|  |  |
| --- | --- |
| **Ex.no.2** | **PASS ONE OF TWO PASS ASSEMBLER** |
| 17.10.20 |

### Header

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<math.h>

typedef struct opnode{

char label[20];

char mne[20];

char op[20];

}opnode;

opnode token();

char\* cal\_loc(char[],char[],char\*);

int mne\_value(char[]);

void fremove();

### Implementation

#include"head.h"

char\* cal\_loc(char op[],char c[],char temp[]){

int len=strlen(temp);

int i=0,j,res=0,z=len-1;

for (i=0;i<len;i++){

if(temp[i]=='A')

j=10;

else if(temp[i]=='B')

j=11;

else if(temp[i]=='C')

j=12;

else if(temp[i]=='D')

j=13;

else if(temp[i]=='E')

j=14;

else if(temp[i]=='F')

j=15;

else{

char \*array=(char\*)malloc(2);

array[0] = temp[i];

array[1] = '\0';

j=atoi(array);

free(array);

}

res=res+(j\*pow(16, z));

z-=1;

}

if(strcmp(c,"BYTE")==0){

if(op[0]=='X'){

res=res+strlen(op)-3;

}

else{

res=res-3+strlen(op)/2;

}

}

else if(strcmp(c,"RESW")==0){

res+=3\*(atoi(op));

}

else if(strcmp(c,"RESB")==0){

res+=atoi(op);

}

else{

res+=3;

}

char \*ans=(char\*)malloc(5);

sprintf(ans,"%X",res);

return ans;

}

int mne\_value(char mne[]){

FILE \*f=fopen("opcode.txt","r");

int val;

char temp\_mne[20];//,\*val=malloc(sizeof(20));

while(!feof(f)){

fscanf(f,"%s %d",temp\_mne,&val);

if(strcmp(mne,temp\_mne)==0){

return val;

}

}

fclose(f);

}

opnode token(){

int n,i=0,f=0;

char l[20],m[20],o[20];

opnode t;

FILE \*f1= fopen("alp.txt", "r");

while(n!='\n'&&n!=EOF){

n=fgetc(f1);

if(n!='#'&&f==0){

l[i]=(char)n;

i+=1;

}

else if(n=='#'&&f==0){

l[i]='\0';

i=0;

strcpy(t.label,l);

//printf("l-%s\n",l);

//printf("t.l-%s\n",t.label);

f=1;

}

else if(n!='#'&&f==1){

m[i]=(char)n;

i+=1;

}

else if(n=='#'&&f==1){

m[i]='\0';

i=0;

strcpy(t.mne,m);

f=2;

}

else if(n!='#'&&f==2){

o[i]=(char)n;

i+=1;

}

o[i]='\0';

strcpy(t.op,o);

}

fclose(f1);

return t;

}

void fremove(){

FILE \*f1= fopen("alp.txt", "r");

FILE \*f2=fopen("a.txt","w");

char str[25];

int ln=0,c=0;

while (!feof(f1))

{

strcpy(str, "\0");

fgets(str, 25, f1);

if (!feof(f2))

{

if (c != ln)

{

fprintf(f2, "%s", str);

}

c++;

}

}

fclose(f1);

fclose(f2);

remove("alp.txt");

rename("a.txt","alp.txt");

}

### Application

#include"head.h"

int main(){

char loc[20];

FILE \*f1,\*f2,\*f3,\*f4,\*f5;

opnode temp;

char li[30],la[20],ma[20],oa[20];

int start\_ad=0;

f1=fopen("alp.txt","r");

f3=fopen("output.txt","w+");

f4=fopen("outop.txt","w");

f5=fopen("outsym.txt","w");

fscanf(f1,"%s",li);

char \*ptr=strtok(li,"#");

int flag=0;

while(ptr!=NULL){

if(flag==0){

strcpy(la,ptr);

flag++;

}

else if(flag==1){

strcpy(ma,ptr);

flag++;

}

else if(flag==2){

strcpy(oa,ptr);

}

else{

printf("1st line over\n");

}

ptr=strtok(NULL,"#");

}

printf("hi\n");

if(strcmp("START",ma)==0){

if(strcmp("1000",oa)==0)

start\_ad=1000;

strcpy(loc,"1000");

}

else{

start\_ad=0;

strcpy(loc,"0000");

}

printf("Assigned\n");

printf("Loc Start : %s\n",loc);

int n,i=0,f=0;

opnode final[10];

final[0]=token();

printf("%s %s %s\n",final[0].label,final[0].mne,final[0].op);

fprintf(f3,"%s %s %s %s\n",loc,final[0].label,final[0].mne,final[0].op);

if (strcmp(final[0].label,"")!=0){

fprintf(f5,"%s %s\n",loc,final[0].label);

