL

CLEAR X

LDT LENGTH

LDCH BUFFER,X

TD =X'05'

JEQ \*-3

WD =X'05'

TIXR T

JLT \*-14

J CLOOP LOOP

ENDFIL CLEAR X

LDT THREE

LDCH EOF,X

TD =X'05'

JEQ \*-3

WD =X'05'

TIXR T

JLT \*-14

J RETADR

EOF BYTE C'EOF'

THREE WORD 3

RETADR RESW 1

LENGTH RESW 1

BUFFER RESB 4096

END FIRST

**Conclusion :** Thus the two-pass macro processor was understood and implemented.