Student Information Management

Project Introduction

Contents

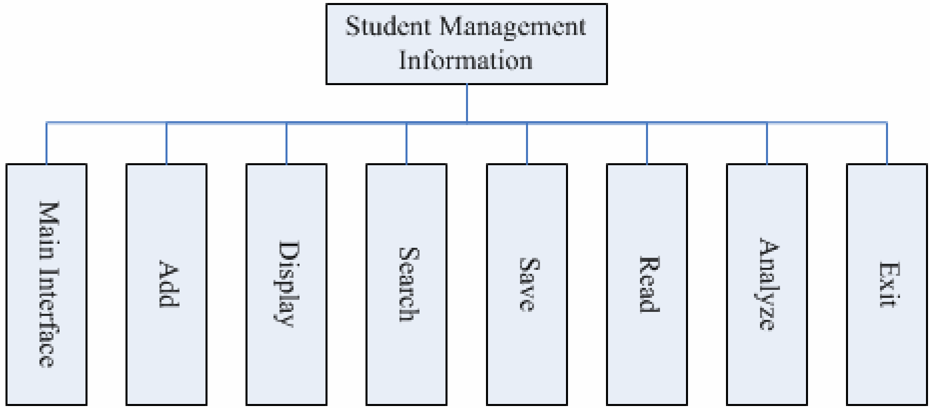
1. Project objectives
2. Project functions
3. Development environment
4. Project iteration

Project Objectives

1. To learn business background of the project by researching off-the-shelf equivalent applications and the function;
2. To learn technologies concerning software development and software system, and to learn the development process of a software project;
3. To grasp the integrated development tool of Microsoft Visual Studio 2010;
4. To grasp basic knowledge of C language, the usage of variable, data type, branch, loop, function, pointer and array;
5 . To understand structure, allocate dynamic memory and linked list;
6. To understand text file and textual file operations in C language;
7. To understand system requirement analysis and design, as well as 3L structure including UI layer, business logic layer and data processing layer, and to be proficiency in iterative development including both technical iteration and functional iteration.
8. To obtain good coding habit, improve C programming capability, and to develop a console program of Student Information Management system.

Project Functions

1. System functional architecture



(1) Main window: displays system name and menu.

(2) Add student info: saves student info in memory.

(3) Display student info: displays student info of memory in form of list.

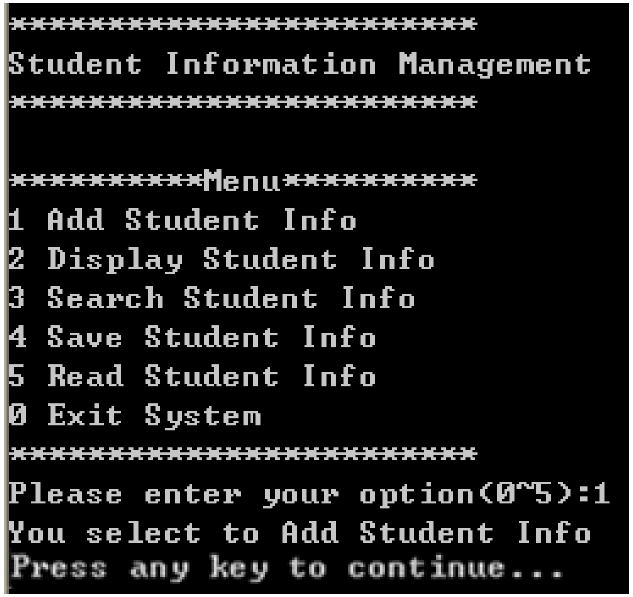
(4) Search student info : finds student info specified by student name and display.

(5) Save student info : saves student info of memory in a textual file.

(6) Read student info : reads student info from text file.

(7) Analyze student information: analyze student information that read from the file.

(8) Exit system: exit the program.



```
************************
Student Information Management
************************

**********Menu**********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
************************
Please enter your option(0~5):1
You select to Add Student Info
Press any key to continue...
```

Implementation

Development Environment

1. Development platform: Windows.

2. Development tool: Microsoft Visual Studio(version 2008, 2010, 2012 etc.)

Example: Visual C++ 2010.

Microsoft® Visual Studio® 2010 Ultimate

For Evaluation Purposes Only

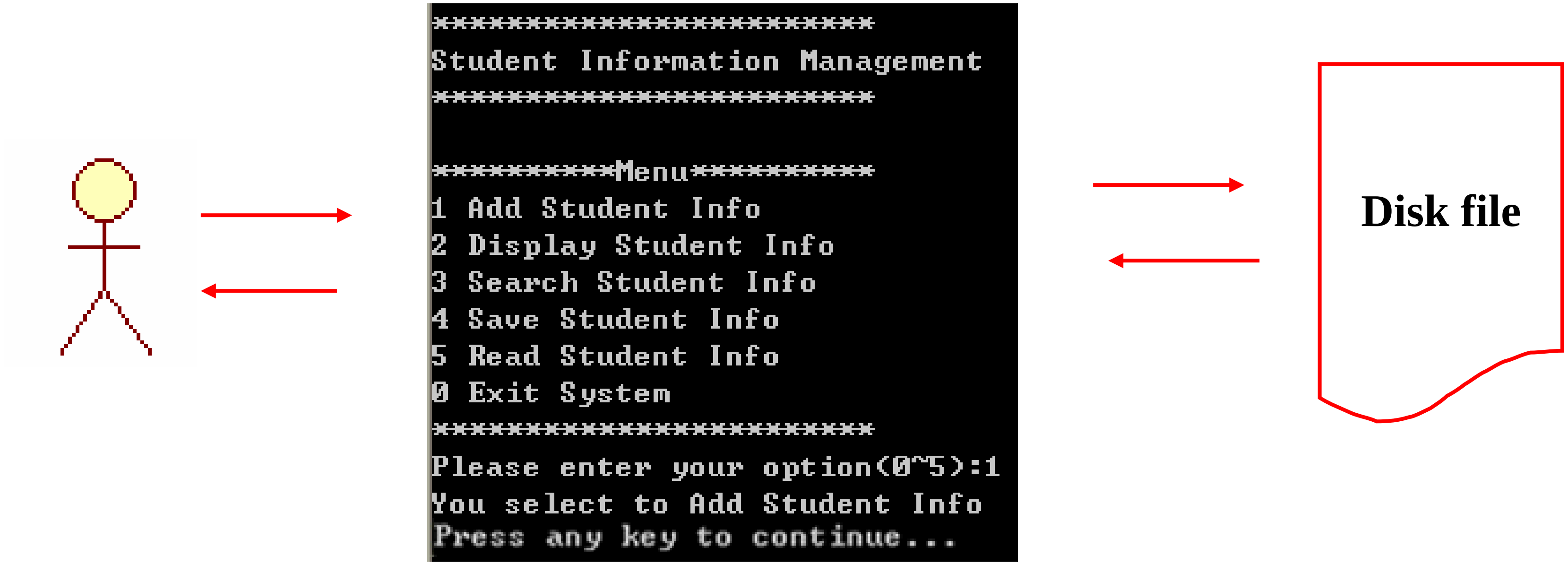
Use of this software should be limited to evaluation purposes only.

This program is protected by U.S. and International copyright laws as described in Help/About.
© 2010 Microsoft Corporation.
All rights reserved.

