

Address Book

Address Book

The online address for all of the code:

Readme to program:

Background of this program:

In this system, I realize some functions:

Special Features :

File organization :

The function design:

main.c:

dataIO.c:

displayALLcontacts.c:

SearchAndModify.c:

ContactAdd.c:

MenuDisplay.c:

usefulTools.c:

Running Picture :

Mac or Linux:

Windows:

The online address for all of the code:

the source code in the github:

`https://github.com/ricocoYan/Homework-AddressBook`

Readme to program:

The application is developed in linux .

```
root@ubuntu: /home/rico-yang/Documents/clion-2020.1.1/bin
File Edit View Search Terminal Help
contactAdd.o: In function `getEmail':
contactAdd.c:(.text+0x1e8): warning: the `gets' function is dangerous and s
hould not be used.
root@ubuntu:/home/rico-yang/CHOME# ./main

*****
*                                     *
*      Welcome to use Address Book   *
*                                     *
*                                     *
*      1. Display all contacts        *
*                                     *
*      2. Add contact                 *
*                                     *
*      3. Search & modify contact     *
*                                     *
*      4. Exit                        *
*                                     *
*                                     *
*****

Please input your choice: 1
The address book is empty.
Press any key to the menu...
*****
*                                     *
*      Welcome to use Address Book   *
```

In order to compile this program, some instructions should be used in terminal.

For Linux:

```
cd "target"
/bin/bash shell
make
./main
```

For Mac:

Download the Clion opening to see

For windows:

Download the Clion opening to see

Background of this program:

Xiao Zhao has to manage his own address book. Each address book includes data such as name, contact number, email and work unit. Where, the name and contact number cannot be blank. Xiaozhao needs to browse, search, add and delete the address book.

In this system, I realize some functions:

Separation of user menu and logic function is required

Write modules such as menu selection, address book view, search, add and delete respectively, and call corresponding functions of different modules according to the user's selection in the main calling function

Use files to save address book information, one line one record

In memory, the address book data read from the file is stored using the structure array or single chain table structure

According to the idea of engineering, the program is organized in the form of multiple documents

Use parameters to transfer data. Try not to use global variables

Special Features :

Using the data structure `linklist` to modify, add, and delete the records.

sort the records by `ASCII` code of their names in ascending order.

using the `json object` to storage their information by introducing the third-part library Cjson.

using the `system call` for the program to elevate the customers' satisfaction.

Add some colors into the set-control under the `windows operating system`.

Handle the three common operating system : mac、Linux、and Windows.

File organization :

`header.h`:The header file, including function prototypes, structure and global variables declarations.

`CJson.h`:The third-part JSON parser library.

`global.h`:The global variables declarations.

`CJson.c`:The third-part JSON parser library.

`main.c`: The entrance of the program

`dataIO.c`: The file operations, including parsing JSON format and the I/O of the data file.

`displayAllContacts.c`: To display all contacts. `searchAndModify.c`: To search for the specified entry(with keyword), and modify or delete the entry.

`contactAdd.c`: To add new entry into the address book. `menuDisplay.c`: To display the main menu and get into performance functions. `usefulTools.c`: Some useful functions which could be used everywhere.

The function design:

Only list some core function code and explain its in detail.

main.c:

The entrance of the program

dataIO.c:

readdataFP:

description:

Read the origin content from the data file and construct a cJSON object for the next operation.

Return Value:

If the file is successfully opened, return a cJSON object. Else, return a NULL pointer.

writedataFP:

description:

Save the in-memory data into the data file in the format of JSON .

parseJson:

description:

parse the JSON format data and construct the linked list.

displayALLcontacts.c:

displayContact

description:

Display the information of specified entry target .

