Brianna Moffett

CS 370

02.07.2020

Lab3 Questions

1. Description: Modify the symbol table, properly comment things, and make sure you understand how the table works.

Modified code:

```
/*
* Modified by: Brianna Moffett
* Changes made:
*   -made header and included it in file
*   -added comments to code
*   -removed headers for functions and placed them in header file
*/
#include<stdio.h>
/* #include<conio.h> */
#include<malloc.h>
#include<string.h>
#include<stdlib.h>
#include "symtabfuncs.h"
int size=0;
    struct SymbTab *first,*last;
    void main()
    {
        int op,y;
        char la[10];
        do
        {
            printf("\n\tSYMBOL TABLE IMPLEMENTATION\n");

printf("\n\t1.INSERT\n\t2.DISPLAY\n\t3.DELETE\n\t4.SEARCH\n\t5.MODIFY\n\t6.EN
D\n");
            printf("\n\tEnter your option : ");
            scanf("%d",&op);
            switch(op)
            {
                case 1:
                    Insert();
                    break;
                case 2:
                    Display();
                    break;
                case 3:
                    Delete();
                    break;
                case 4:
                    printf("\n\tEnter the label to be searched : ");
                    scanf("%s",la);
                    y=Search(la);
                    printf("\n\tSearch Result:");
                    if(y==1)
                        printf("\n\tThe label is present in the symbol
table\n");
                    else
                        printf("\n\tThe label is not present in the symbol
table\n");
                    break;
                case 5:
                    Modify();
                    break;
                case 6:
                    exit(0);
            }
        }while(op<6);

    }  /* and of main */
    void Insert() /*inserts symbol into table*/
    {
        int n;
        char l[10];
        printf("\n\tEnter the label : ");
        scanf("%s",l);
        n=Search(l);
        if(n==1)
            printf("\n\tThe label exists already in the symbol
table\n\tDuplicate can.t be inserted");
        else
        {
            struct SymbTab *p;
            p=malloc(sizeof(struct SymbTab));
            strcpy(p->label,l);
            printf("\n\tEnter the symbol : ");
            scanf("%s",p->symbol);
            printf("\n\tEnter the address : ");
            scanf("%d",&p->addr);
            p->next=NULL;
            if(size==0)
            {
                first=p;
                last=p;
            }
            else
            {
```

```
            last->next=p;
            last=p;
        }
        size++;
    }
    printf("\n\tLabel inserted\n");
}/*of insert*/
void Display() /*displays the table*/
{
    int i;
    struct SymbTab *p;
    p=first;
    printf("\n\tLABEL\t\tSYMBOL\t\tADDRESS\n");
    for(i=0;i<size;i++)
    {
        printf("\t%s\t\t%s\t\t%d\n",p->label,p->symbol,p->addr);
        p=p->next;
    }
}/*of display*/
int Search(char lab[]) /*searches table for specified symbol*/
{
    int i,flag=0;
    struct SymbTab *p;
    p=first;
    for(i=0;i<size;i++)
    {
        if(strcmp(p->label,lab)==0)
            flag=1;
        p=p->next;
    }
    return flag;
}/*of search*/
```

```
void Modify() /*modifies a symbol already inside the table*/
{
    char l[10],nl[10];
    int add,choice,i,s;
    struct SymbTab *p;
    p=first;
    printf("\n\tWhat do you want to modify?\n");
    printf("\n\t1.Only the label\n\t2.Only the address\n\t3.Both the label and address\n");
    printf("\tEnter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            printf("\n\tEnter the old label : ");
            scanf("%s",l);
            s=Search(l);
            if(s==0)
                printf("\n\tLabel not found\n");
            else
            {
                printf("\n\tEnter the new label : ");
                scanf("%s",nl);
                for(i=0;i<size;i++)
                {
                    if(strcmp(p->label,l)==0)
                        strcpy(p->label,nl);
                    p=p->next;
                }
                printf("\n\tAfter Modification:\n");
                Display();
            }
```

