**EXP NO :3 SIMULATION OF STATIC AND DYNAMIC LINKERS**

**AIM:**

To write a C program to implementation for the simulation of static and dynamic linkers.

**ALGORITHM:**

1.program\_linked\_origin:=<link origin>from linker command;

2.For each object module

a) t\_origin:=translation origin of the object module;

om\_size:=size of the object module ;

b)relocation\_factor:=program\_linked\_origin -t\_origin;

c)read the machine language program in work\_area.

d)read LINKTAB of the object module.

e)For each LINKTAB entry with type =PD

name:=symbol;

linked\_address:=translation\_address+relocation factor;

Enter(name,linked\_address)in NTAB;

f)Enter(object module name,program\_linked\_origin) in DTAB;

g)program\_linked\_origin:=program\_linked\_origin+om\_size;

3.For each object module

a)t\_origin:=translation origin of object module;

Program\_linked\_origin:=load address from NTAB;

b)For each LINKTAB entry with type=EXT;

i)address\_in\_work area:=address of work area + program\_linked\_origin-<link origin> + translation origin - t\_origin;

ii)Search symbol in NTAB and COPY its linked address add the linked address to operand address in the word with the address address\_in\_work\_area.

**PROGRAM:**

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

int checkaddr(char var[],int arr[],int n)

{

int m=atoi(var);

int i,flag=0;

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

{

if(m==arr[i])

{

return 1;

}

}

return 0;

}

int main()

{

char str[60],var[10],ext1[10];

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

f1=fopen("C:/Users/Vijay Robinson/Desktop/linker input 1.txt","r");

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

int origin1,origin2,size1=16,s,line=0,i,result,ext,ext2;

fclose(f1);

origin1=atoi(var);

char var1[4][10];

f2=fopen("C:/Users/Vijay Robinson/Desktop/linker input2.txt","r");

fscanf(f2,"%s",var);

origin2=atoi(var);

int tar\_addr=900;

int relocation\_factor1,relocation\_factor2,result\_origin=900;

relocation\_factor1=result\_origin-origin1;

relocation\_factor2=(result\_origin+size1)-origin2;

printf("MODULE P \nORIGIN:%d",origin1);

printf("\nRELOCATION FACTOR: %d\nTARGET ORIGIN: %d\n",relocation\_factor1,result\_origin);

printf("\nMODULE Q\nORIGIN:%d\nTARGET ORIGIN: %d\n",origin2,result\_origin+size1);

printf("RELOCATION FACTOR: %d\n",relocation\_factor2);

FILE \*f6;

f6=fopen("C:/Users/Vijay Robinson/Desktop/ntab.txt","w"); // FOR NTAB CREATION

f4=fopen("C:/Users/Vijay Robinson/Desktop/linktab1.txt","r");

fprintf(f6,"P\t%d\n",result\_origin);

while(!feof(f4))

{

fgets(str,40,f4);

strtok(str,"\n");

char link[3][10];

int y=0;

char \*ptr=strtok(str,"\t");

while(ptr!=NULL)

{

strcpy(link[y],ptr);

ptr=strtok(NULL,"\t");

y++;

}

if(strcmp(link[1],"PD")==0)

{

int temp;

temp=atoi(link[2]);

temp=temp+relocation\_factor1;

fprintf(f6,"%s\t%d\n",link[0],temp);

}

if(strcmp(link[1],"EXT")==0)

{

ext=atoi(link[2]);

//ext=ext+relocation\_factor1;

strcpy(ext1,link[0]);

}

i=0;

}

fclose(f6);

fclose(f4);

f6=fopen("C:/Users/Vijay Robinson/Desktop/ntab.txt","a");

f4=fopen("C:/Users/Vijay Robinson/Desktop/linktab2 (1).txt","r");

fprintf(f6,"Q\t%d\n",origin2+relocation\_factor2);

while(!feof(f4))

{

fgets(str,40,f4);

strtok(str,"\n");

char link[3][10];

int y=0;

char \*ptr=strtok(str,"\t");

while(ptr!=NULL)

{

strcpy(link[y],ptr);

ptr=strtok(NULL,"\t");

y++;

}

if(strcmp(link[1],"PD")==0)

{

fprintf(f6,"%s\t%d\n",link[0],(atoi(link[2])+relocation\_factor2));

}

i=0;

}

fclose(f4);

fclose(f6);

f4=fopen("C:/Users/Vijay Robinson/Desktop/linktab2 (1).txt","r"); // PROCESSING EXTERN

while(!feof(f4))

{

fgets(str,40,f4);

strtok(str,"\n");

char link[3][10];

int y=0;

char \*ptr=strtok(str,"\t");

while(ptr!=NULL)

{

strcpy(link[y],ptr);

ptr=strtok(NULL,"\t");

y++;

}

if(strcmp(link[1],"PD")==0&&strcmp(ext1,link[0])==0)

{

ext2=atoi(link[2]);

ext2=ext2+relocation\_factor2;

}

i=0;

}

fclose(f4);

f1=fopen("C:/Users/Vijay Robinson/Desktop/relocat\_linker-input1.txt","r"); // RELOCATION OF MODULE1 P

while(!feof(f1))

{

fgets(str,40,f1);

line++;