3. Data storage: textual file (.txt).

Development Environment

5. System structure



```
************************
Student Information Management
************************

**********Menu**********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
************************
Please enter your option(0~5):1
You select to Add Student Info
Press any key to continue...
```

Project Iteration

Functional iterations in project development.

S.N.	Stage	Iteration	Function Description
1	Fundamental knowledge	Create project	Create project and display main window.
2		Add and display student info	One student info item in each addition and display.
3		Add and display student info(structure array)	To add student info by loop and display all.
4		Search student info	Search student info according to name entered by user
5	File operation and dynamic array	Build program structure	Establish program frame in a hierarchical structure.
6		Save student info	Save all added student info in a text file.
7		Read student info	Read and parse the student info from a text file.
8		Dynamic array	Save student info in dynamic structure array.
9	Linked list, array, file	Data structure design	Definition of linked list, linked list operation
10		Manage Student Info: (1) Add (linked list) (2) Display (linked list) (3) Search (linked list) (4) Save (linked list) (5) Read (linked list)	Add : Initialize linked list, insert note Display : Traverse linked list, array Search : Traverse linked list, pointer and function Save : Write file operation, linked list, dynamic array, taking array as function parameter Read :Read file operation, linked list, pointer function

Project Iteration and technology

S.N.	Stage	Iteration name	Technology
1	Fundamental knowledge	Create project	Visual Studio IDE, C program structure, I/O, basic data type, constant, variable, operator, expression, if, switch/case, do/while.
2		Add and display student info	Basic data type, variable, function, pointer, array, structure.
3		Add and display student info (structure array)	Structure array, for, conditional compilation.
4		Search student info	Character string and pointer, for, comparison between character strings.
5	File operation and dynamic array	Build program structure	C program structure, source file and head file, pre-processing
6		Save student info	File writing in C, taking pointer and array as parameter
7		Read student info	File reading in C, character string parsing
8		Dynamic array	Dynamic memory distribution, malloc(), free()
9	Linked list, array, file	Data structure design	Definition of linked list, linked list operation
10		Manage Student Info: (1) Add (linked list) (2) Display (linked list) (3) Search (linked list) (4) Save (linked list) (5) Read (linked list)	Add : Initialize linked list, insert note Display : Traverse linked list, array Search : Traverse linked list, pointer and function Save : Write file operation, linked list, dynamic array, taking array as function parameter Read :Read file operation, linked list, pointer function

Student Information Management

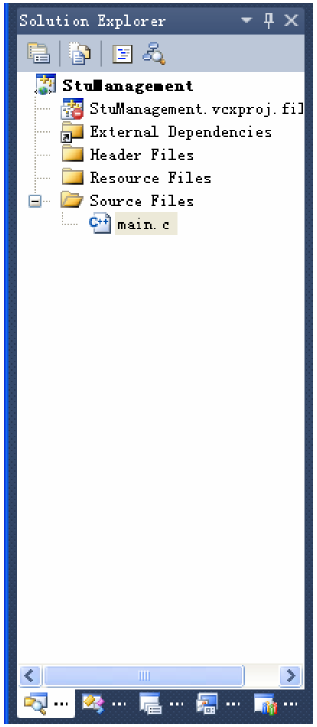
Create Program

Contents

1. Function Introduction
2. System Design
3. Technical analysis
4. Implementation

Function Introduction

1. Create Student Information Management project named "StuManagement".



Solution Explorer

```
StuManagement
  StuManagement.vcxproj.fil
  External Dependencies
  Header Files
  Resource Files
  Source Files
    main.c
```

2. Display system name

```
************************
Student Information Management
************************
```

Function Introduction

3. Display system menu

(1) The system has following menu: “1 Add student info”; “2 Display student info”; “3 Search student info”; “4 Save student info”; “5 Read student info” and “0 Exit system”.

(2) Display format

```
**********Menu**********
1 Add student info
2 Display student info
3 Search student info
4 Save student info
5 Read student info
0 Exit system
************************
```

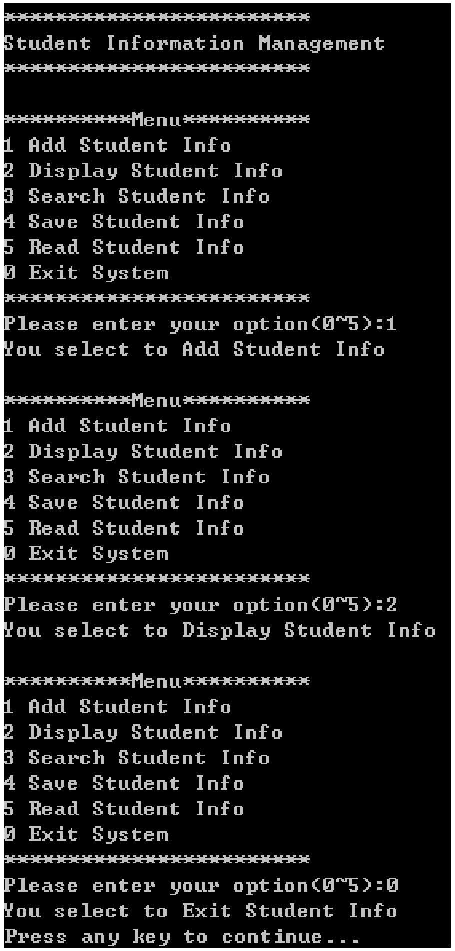
Function Introduction

4. Select menu

There are 6 items in the menu. If user chooses item “Exit system” by inputting 0, then the system displays the name of the item before exit. If the user chooses items other than “Exit system” by inputting number 1~5, then the system displays the name of the selected item before prompts the user to continue.

For example: loop output menu item.
(1) If user chooses item number 1, display “You choose to Add Student Info”, then display the whole system menu(6 pieces).
(2) If user chooses item number 2, display “You choose to Display Student Info”, then display the whole system menu(6 pieces).
(3) If user chooses item number 0, display “You choose Exit System”, then quit the system.

and the like…



```
************************
Student Information Management
************************


**********Menu**********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
************************
Please enter your option(0~5):1
You select to Add Student Info

**********Menu**********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
************************
Please enter your option(0~5):2
You select to Display Student Info

**********Menu**********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
************************
Please enter your option(0~5):0
You select to Exit Student Info
Press any key to continue...
```

System Design

1. Create “Student Information Management” project.

(1) Create an empty project named "StuManagement”.

(2) Add main.c file into the project, and add main() function in it.

2. The design of displaying system name

Call printf() in main() to output the three following string constants.

3. The design of system menu display

Call printf() in main(), to output the following string constant for menu display.

Technical analysis

1. The use of Visual Studio 2010

Method and steps of creating a C project.

2. main()

3. C structure, the running step of a C program.

4. Constant and variable

(1) Character and escape sequence, character string constant.

(2) Definition of variable, initial value assignment and naming.

(3) Common data types: char, int, float, double, etc.

5. Input and output functions

printf() and scanf().

6. Branch structure

If/else and switch/case.

7. Loop structure

do…while.

Implementation

1. “Create Program” function is developed in an iterative way. The iterative order is,

(1) Create Project—create Student Information Management project named "StuManagement".

(2) Display Menu—display system name and system menu, prompt user to select menu item and output user’s selection onto the interface.

(3) Select Menu Item—there are 6 items in system menu. If user selects any number of the 6 menu items from 0 to 5, the system outputs name of the menu item accordingly.