Code:

```
void displayContact(detail * target) {
    printf("Name: %s\n", target->name);
    printf("Phone Number: %s\n", target->phoneNumber);
    if(target->email[0] != '\n') {
        printf("E-mail: %s\n", target->email);
    }
    if(target->address[0] != '\n') {
        printf("Address: %s\n", target->address);
    }
    if(target->workPlace[0] != '\n') {
        printf("Work Place: %s\n", target->workPlace);
    }
    printf("\n");
}
```

```
}
```

input: target

output: N/A

displayALLContact

description:

Display ALL the entries in the address book.

```
//reverse the linklist to display all the entry
void displayAllContacts() {
    detail * iteration;
    if(head == head->next) {

        printf("The address book is empty.\n");

        backMenu();
        return;
    }

    for(iteration = head; ; iteration = iteration->next) {
        if(iteration == iteration->next)
            break;
        displayContact(iteration);
    }

    printf("\n");
    backMenu();
}
```

SearchAndModify.c:

searchContact:

description:

search the target with key words.

code:

```
//search the target with key words
detail ** searchContact() {
    detail * iteration, * firstOne;

    detail ** result = (detail **)malloc(sizeof(detail *) * 10);
    int flag = 0;
    if(head->next == head) {
```

```

        printf("The address book is empty.\n");

        backMenu();
        return NULL;
    }

    while(1) {
        total = 0;
        char target[100];
        //get the target
        printf("Please input the keyword: ");
        scanf("%s", target);

        for(iteration = head; ; iteration = iteration->next) {
            if(strstr(iteration->name, target) != NULL || strstr(iteration->phoneNumber, target) != NULL || strstr(iteration->email, target) != NULL || strstr(iteration->address, target) != NULL || strstr(iteration->workPlace, target) != NULL) {
                result[total++] = iteration;
                printf("[%d]\n", total);
                displayContact(iteration);
                if(flag == 0) {
                    flag = 1;
                    firstOne = iteration;
                }
            }
            if(iteration == iteration->next)
                break;
        }
        if(total > 0) {
            printf("%d result(s) found.\n", total);
            modifyContact(result);
            return result;
        }
        else {

            printf("No result. Try again?(y/n)");
            char choice = getCharChoice("ny");
            if(choice == 0) {

                backMenu();
                return NULL;
            }
        }
    }
}

```

ModifyContact:

description:

modify or delete the entry

deleteContact:

description:

delete the entry

ContactAdd.c:

newNode:

description:

Add a new node into the linked list.

code:

```
detail * newNode() {
    detail * newNode = (detail *)malloc(sizeof(detail));
    newNode->next = newNode;
    return newNode;
}
```

insertNode:

description:

Insert the created node to the appropriate position, to make sure all entries in order.

code:

```
//insert node
//比较
detail * insertNode(detail * node) {
    detail * iteration;
    for(iteration = head; ; iteration = iteration->next) {
        detail * temp;
        if(strcmp(node->name, head->name) <= 0) {

            temp = head;
            head = node;
            head->next = temp;
            return node;
        } //Head situation

        if(iteration->next == iteration->next->next) {
            temp = iteration->next;
            iteration->next = node;
            node->next = temp;
        }
    }
}
```

```

        return node;

    } //Tail situation
    if(strcmp(node->name, iteration->name) >= 0 && strcmp(node->name,
iteration->next->name) <= 0) {
        temp = iteration->next;
        iteration->next = node;
        node->next = temp;
        return node;
    } //Common

}
return NULL;
}

```

ContactAdd:

description:

create a new entry and add information.

getPhoneNumber:

description:

get the current phone number

code:

```

void getPhoneNumber(detail * newNode) {
phone:
    fflush(stdin);
    SetConsoleTextAttribute(hConsole, 0x0a);
    printf("Phone Number: ");
    SetConsoleTextAttribute(hConsole, 0x0f);
    scanf("%[^\\n]", newNode->phoneNumber);
    for(int i = 0; i < strlen(newNode->phoneNumber); i++) {
        if(!(newNode->phoneNumber[i] >= '0' && newNode->phoneNumber[i] <= '9'
|| newNode->phoneNumber[i] == '-')) {
            SetConsoleTextAttribute(hConsole, 0x04);
            printf("The phone number you input is illegal. Try again.\\n");
            goto phone;
        }
    }
    SetConsoleTextAttribute(hConsole, 0x0a);
}

```

getName:

description:

get the current Name

getEmail:

description:

get the current Email

getAdress:

description:

get the current Adress

getworkplace:

description:

get the current workplace

MenuDisplay.c:

menuDisplay:

description:

Display the menu and let the user choose to step in which function.

