# 软件工程

**学院:计算机学院**

**班级：校交1601**

**专业：计算机科学与技术**

**姓名：孙张昱**

**学号：U201613986**

目录

[一、 问题定义及可行性分析 3](#_Toc530343432)

[1. 问题定义 3](#_Toc530343433)

[2 可行性分析 3](#_Toc530343434)

[一、 需求分析 7](#_Toc530343435)

[1. 系统分析结果 7](#_Toc530343436)

[2. 任务概述 8](#_Toc530343437)

[3.需求规定 9](#_Toc530343438)

[4 运行环境规定 11](#_Toc530343439)

[5 设计系统的逻辑关系 12](#_Toc530343440)

[6 数据字典 14](#_Toc530343441)

[二、 概要设计 15](#_Toc530343442)

[1． 引言 15](#_Toc530343443)

[2. 总体设计 15](#_Toc530343444)

[3. 接口设计 17](#_Toc530343445)

[4. 系统数据结构设计 17](#_Toc530343446)

[三、 详细设计 19](#_Toc530343447)

[1. 引言 19](#_Toc530343448)

[2. 程序系统的结构 20](#_Toc530343449)

[3. 程序设计说明 21](#_Toc530343450)

[四、 测试方案 26](#_Toc530343451)

[1,引言 26](#_Toc530343452)

[2 计划 26](#_Toc530343453)

[3.测试数据说明 26](#_Toc530343454)

[4. 运行环境 26](#_Toc530343455)

[5. 使用过程 28](#_Toc530343456)

[结论 31](#_Toc530343457)

[五、 项目管理 32](#_Toc530343458)

[七、体会 33](#_Toc530343459)

[八、附录 34](#_Toc530343460)

# 问题定义及可行性分析

## 问题定义

1. 问题的项目名称:科研信息管理系统
2. 使用方各院系
3. 问题的概括定义：

某大学为了方便管理，各个院系都成立了独立的研究团队，每个团队由若干个教师组成，科研项目的立项、申报、研究、结题全过程都是以团队为单位来开展的。请设计一个程序对各个院系的团队 信息以及团队的科研项目信息进行有效地管理，以期提高项目管理的效率。

1. 项目的目标

4.1 使用方方面：

a. 使学校方便查询各院系各团队科研项目情况

b. 优化学校资源分配

c. 便于进行数据统计和评优

4.2 开发方方面:

a.合理预算开发成本，付出较低的开发费用

b.实现预期的软件功能，满足用户要求

c.提高软件开发软件率，及时交付使用

## 2 可行性分析

1. 引言
   1. 编写目的

某大学为了方便管理，各个院系都成立了独立的研究团队，每个团队由若干个教师组成，科研项目的立项、申报、研究、结题全过程都是以团队为单位来开展的。请设计一个程序对各个院系的团队 信息以及团队的科研项目信息进行有效地管理，以期提高项目管理的效率

* 1. 背景

对某大学各院系科研项目信息进行管理，包括各院系基本信息、科研团队基本信息和科研项目基 本信息。

1. 可行性分析

数据如下：



* 1. 要求

本系统需要实现数据维护，数据查询和数据统计三个主要功能模块，另外根据情况添加辅助功能

模块。下面给出了三个主要模块的功能需求，辅助功能模块根据各人的理解和分析自己设计（鼓励！）。

(1)．数据维护

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据

维护功能，又分为三个子模块：

* 院系基本信息维护：包括对院系基本信息的录入、修改和删除等功能。
* 科研团队基本信息维护：包括对科研团队基本信息的录入、修改和删除等功能。
* 在科研项目基本信息维护：包括对科研项目基本信息的录入、修改和删除等功能。

(2)．数据查询

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据查询功能，又分为三个子模块。（假设院系负责人以科研团队没有同名的情况）

1）院系信息查询功能

* 以院系负责人为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系负责人

为“张三”的院系基本信息。

* 以院系名称的全部或一部分为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系名称中包含“comp”的所有院系基本信息。