(4) Exit System Menu—if user selects Exit System by inputting 0, then the system displays the text before termination. If the selected item is other than Exit System, i.e. any number from 0 to 5 is entered, then the system displays name text of the selected item before system menu for a next loop of selection by the user.

Implementation

Order	Iteration	Functional description	Technology
1	Create project	Create the project of SIM and name it as "StuManagement".	1. Operation of Visual Studio 2. Create project of C 3. Structure of C 4. Running step of a C program 5. main() 6. printf()
2	Display menu	1. Display system name 2. Display system menu 3. Prompt user to select menu item and display it on the interface.	1. Character and escape sequence 2. Character string constant 3. Definition of variable, initial value assignment and naming 4. Common data types (char/int/float/double) 5. printf() 6. scanf()
3	Select menu	When any of the 6 items is selected by the user, i.e. by inputting any number from 0 to 5, the system displays name text of the menu item.	1. Relational operator(==) 2. Relational expression 3. Statement if 4. Statement switch
4	Exit system menu	If Exit System is selected by the user by inputting 0, then the system displays name text of the item before termination. And can loop output main menu.	1. relational operator (!=) 2. do...while loop

Student Information Management

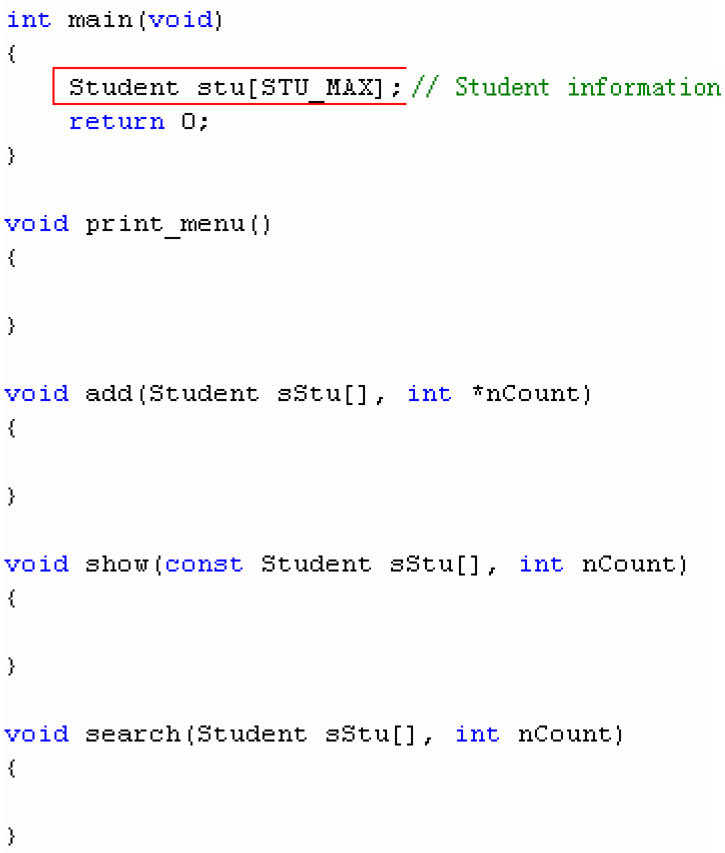
Build Program Structure

Contents

1.Function Introduction
2.System Design
3.Technical analysis
4.Implementation

Function Introduction

1. In the implementation of previous functions, codes are all written in file main.c. In main.c, there are codes of interface processing, data processing and logical processing as well. The readability and expansibility of program are poor.



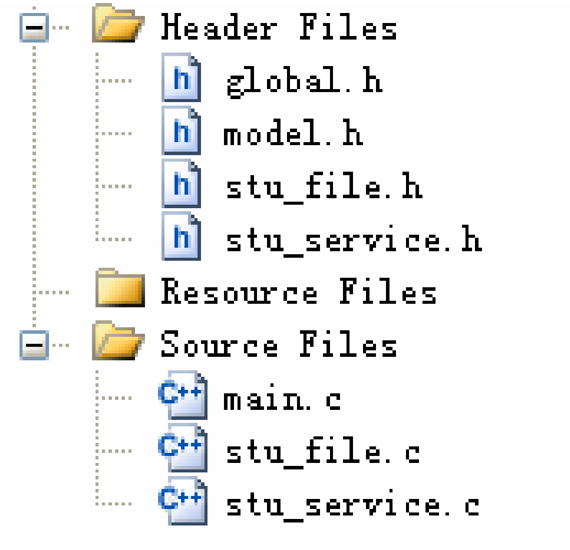
```
int main(void)
{
    Student stu[STU_MAX];// Student information
    return 0;
}

void print_menu()
{

}

void add(Student sStu[], int *nCount)
{

}

void show(const Student sStu[], int nCount)
{

}

void search(Student sStu[], int nCount)
{

}
```

Function Introduction

2. We'll optimize program structure in this iteration.

(1) Take codes of data processing and business processing out of main.c and store in file stu_service.c.

(2) Implement the function of saving input student data in text file, or reversely reading data from file. In this iteration, we'll introduce a new stu_file.c file. The file is used to implement operations on text file.



System Design

Iterative development based on the function of “Search Student Info”.

1.The design of program structure

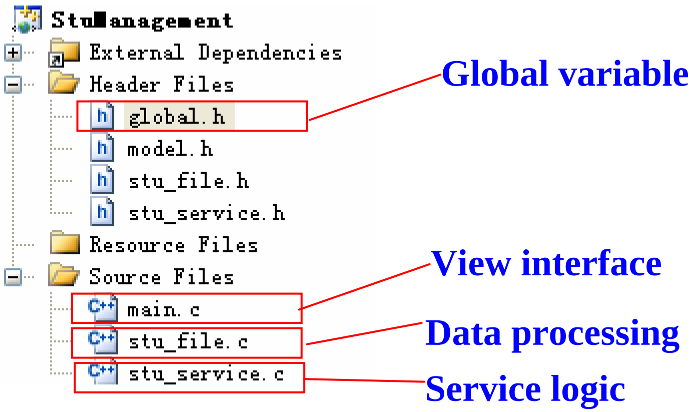
The program is designed in three hierarchical layers, the view layer, logic layer and data access layer. The purpose is to make program structure clear and operational, so as to facilitate later maintenance and upgrading works.

(1) View layer (main.c): receive user inputs, display data and interact with users.

(2) Logic layer (stu_service.c): process relative business logics.

(3) Data access layer (stu_file.c): process data, read or write files.

(4) global.h: define the length of structure array of student info.



```
StuManagement
⊞ External Dependencies
⊟ Header Files
    global.h
    model.h
    stu_file.h
    stu_service.h
  Resource Files
⊟ Source Files
    main.c
    stu_file.c
    stu_service.c
```

Global variable

View interface

Data processing

Service logic

System Design

2. Design of stu_service.c

stu_service.c basically processes business logics. It contains functions listed below.

(1) Variables

Student stu[STU_MAX]: structure array of student info, to save the input student data.

int count: the actual number of student data in structure array.

(2) void add_student(Student sStu)

Save the received student data into structure. It is called by add() in main.c.

(3) Student* search_student(char *input_name)

Base on the input student data, the program searches for information of the input data in structure array and returns result by return value. It is called by search() of main.c.

System Design

3. Design of main.c

main.c processes operations relating to the interface. File main.c contains following functions.

(1) main()

Program entry function. It displays menu and executes corresponding operations according to the selected menu item number.

(2) printf_menu()

It prints menu.

(3) add()

It adds student data, prompts and receives inputted student info. It calls add_student() of stu_service.c to implement the addition.