}

fclose(f1);

int arr1[line];

char str1[30];

f1=fopen("C:/Users/Vijay Robinson/Desktop/relocat\_linker-input1.txt","r");

while(!feof(f1))

{

fgets(str1,40,f1);

strtok(str1,"\n");

char \*ptr=strtok(str1," ");

while(ptr!=NULL)

{

static int m=0;

arr1[m]=atoi(str1);

ptr=strtok(NULL," ");

m++;

s=m;

}

}

fclose(f1);

f3=fopen("C:/Users/Vijay Robinson/Desktop/linker input 1.txt","r");

f5=fopen("C:/Users/Vijay Robinson/Desktop/linker\_output.txt","w");

char str2[60];

i=0;

while(!feof(f3))

{

fgets(str2,40,f3);

strtok(str2,"\n");

char \*ptr=strtok(str2," ");

while(ptr!=NULL)

{

strcpy(var1[i],ptr);

i++;

ptr=strtok(NULL," ");

}

result=checkaddr(var1[0],arr1,s);

if(result==1)

{

int x,c;

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

{

c=atoi(var1[x]);

if(x==0||x==3)

{

c=c+relocation\_factor1;

}

fprintf(f5,"%d ",c);

}

fprintf(f5,"\n");

}

if(result==0)

{

int x,c;

int flag=0;

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

{

c=atoi(var1[x]);

if(x==0)

{

if(c==ext) // CHECKING WHETHER THE ADDRESS IS EQUAL TO THE EXTERN ADDRESS IN LINKTAB

{

flag=1;

}

c=c+relocation\_factor1;

}

if(flag==1&&x==3)

{

c=ext2; // MAKING THE EXTERN TO POINT TO THE ORIGINAL MODOULE ADDRESS

}

fprintf(f5,"%d ",c);

c=0;

}

fprintf(f5,"\n");

flag=0;

}

i=0;

}

fclose(f5);

f2=fopen("C:/Users/Vijay Robinson/Desktop/relocat\_linker-input2.txt","r"); // RELOCATION OF MODULE2 Q

f4=fopen("C:/Users/Vijay Robinson/Desktop/linker input2.txt","r");

f5=fopen("C:/Users/Vijay Robinson/Desktop/linker\_output.txt","a");

line=0;

i=0;

while(!feof(f2))

{

fgets(str,40,f2);

line++;

}

fclose(f2);

f2=fopen("C:/Users/Vijay Robinson/Desktop/relocat\_linker-input2.txt","r");

arr1[line];

while(!feof(f2))

{

fgets(str1,40,f2);

strtok(str1,"\n");

char \*ptr=strtok(str1," ");

while(ptr!=NULL)

{

static int m=0;

arr1[m]=atoi(str1);

ptr=strtok(NULL," ");

m++;

s=m;

}

}

fclose(f2);

while(!feof(f4))

{

fgets(str2,40,f4);

strtok(str2,"\n");

char \*ptr=strtok(str2," ");

while(ptr!=NULL)

{

strcpy(var1[i],ptr);

i++;

ptr=strtok(NULL," ");

}

result=checkaddr(var1[0],arr1,s);

if(result==1)

{

int x,c;

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

{

c=atoi(var1[x]);

if(x==0||x==3)

{

c=c+relocation\_factor2;

}

fprintf(f5,"%d ",c);

}

fprintf(f5,"\n");

}

if(result==0)

{

int x,c;

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

{

c=atoi(var1[x]);

if(x==0)

{

c=c+relocation\_factor2;

}

fprintf(f5,"%d ",c);

}

fprintf(f5,"\n");

}

i=0;

}

fclose(f5);

printf("\nEXTERN:\nLINKTAB ADDRESS: %d\nTARGET ADDRESS: %d\nNAME OF THE EXTERN: %s\n",ext,ext2,ext1);

return 0;

}

**INPUT:**

**relocat\_linker\_input1.txt:**

601

602

603

604

606

607

608

609

610

611

**relocat\_linker\_input2.txt:**

200

201

202

203

204

**linker\_input1.txt:**

601 9 0 613

602 4 2 615

603 5 2 616

604 3 2 616

605 4 3 616

606 1 3 615

607 5 3 616

608 6 3 613

609 7 2 604

610 5 2 614

611 10 0 614

612 0 0 0

613 11 1 111

614 11 1 111

615 0 0 111

616 11 1 111

**linker\_input2txt:**

200 09 0 206

201 04 1 208

202 01 1 206

203 05 1 207

204 10 0 207

205 00 0 000

206 11 1 111

207 11 1 111

208 00 0 025

**linktab1txt:**

RESULT PD 614

ALPHA EXT 605

**linktab2(1).txt:**

ALPHA PD 208

**OUTPUT :**

**linker\_output.txt:**

900 9 0 912

901 4 2 914

902 5 2 915

903 3 2 915

904 4 3 924

905 1 3 914

906 5 3 915

907 6 3 912

908 7 2 903

909 5 2 913

910 10 0 913

911 0 0 0

912 11 1 111

913 11 1 111

914 0 0 10

915 11 1 111

916 9 0 922

917 4 1 924

918 1 1 922

919 5 1 923

920 10 0 923

921 0 0 0

922 11 1 111

923 11 1 111

924 0 0 25

**RESULT:**

Thus implementation of static and dynamic linkers using c program is verified and executed successfully.