2）科研团队基本信息查询功能

* 团队名称的全部或一部分为条件查找并显示满足条件的科研团队基本信息。例如，查找并显

示团队名称中包含“网络安全”的科研团队基本信息。

* 以教师人数为条件查找并显示满足条件的科研团队基本信息。例如，查找并显示教师人数大

于 5 人的所有科研团队基本信息。

3）科研项目基本信息查询功能

* 项目编号为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示项目编号为

“0121210044”的科研项目基本信息。

* 以所属团队为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示所属团队为

“网络安全”的所有科研项目基本信息。

(3)．数据统计

本模块实现五个方面的数据统计功能，前四个功能需求已给出，第五个自行设计。

1. 统计各院系教师总数，研究生总数，及研究生与教师的人数比（保留 2 位小数），按学生教师人数比值降序排序后，输出统计结果。
2. 统计某年度各院系科研项目数，973 项目数，863 项目数，及科研总经费，科研项目数降序排序后输出。
3. 统计历年来类别为国家自然科学基金的科研项目数最多的 10 个科研团队，按项目数降序排序后输出科研团队名称，类别为国家自然科学基金的科研项目数，以及项目经费总数。
4. 统计科研项目数和教师人数的比值最高的 5 个科研团队，按比值（保留两位小数）降序排序后，输出科研团队名称，教师人数，科研项目数，项目数和教师人数比值。
5. 有关科研项目其他方面的数据统计。
   1. 目标

实现对信息的统计增删查改

* 1. 条件、假定和限制

信息更新不及时，需要人工进行录入

* 1. 评价尺度

效益大于成本：

技术可行，现有技术可以承担开发任务，并且开发时间允许

操作可行：软件能被用户快速上手

1. 系统建议
   1. 费用开支

设备费用，系统维护费用，开发人员工资

* 1. 人员

开发人员1人，测试1人，用户体验1人

* 1. 设备

PC，客户端，软件

* 1. 局限性

硬件设施有限，第一次开发，开发人员缺乏经验

* 1. 影响
     1. 对原来工作影响

不要专人进行记录统计分析

* + 1. 对软件影响

操作系统：windows

开发工具：c++  
软件平台：windows

* + 1. 对用户单位影响

提高了工作效率

* 1. 技术可行性

该软件不是大型软件，不需要很高的硬件。软件逻辑并不困难，对开发人员要求不高

1. 结论

该软件的实现条件已经完全具备。没有资金压力，对人员要求不高，可以进行开发

# 需求分析

## 系统分析结果

。1.1编写目的

某大学为了方便管理，各个院系都成立了独立的研究团队，每个团队由若干个教师组成，科研项目的立项、申报、研究、结题全过程都是以团队为单位来开展的。请设计一个程序对各个院系的团队信息以及团队的科研项目信息进行有效地管理，以期提高项目管理的效率。

* 1. 背景

对某大学各院系科研项目信息进行管理，包括各院系基本信息、科研团队基本信息和科研项目基本信息。

1. 院系基本信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 院系名称 | char[20] | “Computer” |  |
| 负责人 | char[12] | “张三” |  |
| 联系电话 | char[15] | “13312345678” |  |

1. 科研团队基本信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 团队名称 | char[30] | “网络安全” |  |
| 负责人 | char[12] | “李四” |  |
| 教师人数 | int | 5 |  |
| 研究生人数 | int | 20 |  |
| 所属院系 | char[20] | “Computer” |  |

1. 科研项目基本信息表

|  |  |  |
| --- | --- | --- |
| 中文字段名 | 类型及长度 | 举例 |
| 项目编号 | char[15] | “0121210044” |
| 项目类别 | char | ‘1’-‘5’分别表示 973 计划项目、国家自然科学基金项目、863 计划项目、国际合作项目、横向项目 |
| 起始时间 | char[8] | “2015/03” |
| 项目经费 | float | 23.5 单位： 万元 |
| 项目负责人 | char[12] | “王五” |
| 所属团队 | char[30] | “网络安全” |

## 2. 任务概述

2.1 目标

实现以下功能

(1)．数据维护

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据

维护功能，又分为三个子模块：

* 院系基本信息维护：包括对院系基本信息的录入、修改和删除等功能。
* 科研团队基本信息维护：包括对科研团队基本信息的录入、修改和删除等功能。
* 在科研项目基本信息维护：包括对科研项目基本信息的录入、修改和删除等功能。

(2)．数据查询

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据查询功能，又分为三个子模块。（假设院系负责人以科研团队没有同名的情况）