(4) show()

It displays student data onto the console by taking all student data out of global structure array stu of stu_service.

(5) search()

Search student data, prompt and receive the input searching data. It calls search_student() of stu_service.c to implement searching of student data before displaying onto the console.

Technical analysis

1. Pre-processing.
2. Command of file inclusion.
3. Command of conditional compilation.
4. Command of macro definition.

Implementation

1. Import project StuManagement created in iteration “Search Student Info”.

2. Add file to business logic layer.

(1) Add source file stu_service.c.

(2) Add head file stu_service.h.

3. Add global.h.

Implementation

4. Add file to data access layer

Same as business logic layer, add stu_file.h file and stu_file.c file respectively.

5. Define global variable

(1) Define global variables in stu_service.c, which are structure array stu and int variable nCount.

Structure array stu is used to save student data, while int variable nCount stores the number of student data in structure array.

(2) Include header file model.h in stu_service.c, declaring structure type struct Student.

(3) Add symbol constant STU_MAX in header file global.h, indicating structure array size.

Implementation

6. Add and modify code of student data according to program structure.

(1) Define void add_student(Student sStu) in stu_service.c and save student structure into structure array Student stu[STU_MAX].

(2) Declare void add_student(Student sStu) in stu_service.h.

```
#ifndef STUDENT_SERVICE_H
#define STUDENT_SERVICE_H
#include "model.h"
//Declare function
void add_student(Student _stu_info);
#endif
```

(3) Quote global variable stu and nCount from file stu_service.c and in main.c.

(4) Modify add() in main.c. Call add_student() of stu_service.c to save input student data into global variable stu.

Implementation

7. Modify code of “Display Student Info”.

(1) Modify show() in file main.c.

(2) Get student data out of global structure array by show(), and display.

8. Modify code of “Search Student Info”.

(1) Add Student *search_student(char *input_name) into file stu_service.c.

(2) According to the input student name, search more information in global variable stu by search_student().

(3) Declare search_student() in file stu_service.h.

(4) Modify search() in main.c, and call search_student() to search student data before display.

Student Information Management

Add and Display Student Info (structure array)

Contents

1.Function Introduction

2.System Design

3.Technical analysis

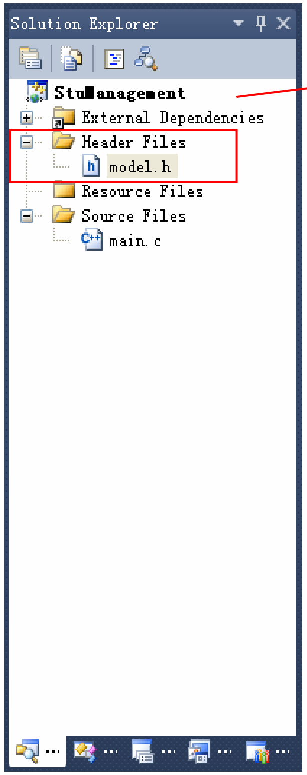
4.Implementation

Function Introduction

Function to be implement in this iteration:

1. To add header file model.h

Create a new header file, .h file, in the project by addition, which is used to declare structure type struct Student. The header file is named model.h.



When there are multiple source files in SIM system and that the structure type struct Student is applied to all source files besides main.c, the declaration of structure type is usually written in a header file .h. Include the header file in source files to which it is applied. It's easy to maintenance and operation in this way. For example, save the definition of show() in file show.c.

Function Introduction

2. Introduction to function of inputting multiple pieces of student info and displaying in form of list.

(1) Input multiple pieces of student info, maximum 50 pieces, and save into structure array before display in form of list.

(2) The student information has 7 segments:

Student ID, Name, Gender, Age, C score, Math score and English score.

1) Student ID: only numbers with maximum length in 10.

2) Name: only letters, no ## is allowed. The max length is 30.

3) Gender: M or F. M for male, and F for female.

4) Age: ranges in value from 0 to 150.

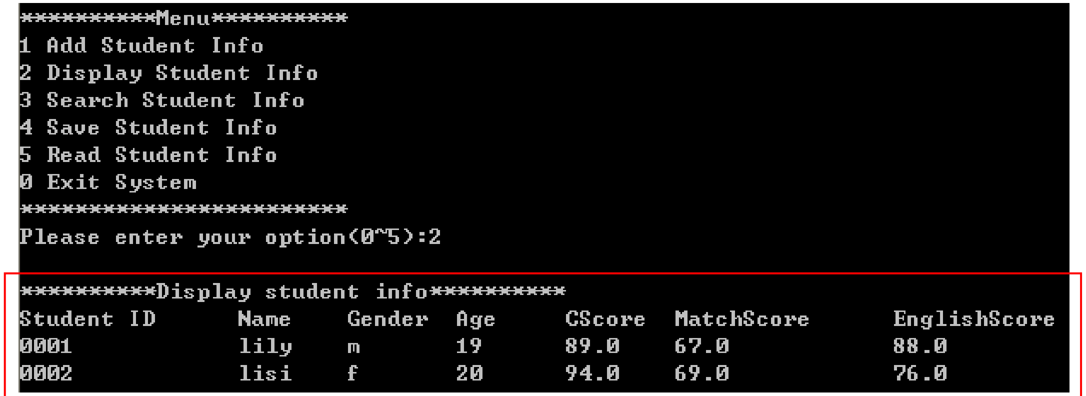
5) C score: only one decimal. Value ranges from 0 to 100.

6) Math score: only decimal. Value ranges from 0 to 100.

7) English score: only one decimal. Value ranges from 0 to 100.

3. Output

When user selects the item Display Student Info, the system displays in form of list



```
**********Menu***********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
************************
Please enter your option(0~5):2

**********Display student info***********
Student ID      Name    Gender  Age     CScore  MatchScore      EnglishScore
0001            lily    m       19      89.0    67.0            88.0
0002            lisi    f       20      94.0    69.0            76.0
```

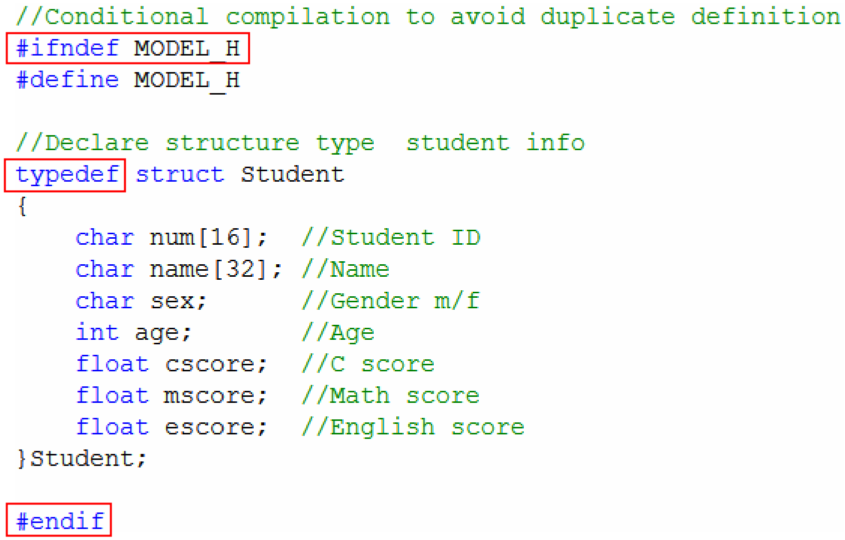
System Design

1. Design of header file model.h

(1) Add header file model.h under project StuManagement.

