\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14BIT058 ROY JASPER CS

EX14 > LINKING LOADER

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<math.h>

char yy[10][100];

char xx[10][100];

char \*get(char ll[100])

{

FILE \*fp2;

char u[100],v[100];

fp2=fopen("def.c","r");

while(!feof(fp2))

{

fscanf(fp2,"%s %s ",u,v);

if(strcmp(ll,u)==0)

{

int i=strlen(v);

int k=0,j=0;

char \*he = (char\*)malloc(sizeof(char)\*100);

for (j = 0 ;j<i;j++)

{

he[k++]=v[j];

}

he[k]='\0';

return he;

}

}

fclose(fp2);

}

void binary(char x)

{

if((int)x==48)

{

printf("0000 ");

}

else if((int)x==49)

{

printf("0001 ");

}

else

{

int c[10],e[10];

int r,l,a,i=0,count=0;

a=(int)x;

if(a>57)

a=a-55;

else

a=a-48;

if(a<4)

printf("00");

else if(a>=4 && a<8)

printf("0");

while(a!=0)

{

c[i] = a%2;

a = a / 2;

count++;

i++;

}

for(i=count-1; i>=0; i--)

{

printf("%d", c[i]);

}

printf(" ");

}

}

char \*dectohexa(int quotient)

{

int temp,i=1,j;

char h[100];

while(quotient!=0){

temp = quotient % 16;

if( temp < 10)

temp =temp + 48;

else

temp = temp + 55;

h[i++]= temp;

quotient = quotient / 16;

}

int k=0;

char \*he = (char\*)malloc(sizeof(char)\*100);

for (j = i -1 ;j> 0;j--)

he[k++]=h[j];

he[k]='\0';

return he;

}

int search(char a[],int n)

{

int i;

if(n<0)

return 99;

for(i=0;i<=n;i++)

{

if(strcmp(a,yy[i])==0)

{

return i;

}

}

return 99;

}

int check(char x)

{

if(x=='\n' || x=='^')

return 0;

else

return 1;

}

main()

{

FILE \*fp;

fp=fopen("p.c","r");

char x,y[100],z[100],s[100],loc[100],n[100],l[100],start[100],name[100],length[100];

int g,h,f;

printf("\n enter the relocation address :");

scanf("%s",length);

sscanf(length,"%X",&f);

x=fgetc(fp);

int k=0,i=0,j;

while(x!=EOF)

{

if(x=='M')

{

x=fgetc(fp);

x=fgetc(fp);

j=0;

while(x!='^')

{

yy[i][j]=x;

x=fgetc(fp);

j++;

}

yy[i][j]='\0';

sscanf(yy[i],"%X",&h);

g=f+h;

strcpy(yy[i],dectohexa(g));

x=fgetc(fp);

x=fgetc(fp);

x=fgetc(fp);

x=fgetc(fp);

char g,ki[100];

g=x;

x=fgetc(fp);

k=0;

while(x!='\n')

{

ki[k]=x;

x=fgetc(fp);

k++;

}

ki[k]='\0';

int u=search(yy[i],i-1);

if(u==99)

{

strcpy(xx[i],get(ki));

i++;

}

else

{

char qq[100],rr[100];

int ee,ff,mm;

strcpy(qq,get(ki));

strcpy(rr,xx[u]);

sscanf(qq,"%X",&ee);

sscanf(rr,"%X",&ff);

if(g=='-')

{

mm=ff-ee;

}

else

{

mm=ff+ee;

}

strcpy(xx[u],dectohexa(mm));

}

}

else

{

while(x!='\n')

{

x=fgetc(fp);

}

}

x=fgetc(fp);

}

rewind(fp);

for(j=0;j!=i;j++)

{

// printf("\n %s :%s\n",yy[j],xx[j]);

}

int count=i;

printf("\n \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\tPASS2 OF LINKING LOADER\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n ");

x=fgetc(fp);

while(x!=EOF)

{

if(x=='H')

{

x=fgetc(fp);

while(x!='\n')

{

x=fgetc(fp);

}

}

else if(x=='M'|| x=='R'||x=='D')

{

x=fgetc(fp);

while(x!='\n')

{

x=fgetc(fp);

}

}

else if(x=='T')

{

i=0;

x=getc(fp);

x=getc(fp);

while(x!='^')

{

y[i]=x;

i++;

x=getc(fp);

}

y[i]='\0';

sscanf(y,"%X",&h);

g=f+h;

strcpy(loc,dectohexa(g));

x=getc(fp);

x=getc(fp);

x=getc(fp);

x=getc(fp);

kia:

while(x!='\n')