1）院系信息查询功能

* 以院系负责人为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系负责人

为“张三”的院系基本信息。

* 以院系名称的全部或一部分为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系名称中包含“comp”的所有院系基本信息。

2）科研团队基本信息查询功能

* 团队名称的全部或一部分为条件查找并显示满足条件的科研团队基本信息。例如，查找并显

示团队名称中包含“网络安全”的科研团队基本信息。

* 以教师人数为条件查找并显示满足条件的科研团队基本信息。例如，查找并显示教师人数大

于 5 人的所有科研团队基本信息。

3）科研项目基本信息查询功能

* 项目编号为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示项目编号为

“0121210044”的科研项目基本信息。

* 以所属团队为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示所属团队为

“网络安全”的所有科研项目基本信息。

(3)．数据统计

本模块实现五个方面的数据统计功能，前四个功能需求已给出，第五个自行设计。

1. 统计各院系教师总数，研究生总数，及研究生与教师的人数比（保留 2 位小数），按学生教师人数比值降序排序后，输出统计结果。
2. 统计某年度各院系科研项目数，973 项目数，863 项目数，及科研总经费，科研项目数降序排序后输出。
3. 统计历年来类别为国家自然科学基金的科研项目数最多的 10 个科研团队，按项目数降序排序后输出科研团队名称，类别为国家自然科学基金的科研项目数，以及项目经费总数。
4. 统计科研项目数和教师人数的比值最高的 5 个科研团队，按比值（保留两位小数）降序排序后，输出科研团队名称，教师人数，科研项目数，项目数和教师人数比值。
5. 有关科研项目其他方面的数据统计。

2.2 用户特点

用户为各院系管理人员，有基本的电脑常识即可。使用高峰在每学期开始和末端。

2.3 假定和约束

本软件是一款小型的信息系统管理软件，实现对信息基本的查找，录入，统计，修改功能。不要ui。

## 3.需求规定

3.1 对功能的规定

实现以下功能

(1)．数据维护

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据

维护功能，又分为三个子模块：

* 院系基本信息维护：包括对院系基本信息的录入、修改和删除等功能。
* 科研团队基本信息维护：包括对科研团队基本信息的录入、修改和删除等功能。
* 在科研项目基本信息维护：包括对科研项目基本信息的录入、修改和删除等功能。

(2)．数据查询

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据查询功能，又分为三个子模块。（假设院系负责人以科研团队没有同名的情况）

1）院系信息查询功能

* 以院系负责人为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系负责人

为“张三”的院系基本信息。

* 以院系名称的全部或一部分为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系名称中包含“comp”的所有院系基本信息。

2）科研团队基本信息查询功能

* 团队名称的全部或一部分为条件查找并显示满足条件的科研团队基本信息。例如，查找并显

示团队名称中包含“网络安全”的科研团队基本信息。

* 以教师人数为条件查找并显示满足条件的科研团队基本信息。例如，查找并显示教师人数大

于 5 人的所有科研团队基本信息。

3）科研项目基本信息查询功能

* 项目编号为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示项目编号为

“0121210044”的科研项目基本信息。

* 以所属团队为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示所属团队为

“网络安全”的所有科研项目基本信息。

(3)．数据统计

本模块实现五个方面的数据统计功能，前四个功能需求已给出，第五个自行设计。

1. 统计各院系教师总数，研究生总数，及研究生与教师的人数比（保留 2 位小数），按学生教师人数比值降序排序后，输出统计结果。
2. 统计某年度各院系科研项目数，973 项目数，863 项目数，及科研总经费，科研项目数降序排序后输出。
3. 统计历年来类别为国家自然科学基金的科研项目数最多的 10 个科研团队，按项目数降序排序后输出科研团队名称，类别为国家自然科学基金的科研项目数，以及项目经费总数。
4. 统计科研项目数和教师人数的比值最高的 5 个科研团队，按比值（保留两位小数）降序排序后，输出科研团队名称，教师人数，科研项目数，项目数和教师人数比值。