(2) Declare structure variable type struct Student in file model.h. In order to avoid duplicate definition of structure type, conditional compiling command of #ifndef ...#endif is used.

Use typedef, name structure struct Student as “Student”.



```
//Conditional compilation to avoid duplicate definition
#ifndef MODEL_H
#define MODEL_H

//Declare structure type  student info
typedef struct Student
{
    char num[16];  //Student ID
    char name[32]; //Name
    char sex;      //Gender m/f
    int age;       //Age
    float cscore;  //C score
    float mscore;  //Math score
    float escore;  //English score
}Student;

#endif
```

The role of typedef is to declare a simple structure name for the complex type, such as structure type.

Note： typedef is used to give a new type name to the existing data type. It does not create any new data type.

(3) **Include header file model.h in file main.c.**

Structure type struct Student is used in main() in file main.c, so you need to include header file model.h ahead of main(). The method is #include “model.h”.

System Design

2. Define structure array to save student data.

In the beginning of main (), alter the structure variable stu defined in the iteration of “Add and Display Student Info” into structure array with size in 50.

```
//The maximum number of input pieces is 50.
#define STU_MAX 50
......
int main (  )
{
    int select = -1;//the selected menu item
    Student stu[STU_MAX];//SI
    int count = 0;//the actual number of student data
    ......
    return 0;
}
```

System Design

3. Design of add()

Call add() in main() to get the input student data and save it in structure array. The call is located after case 1 in switch statement.

(1) Function: input a piece of student info.

(2) Prototype: void add(Student sStu[], int *nCount);

(3) Function parameter:

Student sStu[]: the name of structure array of student information.

int *nCount: the number of student information in structure array.

4. Design of show()

Call show() in main() to display student data stored in structure array. The call is located after case 2 in switch statement.

(1) Function: display student information.

(2) Prototype: void show(const Student sStu[], int nCount);

(3) Function parameter:

const struct Student sStu[]: the name of structure array for student data.

int nCount: the actual number of student data in structure array.

Technical analysis

1. structure array

(1) Define structure array.

(2) Access the member of structure array.

2. Pre-compilation command

(1) Macro definition.

(2) Conditional compilation.

(3) File include.

3. for loop

4. typedef

5. Source file and Header file

Implementation

Develop the function of “Add and Display (structure array)” in an iterative way.
The procedure is:

(1) **Add and Display (structure array):** Input multiple pieces of student data (maximum 50 pieces) and save them into structure array. Display data in form of list.

(2) Optimize program structure: Introduce header file (model.h) to declare structure type of struct Student.

S.N.	Iteration	Functional description	Technology
1	Add and Display (structure array)	Input multiple pieces of student data, maximum 50 pieces, and save them into structure array. Display data in form of list.	1. Define structure array 2. Access member of structure array 3. Macro definition (#define) 4. Loop for 5. typedef
2	Optimize program structure	Introduce header file model.h to declare structure type struct Student.	1. Conditional compilation (#ifndef...#endif) 2. File inclusion (#include) 3. Source file and Header file

Student Information Management

Read Student Info

Contents

1.Function Introduction
2.System Design
3.Technical analysis
4.Implementation

Function Introduction

In the last iteration, we implemented the function of saving student info in student.txt under project directory.

In this iteration, we'll implement reading and parsing student info.

Functions are described as follow.

1. Introduction

Read student info from file student.txt under project directory, and save in memory. And display student info onto the console.

2. Output

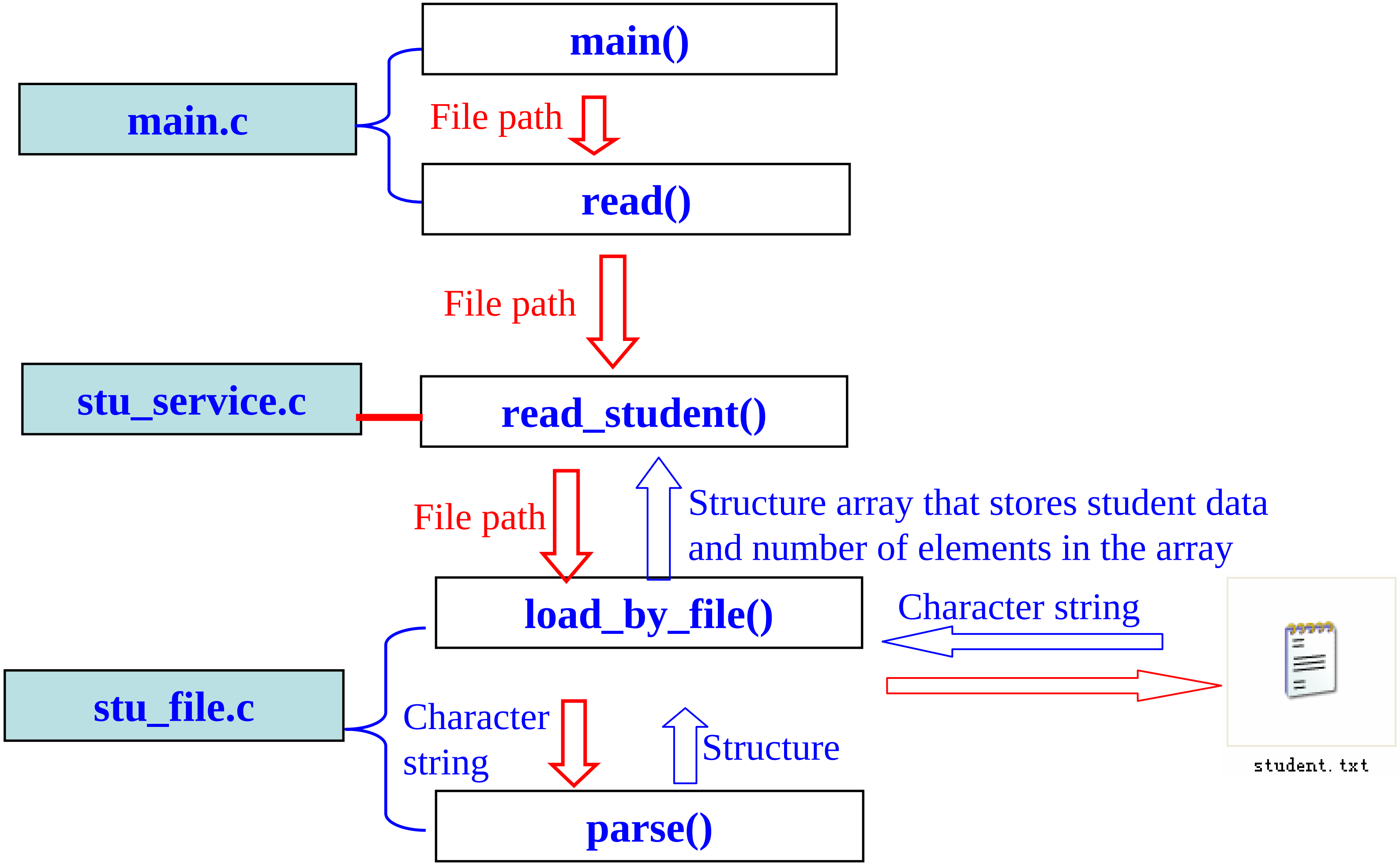
(1) For reading failure, the program prompts "Student Info Reading Failure".

