|  |  |
| --- | --- |
| Ex.no.1b | **IMPLEMENTING SYMBOL TABLE USING BINARY FILE** |
| 03.10.20 |

## Head.h

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

typedef struct node{

char sybm[10];

char type[10];

int len;

}node;

int insert();

node search(char[]);

void display(node);

int modify();

int deletee();

void displayall();

## Implementation.c

#include "head.h"

int insert(){

node temp,t;

printf("Enter symbol : ");

scanf("%s",temp.sybm);

printf("Enter type : ");

scanf("%s",temp.type);

printf("Enter length : ");

scanf("%d",&temp.len);

int flag = 0;

FILE \*f1 = fopen("t1.bin", "ab+");

FILE \*f2 = fopen("t2.bin", "ab+");

if((!f1) || (!f2)){

printf("Unable to Read file!Error\n");

return 0;

}

while(fread(&t,sizeof(node),1,f1)!=NULL){

if(strcmp(t.sybm,temp.sybm) > 0 && flag != 1){

fwrite(&temp,sizeof(node),1,f2);

fwrite(&t,sizeof(node),1,f2);

flag=1;

}

else{

fwrite(&t,sizeof(node),1,f2);

}

}

if(flag == 0){

fwrite(&temp,sizeof(node),1,f2);

}

fclose(f1);

fclose(f2);

remove("t1.bin");

rename("t2.bin","t1.bin");

return 1;

}

int deletee(){

printf("Enter a Symbol to delete : ");

char tempSymbol[10];

scanf("%s",tempSymbol);

FILE \*f1 = fopen("t1.bin", "rb");

FILE \*f2 = fopen("t2.bin", "ab+");

node t;

int flag=0;

if((!f1) || (!f2)){

printf("Unable to Read file!Error\n");

return 0;

}

while(fread(&t,sizeof(node),1,f1)!=NULL){

if(strcmp(t.sybm,tempSymbol) == 0){

flag=1;

break;

}

fwrite(&t,sizeof(node),1,f2);

}

if(flag==0){

printf("Record not found\n");

return 0;

}

fclose(f1);

fclose(f2);

remove("t1.bin");

rename("t2.bin", "t1.bin");

return 1;

}

node search(char symbl[]){

node t;

FILE \*f = fopen("t1.bin", "rb");

if(!f){

printf("Unable to Read file!Error\n");

strcpy(t.sybm,"\0");

return t;

}

while(fread(&t,sizeof(node),1,f))

if(strcmp(t.sybm,symbl) == 0){

fclose(f);

return t;

}

fclose(f);

strcpy(t.sybm,"\0");

return t;

}

void display(node t){

if(strcmp(t.sybm,"\0") == 0)

printf("Symbol not found\n");

else

printf("\n%s\t%s\t%d\n",t.sybm,t.type,t.len);

}

void displayall(){

FILE \*f1 = fopen("t1.bin", "rb");

node t;

while(fread(&t,sizeof(node),1,f1)){

printf("%s\t%s\t%X\n",t.sybm,t.type,t.len);

}

fclose(f1);

}

int modify(){

node temp,t;

printf("Enter symbol : ");

scanf("%s",temp.sybm);

printf("Enter type to be modified : ");

scanf("%s",temp.type);

printf("Enter length to be modified : ");

scanf("%d",&temp.len);

int flag=0;

FILE \*f1 = fopen("t1.bin", "rb");

FILE \*f2 = fopen("t2.bin", "ab+");

if((!f1) || (!f2)){

printf("Unable to Read file!Error\n");

return 0;

}

while(fread(&t,sizeof(node),1,f1)!=NULL){

if(strcmp(t.sybm,temp.sybm) == 0)

fwrite(&temp,sizeof(node),1,f2);

else

fwrite(&t,sizeof(node),1,f2);

}

fclose(f1);

fclose(f2);

remove("t1.bin");

rename("t2.bin", "t1.bin");

return 1;

}

## Application.c

#include "head.h"

int main(){

int ch;

printf("\n1-Insert\n2-Search\n3-Modify\n4-Delete\n5-Display\n6- Exit\n");

while(1){

printf("Enter Your Choice : ");

scanf("%d",&ch);

switch(ch){

case 1: if(insert()){

printf("\nSuccessfully inserted the Symbol\n");

}

else{

printf("Insertion failed\n");

}

break;

case 2: printf("Enter a Symbol to Search : ");

char temp[10];

scanf("%s",temp);

display(search(temp));

break;

case 3:

if(modify())

printf("\nModification Successfull\n");

else

printf("\nModification failed\n");

break;

case 4:

if(deletee())

printf("\nSymbol deleted successfully\n");

else

printf("\nFailed to delete...\n");

break;

case 5:

displayall();

break;

case 6:

exit(0);

default:

printf("Enter valid options\n");

}

}}

## Output:

## 

## Result:

The symbol table implementation using binary file was executed successfully for the insert, delete, modify, search and display options.