{

int hh;

sscanf(loc,"%X",&hh);

hh=hh+1;

char temp[20];

strcpy(temp,dectohexa(hh));

int k=search(temp,count);

printf(" 00%s ",loc);

binary(x);

x=fgetc(fp);

binary(x);

printf("\n");

int zz;

sscanf(loc,"%X",&zz);

zz=zz+1;

strcpy(loc,dectohexa(zz));

i=0;

x=fgetc(fp);

while(check(x)!=0)

{

z[i]=x;

x=fgetc(fp);

i++;

}

z[i]='\0';

int pp=strlen(z);

if(k==1)

{

int ee,g;

sscanf(z,"%X",&ee);

g=ee+f;

strcpy(z,dectohexa(g));

}

for(i=0;i<pp;i++)

{

printf(" 00%s ",loc);

binary(z[i]);

i=i+1;

binary(z[i]);

printf("\n");

int zz;

sscanf(loc,"%X",&zz);

zz=zz+1;

strcpy(loc,dectohexa(zz));

}

if(x=='\n')

goto kia;

x=fgetc(fp);

printf("\n");

}

}

else if(x=='E')

{

exit(0);

}

x=getc(fp);

}

}

H^PROGA^000000^000063

D^LISTA^000040^ENDA^000054

T^000020^0A^2A01D^77100004^050014

M^000021^05^+LISTB

E^000000

H^PROGB^000000^00007F

D^LISTB^000060^ENDB^000070

T^000020^0A^2A01D^77100004^050014

M^000021^05^+LISTB

E^000000

H^PROGC^000000^000051

D^LISTC^000030^ENDC^000042

T^000020^0A^2A01D^77100004^050014

M^000021^05^+LISTB

E^000000

enter the no of object programs 3

enter the file name a.c

b.c

c.c

enter the relocation address 4000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

programname symbolname address length

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROGA 4000 0063

LISTA 4040

ENDA 4054

PROGB 4063 007F

LISTB 40C3

ENDB 40D3

PROGC 40E2 0051

LISTC 4112

ENDC 4124

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DEFTAB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LISTA 4040

ENDA 4054

LISTB 40C3

ENDB 40D3

LISTC 4112

ENDC 4124

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<math.h>

char yy[10][100];

char xx[10][100];

char \*get(char ll[100])

{

FILE \*fp2;

char u[100],v[100];

fp2=fopen("def.c","r");

while(!feof(fp2))

{

fscanf(fp2,"%s %s ",u,v);

if(strcmp(ll,u)==0)

{

int i=strlen(v);

int k=0,j=0;

char \*he = (char\*)malloc(sizeof(char)\*100);

for (j = 0 ;j<i;j++)

{

he[k++]=v[j];

}

he[k]='\0';

return he;

}

}

fclose(fp2);

}

void binary(char x)

{

if((int)x==48)

{

printf("0000 ");

}

else if((int)x==49)

{

printf("0001 ");

}

else

{

int c[10],e[10];

int r,l,a,i=0,count=0;

a=(int)x;

if(a>57)

a=a-55;

else

a=a-48;

if(a<4)

printf("00");

else if(a>=4 && a<8)

printf("0");

while(a!=0)

{

c[i] = a%2;

a = a / 2;

count++;

i++;

}

for(i=count-1; i>=0; i--)

{

printf("%d", c[i]);

}

printf(" ");

}

}

char \*dectohexa(int quotient)

{

int temp,i=1,j;

char h[100];

while(quotient!=0){

temp = quotient % 16;

if( temp < 10)

temp =temp + 48;

else

temp = temp + 55;

h[i++]= temp;

quotient = quotient / 16;

}

int k=0;

char \*he = (char\*)malloc(sizeof(char)\*100);

for (j = i -1 ;j> 0;j--)

he[k++]=h[j];

he[k]='\0';

return he;

}

int search(char a[],int n)

{

int i;

if(n<0)

return 99;

for(i=0;i<=n;i++)

{

if(strcmp(a,yy[i])==0)

{

return i;

}

}

return 99;

}

int check(char x)

{

if(x=='\n' || x=='^')

return 0;

else

return 1;

}

main()

{

FILE \*fp;

fp=fopen("p.c","r");

char x,y[100],z[100],s[100],loc[100],n[100],l[100],start[100],name[100],length[100];

int g,h,f;

printf("\n enter the relocation address :");

scanf("%s",length);

sscanf(length,"%X",&f);

x=fgetc(fp);

int k=0,i=0,j;

while(x!=EOF)

{

if(x=='M')

{

x=fgetc(fp);

x=fgetc(fp);

j=0;

while(x!='^')