(2) For reading success, the program prompts "Student Info Reading Success", with file path being provided.

System Design

1. Function design of Read Student Info.

Read Student Info is to parse student info out of file and save in global array stu in stu_service.c.



System Design

2. main.c file design

main.c is for interface processing, which contains main() and save().

(1) main()

It’s program entry function, which displays system menu and receives user input figure.

When user inputs “5” for Reading, _getcwd() is used to obtain the absolute path of project directory before setting file reading path as “Project Directory + student.txt”.

Next is to pass the path to read().

(2) read()

1) Prototype: void read(char* path)

2) Parameter: path of the textual file read by char* path, which is an absolute path.

3) Function: receive file path from main(), and pass it to stu_service.c.

System Design

3. stu_service.c file design

File stu_service.c is logic processing file. Student data is stored in stu array and number of student data is stored in count. Function read_student() is used to read student info from the textual file under specified path.

(1) Prototype: int read_student(char* path)

(2) Parameter: char* path is used to save the path of file that stores student info, which is absolute path.

(3) Function: pass the file path into file stu_file.c, and get read student info structure array and array elements.

(4) Return value: 0 for saving failure, 1 for saving success.

System Design

4. stu_file.c file design

File stu_file.c is designed for data processing, mainly data exchanges between program and file.

(1) load_by_file() function

1) Prototype: int load_by_file(char* path, Student *sStu, int* count)

2) Parameter: char* path for file path; Student *sStu for student info structure array pointer; int* count for count pointer of array elements.

3) Function:

Read textual file line by line under specified path.

Convert each textual line into a student info structure by parsing.

Save all student info structure in the array of structure.

Return student info structure array and element number by parameters.

4) Return value: 0 for reading failure, 1 for reading success.

System Design

(2) parse() function

1) Prototype: Student parse(char* info)

2) Parameter: char* info for student info character string array pointer.

3) Function: convert student info character string into student info structure.

4) Return value: student info structure after parsing.

Technical analysis

1. _getcwd() function

2. structure array

3. Array and pointer as function parameter.

4. File reading operation in C.

5. Parsing character string.

Implementation

1. Import project StuManagement created in iteration “Save Student Info”.

2. To implement reading in view layer main.c.

(1) Get file path of the current project directory in main(), which is absolute path.

(2) When user enters 5 to Read Student Info, the program passes file path to read().

(3) Define function void read(char* path) in main.c. Call read_student() from file stu_service.c.

3. To implement reading in business logic layer stu_service.c.

Add int read_student(char *path) in stu_service.c, to receive file path from interface. Call load_by_file() from file stu_file.c, to obtain the student info structure array and student number under the path.

4. To implement reading in data access layer stu_file.c.

(1) Add load_by_file() to stu_file.c, read text file under specified path, and pass reading content to parse() for parsing.

(2) Parse() resolves one row of text into a student structure variable, and returns to load_by_file().

(3) Function load_by_file() saves parsing result into the array of structure Student, and passes structure array and element number to stu_service.c by parameter.

5. Compile and run the program.

Student Information Management

Save Student Information

Contents

1.Function Introduction

2.System Design

3.Technical analysis

4.Implementation

Function Introduction

In the last iteration, we design the program in three hierarchical layers, the view layer, business logic layer and data access layer. And basing on the hierarchical structure, the function of Add, Display and Search Student Info is created.

In this iteration, we'll implement the function of saving added student info into specified textual file basing on the hierarchical structure.

1. Introduction

After the addition of student info, and given the fact that item number 4, or say Save Student Info, is selected, the program attaches new student data to the end of student.txt under project directory by saving.

2. Output

(1) If saving failed, prompt user of the message "Saving of student data failed."

(2) If saving succeeded, prompt user of the message "Saving of student data succeeded", as well as information of file path.

The format of student data in text file is as follow.

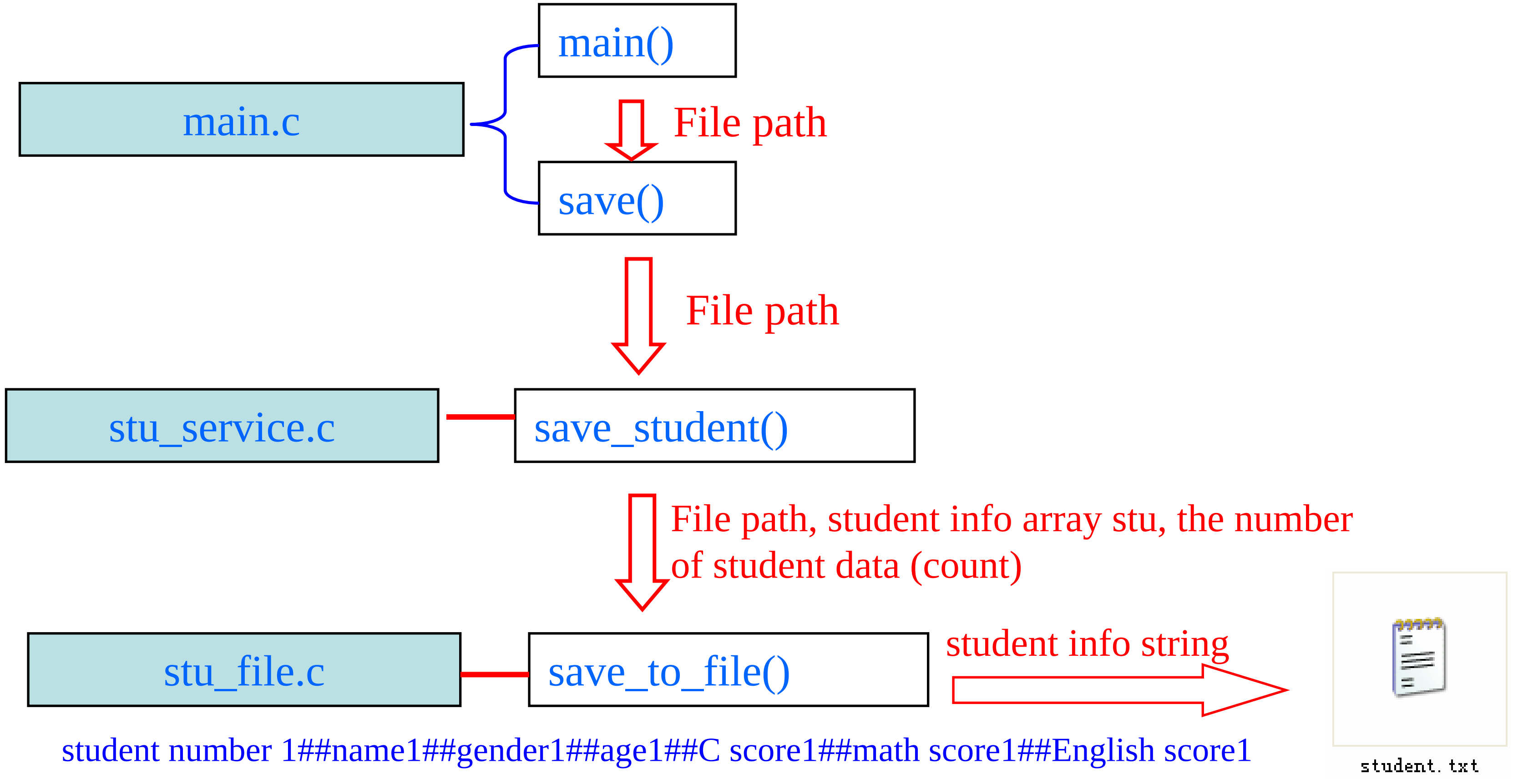
```
student num1##name1##gender1##age1##C score1##Math score1##English score1
student num2##name2##gender2##age2##C score2##Math score2##English score2
```

System Design

1. Functional design of Save Student Info.

In the iteration of “Build Program Structure”, the program is designed in three layers, the view layer of main.c, the logic layer of stu_service.c and the data access layer of stu_file.c.