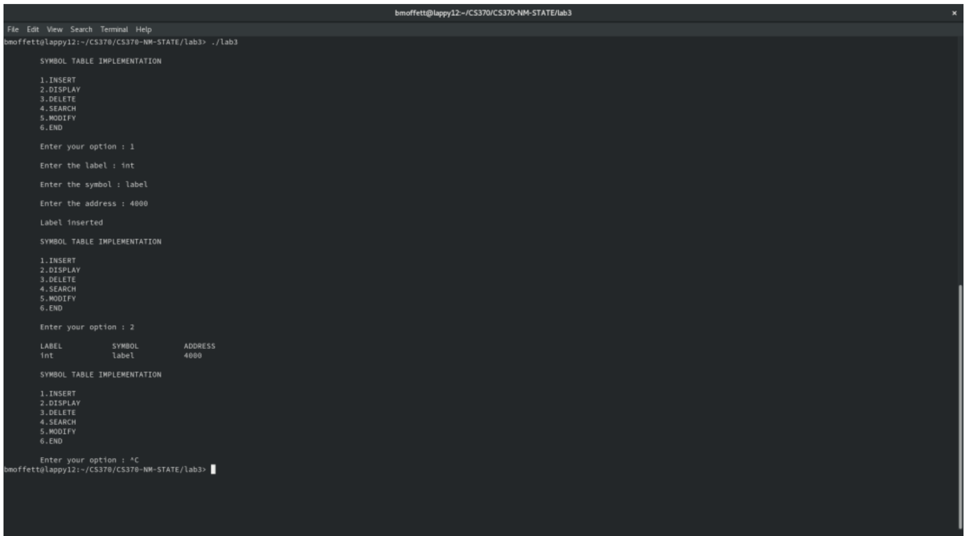
```
                break;
            case 2:
                printf("\n\tEnter the label where the address is to be
modified : ");
                scanf("%s",l);
                s=Search(l);
                if(s==0)
                    printf("\n\tLabel not found\n");
                else
                {
                    printf("\n\tEnter the new address : ");
                    scanf("%d",&add);
                    for(i=0;i<size;i++)
                    {
                        if(strcmp(p->label,l)==0)
                            p->addr=add;
                        p=p->next;
                    }
                    printf("\n\tAfter Modification:\n");
                    Display();
                }
                break;
            case 3:
                printf("\n\tEnter the old label : ");
                scanf("%s",l);
                s=Search(l);
                if(s==0)
                    printf("\n\tLabel not found\n");
                else
                {
                    printf("\n\tEnter the new label : ");
                    scanf("%s",nl);
                    printf("\n\tEnter the new address : ");
                    scanf("%d",&add);
                    for(i=0;i<size;i++)
                    {
                        if(strcmp(p->label,l)==0)
                        {
                            strcpy(p->label,nl);
                            p->addr=add;
                        }
                        p=p->next;
                    }
                    printf("\n\tAfter Modification:\n");
                    Display();
                }
                break;
        }
    } /*of modify*/
    void Delete() /*removes a symbol from the table*/
    {
        int a;
        char l[10];
        struct SymbTab *p,*q;
        p=first;
        printf("\n\tEnter the label to be deleted : ");
        scanf("%s",l);
        a=Search(l);
        if(a==0)
            printf("\n\tLabel not found\n");
        else
        {
            if(strcmp(first->label,l)==0)
                first=first->next;
            else if(strcmp(last->label,l)==0)
            {
                q=p->next;
                while(strcmp(q->label,l)!=0)
                {
                    p=p->next;
                    q=q->next;
                }
                p->next=NULL;
                last=p;
            }
            else
            {
                q=p->next;
                while(strcmp(q->label,l)!=0)
                {
                    p=p->next;
                    q=q->next;
                }
                p->next=q->next;
            }
            size--;
            printf("\n\tAfter Deletion:\n");
            Display();
        }
    } /*of delete*/
```

Header file:

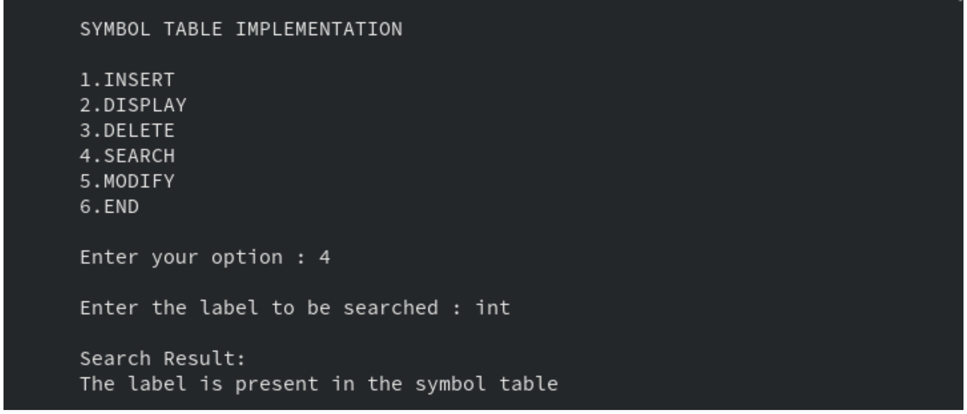
```
  #ifndef SYMTABFUNCS_H
  #define SYMTABFUNCS_H
  #include<stdio.h>
  /* #include<conio.h> */
  #include<malloc.h>
  #include<string.h>
  #include<stdlib.h>
```

```
/*
* Modifed by: Brianna Moffett
* Changes made:
*  -created header file
*/
void Insert();
void Display();
void Delete();
int Search(char lab[]);
void Modify();
struct SymbTab
{
    char label[10],symbol[10];
    int addr;
    struct SymbTab *next;};
#endif
```