{

yy[i][j]=x;

x=fgetc(fp);

j++;

}

yy[i][j]='\0';

sscanf(yy[i],"%X",&h);

g=f+h;

strcpy(yy[i],dectohexa(g));

x=fgetc(fp);

x=fgetc(fp);

x=fgetc(fp);

x=fgetc(fp);

char g,ki[100];

g=x;

x=fgetc(fp);

k=0;

while(x!='\n')

{

ki[k]=x;

x=fgetc(fp);

k++;

}

ki[k]='\0';

int u=search(yy[i],i-1);

if(u==99)

{

strcpy(xx[i],get(ki));

i++;

}

else

{

char qq[100],rr[100];

int ee,ff,mm;

strcpy(qq,get(ki));

strcpy(rr,xx[u]);

sscanf(qq,"%X",&ee);

sscanf(rr,"%X",&ff);

if(g=='-')

{

mm=ff-ee;

}

else

{

mm=ff+ee;

}

strcpy(xx[u],dectohexa(mm));

}

}

else

{

while(x!='\n')

{

x=fgetc(fp);

}

}

x=fgetc(fp);

}

rewind(fp);

for(j=0;j!=i;j++)

{

// printf("\n %s :%s\n",yy[j],xx[j]);

}

int count=i;

printf("\n \n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\tPASS2 OF LINKING LOADER\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n ");

x=fgetc(fp);

while(x!=EOF)

{

if(x=='H')

{

x=fgetc(fp);

while(x!='\n')

{

x=fgetc(fp);

}

}

else if(x=='M'|| x=='R'||x=='D')

{

x=fgetc(fp);

while(x!='\n')

{

x=fgetc(fp);

}

}

else if(x=='T')

{

i=0;

x=getc(fp);

x=getc(fp);

while(x!='^')

{

y[i]=x;

i++;

x=getc(fp);

}

y[i]='\0';

sscanf(y,"%X",&h);

g=f+h;

strcpy(loc,dectohexa(g));

x=getc(fp);

x=getc(fp);

x=getc(fp);

x=getc(fp);

kia:

while(x!='\n')

{

int hh;

sscanf(loc,"%X",&hh);

hh=hh+1;

char temp[20];

strcpy(temp,dectohexa(hh));

int k=search(temp,count);

printf(" 00%s ",loc);

binary(x);

x=fgetc(fp);

binary(x);

printf("\n");

int zz;

sscanf(loc,"%X",&zz);

zz=zz+1;

strcpy(loc,dectohexa(zz));

i=0;

x=fgetc(fp);

while(check(x)!=0)

{

z[i]=x;

x=fgetc(fp);

i++;

}

z[i]='\0';

int pp=strlen(z);

if(k==1)

{

int ee,g;

sscanf(z,"%X",&ee);

g=ee+f;

strcpy(z,dectohexa(g));

}

for(i=0;i<pp;i++)

{

printf(" 00%s ",loc);

binary(z[i]);

i=i+1;

binary(z[i]);

printf("\n");

int zz;

sscanf(loc,"%X",&zz);

zz=zz+1;

strcpy(loc,dectohexa(zz));

}

if(x=='\n')

goto kia;

x=fgetc(fp);

printf("\n");

}

}

else if(x=='E')

{

exit(0);

}

x=getc(fp);

}

}

H^PROGA^000000^000063

D^LISTA^000040^ENDA^000054

R^LISTB^ENDB^LISTC^ENDC

T^000020^0A^03201D^77100004^050014

T^00054^0F^000014^FFFFF6^00003F^000014^FFFE22

M^000024^05^+LISTB

M^000054^06^+LISTC

M^000057^06^+ENDC

M^000057^06^-LISTC

M^00005A^06^+ENDC

M^00005A^06^-LISTC

M^00005D^06^+ENDB

M^00005A^06^+PROGA

M^00005D^06^-LISTB

M^000060^06^+LISTC

M^000060^06^-PROGA

E^000020

enter the relocation address :4000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PASS2 OF LINKING LOADER

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

004020 0000 0011

004021 0010 0000

004022 0001 1101

004023 0111 0111

004024 0001 0000

004025 0000 0000

004026 0000 0100

004027 0000 0101

004028 0000 0000

004029 0001 0100

004054 0000 0000

004055 0000 0000

004056 0001 0100

004057 1111 1111

004058 1111 1111

004059 1111 0110

00405A 0000 0000

00405B 0000 0000

00405C 0011 1111

00405D 0000 0000

00405E 0000 0000

00405F 0001 0100

004060 1111 1111

004061 1111 1110

004062 0010 0010