The function of Add Student Info is to save student data into the global array of stu in stu_service.c. To save student data is to save data of the global array in file.



student number 1##name1##gender1##age1##C score1##math score1##English score1

student number 2##name2##gender2##age2##C score2##math score2##English score2

System Design

2. Design of main.c file

File main.c processes operations relating to the interface, including two functions of main() and save().

(1) main()

It is the program entry function. Display system menu and receive item number selected by the user.

When user selects item number 4 to Save Student Info, _getcwd() is used to get the absolute path of project folder. Then path is set as “Project Directory+student.txt”. And then pass it to save().

(2) save()

1) Prototype: void save(char* path).

2) Parameter: char* path saves the path of file that stores student data (absolute path).

3) Function: receive the file path passed from main(), and pass it to file stu_service.c.

System Design

3. Design of file stu_service.c

File stu_service.c is logic processing file, storing student data in array stu and number of student data in count. Function save_student() saves student data in textual file.

(1) Prototype: int save_student(char* path)

(2) Argument: char* path saves the path of file that stores student data (absolute path).

(3) Function: pass information such as file path, student data array and data number to file stu_file.c.

(4) Return value: 0 indicates saving failure; 1 indicates saving success.

System Design

4. Design of stu_file.c

File stu_file.c is data processing file. It processes data exchanges between program and the file. In the function of saving student data, save_to_file() is used to save student data to the end of textual file under specified path..

(1) Prototype: int save_to_file (char* path, Student stu[], int count).

(2) Parameter: char* path stores file path; Student stu[] is the array that stores student data; while int count stores the number of element in the array.

(3) Function: convert student data in the array into character strings, and save them in a textual file under specified path in sequence.

(4) Return value: 0 indicates saving failure; 1indicates saving success.

Technical analysis

1. _getcwd()
2. structure array
3. Take array and pointer as function parameters.
4. The basic knowledge of C file.
5. C file operation.

Implementation

1. Import project StuManagement created in iteration “Build Program Structure”.

2. Implement student data saving in data access layer (stu_file.c).

Add save_to_file() into file stu_file.c, to receive and transfer data such as file path, student data array and student data number. Save student data to the end of file that is stored under the path.

3. Implement student data saving in business logic layer (stu_service.c).

Add int save_student(char *path) into file stu_service.c, to receive file path from view layer. Call save_to_file() from file stu_file.c, and then pass file path, student data structural array and the element number to save_to_file().

4. Implement student data saving in view layer (main.c).

(1) Get the file path of current project folder in main(), which is absolute path.

(2) When the user selects 4 to Save Student Info, the program passes file path to save().

(3) Define void save(char* path) in main.c. Call save_student() from file stu_service.c.

5. Compile and run program.

Student Information Management

Search Student Info

Contents

1.Function Introduction

2.System Design

3.Technical analysis

4.Implementation

Function Introduction

In the iteration of “Add and Display Student Info (structure array)”, we implemented functions of inputting multiple pieces of student info, maximum 50 pieces, and saving it into the structure array before displaying it in form of list.

Function to be implement in this iteration is as follow.

1. Introduction

(1) After user selects to Search Student Info, the system prompts user to input student name for searching. According to name entered, the system traverses structure array to find student data. If data is found, display it in form of list. If no data is found, prompt user of the message.

(2)The student info has 7 segments.

Student ID, Name, Gender, Age, C score, Math score and English score.

1) Student ID: only numbers with maximum length in 10.

2) Name: only letters, no ## is allowed. The max length is 30.

3) Gender: M or F. M for male, and F for female.

4) Age: ranges in value from 0 to 150.

5) C score: only one decimal. Value ranges from 0 to 100.

6) Math score: only decimal. Value ranges from 0 to 100.

7) English score: only one decimal. Value ranges from 0 to 100.

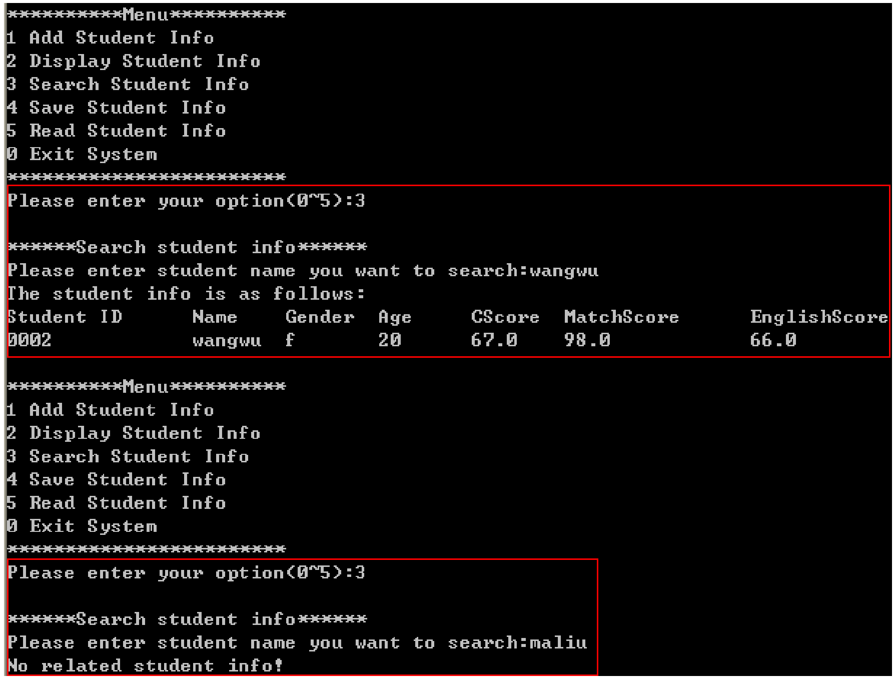
Function Introduction

2. Output

(1) If data is found, the program displays it in form of list. That is to say, to display in seven columns.

The seven columns are Student ID, Name, Gender, Age, CScore, MScore and EScore.

(2) If no data is found, the program outputs message “No data is found!”.



```
***********Menu***********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
**************************
Please enter your option(0~5):3

******Search student info******
Please enter student name you want to search:wangwu
The student info is as follows:
Student ID      Name    Gender  Age     CScore  MatchScore      EnglishScore
0002            wangwu  f       20      67.0    98.0            66.0

***********Menu***********
1 Add Student Info
2 Display Student Info
3 Search Student Info
4 Save Student Info
5 Read Student Info
0 Exit System
**************************
Please enter your option(0~5):3

******Search student info******
Please enter student name you want to search:maliu
No related student info!
```

System Design

Iterative development based on the function of “Add and Display Student Info (structure array)”.

1. **Program procedure design**

(1) Execute do...while loop after system name display.

(2) do...while executes the loop body first. 3 steps in the loop.

Step 1: call print_menu() to display system menu.

Step 2: get the menu item selected by user and save it in integer variable select.