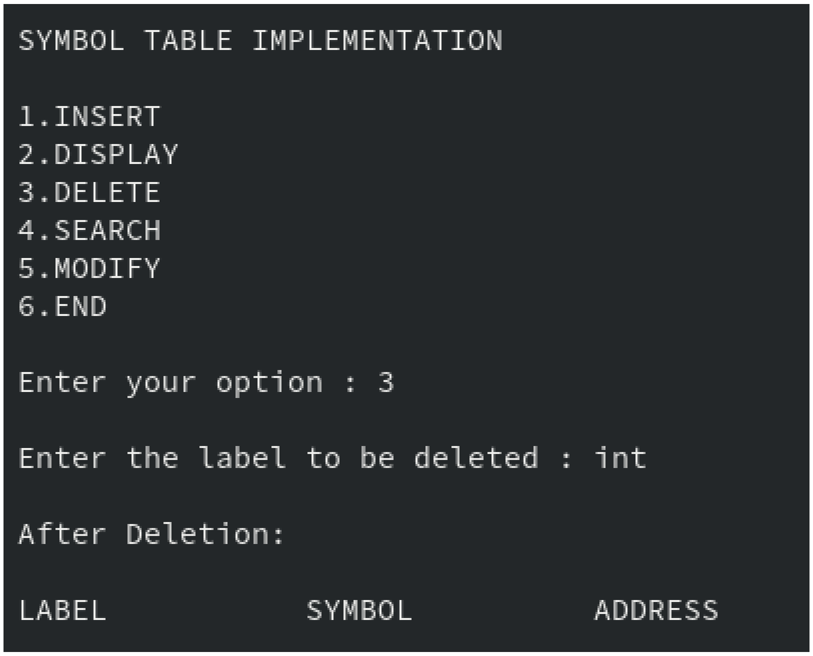
2. Main Structure: symbol table
    - Built using a linked list
    - Fields:
        - -label
        - -symbol
        - -address (this is where the symbol is stored in memory)
3. Actions:


bmoffett@lappy12:~/CS370/CS370-NM-STATE/lab3
File Edit View Search Terminal Help

```
bmoffett@lappy12:~/CS370/CS370-NM-STATE/lab3> ./lab3

        SYMBOL TABLE IMPLEMENTATION

        1.INSERT
        2.DISPLAY
        3.DELETE
        4.SEARCH
        5.MODIFY
        6.END

        Enter your option : 1

        Enter the label : int

        Enter the symbol : label

        Enter the address : 4000

        Label inserted

        SYMBOL TABLE IMPLEMENTATION

        1.INSERT
        2.DISPLAY
        3.DELETE
        4.SEARCH
        5.MODIFY
        6.END

        Enter your option : 2

        LABEL           SYMBOL          ADDRESS
        int             label           4000

        SYMBOL TABLE IMPLEMENTATION

        1.INSERT
        2.DISPLAY
        3.DELETE
        4.SEARCH
        5.MODIFY
        6.END

        Enter your option : ^C
bmoffett@lappy12:~/CS370/CS370-NM-STATE/lab3> 
```



```
        SYMBOL TABLE IMPLEMENTATION

        1.INSERT
        2.DISPLAY
        3.DELETE
        4.SEARCH
        5.MODIFY
        6.END

        Enter your option : 4

        Enter the label to be searched : int

        Search Result:
        The label is present in the symbol table
```



```
SYMBOL TABLE IMPLEMENTATION

1.INSERT
2.DISPLAY
3.DELETE
4.SEARCH
5.MODIFY
6.END

Enter your option : 3

Enter the label to be deleted : int

After Deletion:

LABEL           SYMBOL          ADDRESS
```