code:

```
#include "header.h"
#include <unistd.h>

int menuDisplay() {
    while(1) {
        sleep(0.5);
        printf("\n");
        printf("*****\n");
        printf(" *                               *\n");
        printf(" *   Welcome to use Address Book   *\n");
        printf(" *                               *\n");
        printf(" *                               *\n");
        printf(" *                               *\n");
        printf(" *   1. Display all contacts        *\n");
        printf(" *                               *\n");
        printf(" *   2. Add contact                *\n");
        printf(" *                               *\n");
        printf(" *   3. Search & modify contact    *\n");
        printf(" *                               *\n");
        printf(" *   4. Exit                      *\n");
        printf(" *                               *\n");
        printf(" *                               *\n");
        printf(" *                               *\n");
        printf("*****\n");
        printf("\n");
    }
}
```

```

        printf("Please input your choice: ");

        int choice;
        choice = getIntChoice(1, 4);
        if(choice == -1) {

            fflush(stdin);
            continue;
        }
        return choice;
    }
}

```

usefulTools.c:

getIntChoice:

description:

goto the correct procedure.

backMenu:

description:

go back to the main menu

getcharChoice:

description:

Get the user's character input and return the corresponding integer.

Running Picture :

Mac or Linux:

```

"/Users/rico-yang/Desktop/Homework AddressBook/cmake-build-
debug/Homework_AddressBook"

```

```

*****
*                                     *
*      Welcome to use Address Book   *
*                                     *
*                                     *
*                                     *
*      1. Display all contacts        *
*                                     *
*      2. Add contact                *
*                                     *

```

```
*      3. Search & modify contact      *
*
*      4. Exit                          *
*
*
*
*
*****
```

Please input your choice: 2

Please input the information:

Name: yang

Phone Number: 13811720972

E-mail(optional): warning: this program uses gets(), which is unsafe.
839891341@qq.com

Address(optional): ShenZhen

Work Place(optional): ShenZhen

Do you want to add a new one(y/n)?

Please input your choice: n

Press any key to the menu...

```
*****
*
*      Welcome to use Address Book      *
*
*
*
*
*      1. Display all contacts          *
*
*      2. Add contact                  *
*
*      3. Search & modify contact      *
*
*      4. Exit                        *
*
*
*
*****
```

Please input your choice: 1

Name: yang

Phone Number: 13811720972

E-mail: 839891341@qq.com

Address: ShenZhen

Work Place: ShenZhen

Name: zhang

Phone Number: 13811720972

E-mail: 839891341@qq.com

Address: Shenzhen

Work Place: Shenzhen

Name: yang
Phone Number: 13811720972
E-mail: 839891341@qq.com
Address: Shenzhen
Work Place: Shenzhen

Name: zhang
Phone Number: 13811720972
E-mail: 839891341@qq.com
Address: Shenzhen
Work Place: Shenzhen

Press any key to the menu...

```
*****
*                                     *
*      Welcome to use Address Book   *
*                                     *
*                                     *
*                                     *
*                                     *
*      1. Display all contacts        *
*                                     *
*      2. Add contact                 *
*                                     *
*      3. Search & modify contact     *
*                                     *
*      4. Exit                        *
*                                     *
*                                     *
*                                     *
*****
```

Please input your choice: 3
Please input the keyword: yang

[1]
Name: yang
Phone Number: 13811720972
E-mail: 839891341@qq.com
Address: ShenZhen
Work Place: ShenZhen

[2]
Name: yang
Phone Number: 13811720972
E-mail: 839891341@qq.com
Address: Shenzhen
Work Place: Shenzhen

```
2 result(s) found.
Do you want to modify?(y/n)1
Your input is illegal. Try again.
y
Please input the number you want to modify: 1
Which field do you want to modify?(d for delete this item, n for name, p for
phone number,
e for email, a for address, w for workplace) n
Name: zhang
Modified successful!
Press any key to the menu...

*****
*                                     *
*   Welcome to use Address Book     *
*                                     *
*                                     *
*                                     *
*                                     *
*   1. Display all contacts          *
*                                     *
*   2. Add contact                   *
*                                     *
*   3. Search & modify contact       *
*                                     *
*   4. Exit                          *
*                                     *
*                                     *
*                                     *
*****

Please input your choice: 4
```

Windows:

```
*****
*                                     *
*   Welcome to use Address Book     *
*                                     *
*                                     *
*   1. Display all contacts          *
*                                     *
*   2. Add contact                   *
*                                     *
*   3. Search & modify contact       *
*                                     *
*   4. Exit                          *
*                                     *
*                                     *
*                                     *
*****
```