Step 3: base on user selection, or value of select. If user selects to Search Student Info, or select value is 3, the program calls **search()** to input student name for searching before actually search it according to name data.

(3) After the execution of do...while loop, the program decides whether user selects to Exit System. The loop condition is select != 0. If yes, terminate the program; Otherwise, continue the loop cycle.

System Design

2. Design of search()

(1) Define search() after show() in file main.c.

(2) Call search() in main, after case 3 in switch statement.

(3) **search() function**

Inputs student name for searching. According to name information, the program traverses structure array to find the first name same as input. Display the student data in form of list if the name is found. Otherwise, prompt user of message “No data is found”.

① **Function:** search student info.

② **Prototype:** void search(Student sStu[], int nCount) ;

③ **Parameter:**

Student sStu[]: the name of structure array taking student info.

int nCount: the actual number of student data in structure array.

Technical analysis

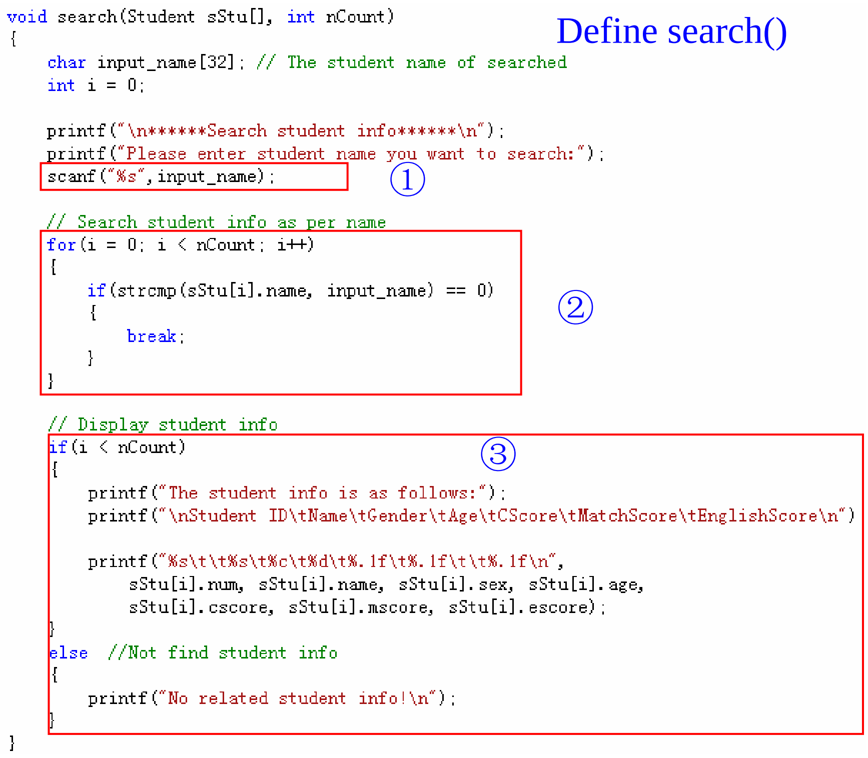
1. Access the member of structure array
2. for loop
3. break statement
4. if statement
5. strcmp()

Implementation

1. Import project StuManagement created in iteration “Add and Display Student Info (structure array)”.

2. Add search().

(1) Define search() after show() in file main.c.



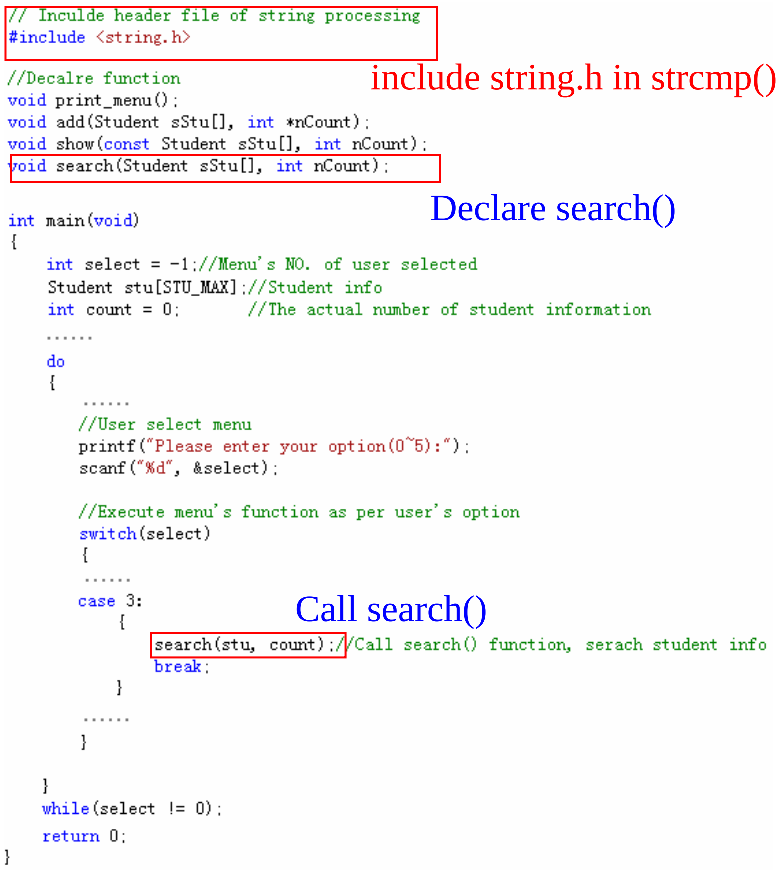
Define search()

```
void search(Student sStu[], int nCount)
{
    char input_name[32]; // The student name of searched
    int i = 0;

    printf(“\n******Search student info******\n”);
    printf(“Please enter student name you want to search:”);
    scanf(“%s”,input_name);

    // Search student info as per name
    for(i = 0; i < nCount; i++)
    {
        if(strcmp(sStu[i].name, input_name) == 0)
        {
            break;
        }
    }

    // Display student info
    if(i < nCount)
    {
        printf(“The student info is as follows:”);
        printf(“\nStudent ID\tName\tGender\tAge\tCScore\tMatchScore\tEnglishScore\n”)

        printf(“%s\t\t%s\t%c\t%d\t%.1f\t%.1f\t\t%.1f\n”,
            sStu[i].num, sStu[i].name, sStu[i].sex, sStu[i].age,
            sStu[i].cscore, sStu[i].mscore, sStu[i].escore);
    }
    else  //Not find student info
    {
        printf(“No related student info!\n”);
    }
}
```

Implementation

(2) Call search after “case 3:” in main().

(3) Declare search() ahead of main().



```
// Inculde header file of string processing
#include <string.h>

//Decalre function
void print_menu();
void add(Student sStu[], int *nCount);
void show(const Student sStu[], int nCount);
void search(Student sStu[], int nCount);

int main(void)
{
    int select = -1;//Menu’s NO. of user selected
    Student stu[STU_MAX];//Student info
    int count = 0;      //The actual number of student information
    ......
    do
    {
        ......
        //User select menu
        printf(“Please enter your option(0~5):”);
        scanf(“%d”, &select);

        //Execute menu’s function as per user’s option
        switch(select)
        {
        ......
        case 3:
            {
                search(stu, count);//Call search() function, serach student info
                break;
            }
        ......
        }

    }
    while(select != 0);
    return 0;
}
```

include string.h in strcmp()

Declare search()

Call search()

Thanks