}

fremove();

i++;

while(strcmp(final[i-1].mne,"END")!=0){

final[i]=token();

printf("%s %s %s\n",final[i].label,final[i].mne,final[i].op);

char \*temp\_loc=cal\_loc(final[i].op,final[i].mne,loc);

strcpy(loc,temp\_loc);

printf("loc : %s",loc);

fprintf(f3,"%s %s %s %s\n",loc,final[i].label,final[i].mne,final[i].op);

if(strcmp(final[i].mne,"START")!=0 && strcmp(final[i].mne,"END")!=0 && strcmp(final[i].mne,"WORD")!=0 && strcmp(final[i].mne,"BYTE")!=0 && strcmp(final[i].mne,"RESW")!=0 &&strcmp(final[i].mne,"RESB")!=0){

int temp\_mne=mne\_value(final[i].mne);

fprintf(f4,"%s %d\n",final[i].mne,temp\_mne);

}

if (strcmp(final[i].label,"")!=0){

fprintf(f5,"%s %s\n",loc,final[i].label);

}

fremove();

i++;

}

fclose(f1);

fclose(f3);

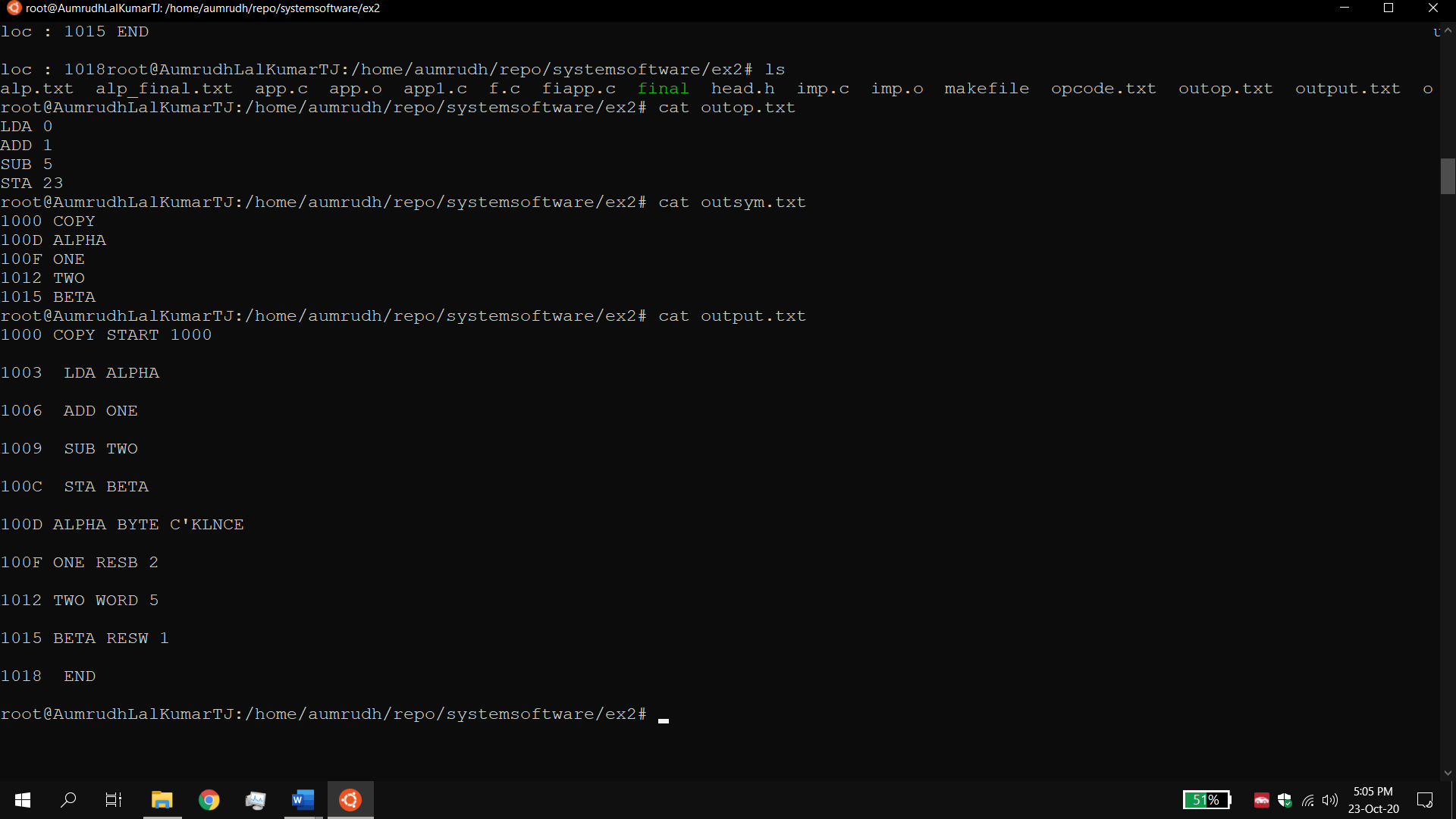
fclose(f4);

fclose(f5);

}

### Output

### 



## Result:

The pass 1 of linking loader is executed successfully. The location counter for each line is assigned and the opcode and symtab is created.