有关科研项目其他方面的数据统计

有基本的用户界面

3.2 对性能的规定

3.2.1 时间特性要求

a. 响应时间：界面按键和功能能够立刻给出结果

b，更新处理时间，能在修改完数据后立即保存并及时调用

c. 保存时间：能够立即保存现有数据

3.3 输入输出要求

输入和输出不能超过所规定的数据类型的最大范围，不接受非法输入

3.4 故障处理要求

遇到非法操作时能够给出提示并退回上一层功能

3.5 其他方面要求

程序易读性强

软件的可靠性高

能满足基本要求

## 4 运行环境规定

4.1设备

a.处理器及内存容量 intel i3 以上内存不小于512MB

b 外存容量，联机或脱机，媒体及存储格式:外存200GB 设备为winxp以上pc

c功能键或其他专用硬件 无

4.2 支持软件

a.操作系统：winxp及以上

b编译程序及测试程序为codeblocks 16.01

4.3接口

无

## 5 设计系统的逻辑关系

5.1 功能结构设计



5.2 文件功能设计



5.3 数据查询



5.4 数据统计

5.5 数据结构

…

head

院系

1

院系

2

院系n

NULL

…

团队

1

基本信息

团队

2

基本信息

团队

n

基本信息

科研项目信息链

1

科研项目信息链

2

科研项目信息链

n

head1

团队

1

信息

团队

2

信息

团队

n

信息

团队

1

项目

1

团队

1

项目

2

团队

1

项目

n

团队

2

项目

1

团队

2

项目

2

团队

2

项目

n

团队

n

项目

1

团队

n

项目

2

团队

n

项目

n

head2

院系

1

院系

2

院系

3

## 6 数据字典

6.1 学院

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 院系名称 | char[20] | “Computer” |  |
| 负责人 | char[12] | “张三” |  |
| 联系电话 | char[15] | “13312345678” |  |

6.2 团队

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 团队名称 | char[30] | “网络安全” |  |
| 负责人 | char[12] | “李四” |  |
| 教师人数 | int | 5 |  |
| 研究生人数 | int | 20 |  |
| 所属院系 | char[20] | “Computer” |  |

6.3 项目

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 团队名称 | char[30] | “网络安全” |  |
| 负责人 | char[12] | “李四” |  |
| 教师人数 | int | 5 |  |
| 研究生人数 | int | 20 |  |
| 所属院系 | char[20] | “Computer” |  |

# 概要设计

## 引言

为了方便管理，各个院系都成立了独立的研究团队，每个团队由若干个教师组成，科研项目的立项、申报、研究、结题全过程都是以团队为单位来开展的。设计一个程序对各个院系的团队信息以及团队的科研项目信息进行有效地管理，以期提高项目管理的效率。

* 1. 编写目的

对某大学各院系科研项目信息进行管理，包括各院系基本信息、科研团队基本信息和科研项目基本信息。

* 1. 背景说明

用户：各院系相关人员

## 总体设计

2,1 需求规定

(1)．数据维护

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据

维护功能，又分为三个子模块：

* 院系基本信息维护：包括对院系基本信息的录入、修改和删除等功能。
* 科研团队基本信息维护：包括对科研团队基本信息的录入、修改和删除等功能。
* 在科研项目基本信息维护：包括对科研项目基本信息的录入、修改和删除等功能。

(2)．数据查询

本模块实现对院系基本信息，科研团队基本信息，及科研项目基本信息等三方面基本信息的数据查询功能，又分为三个子模块。（假设院系负责人以科研团队没有同名的情况）

1）院系信息查询功能

* 以院系负责人为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系负责人

为“张三”的院系基本信息。

* 以院系名称的全部或一部分为条件来查找并显示满足条件的院系基本信息。例如，查找并显示院系名称中包含“comp”的所有院系基本信息。

2）科研团队基本信息查询功能

* 团队名称的全部或一部分为条件查找并显示满足条件的科研团队基本信息。例如，查找并显

示团队名称中包含“网络安全”的科研团队基本信息。

* 以教师人数为条件查找并显示满足条件的科研团队基本信息。例如，查找并显示教师人数大

于 5 人的所有科研团队基本信息。

3）科研项目基本信息查询功能

* 项目编号为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示项目编号为

“0121210044”的科研项目基本信息。

* 以所属团队为条件查找并显示满足条件的科研项目基本信息。例如，查找并显示所属团队为

“网络安全”的所有科研项目基本信息。

(3)．数据统计

本模块实现五个方面的数据统计功能，前四个功能需求已给出，第五个自行设计。

1. 统计各院系教师总数，研究生总数，及研究生与教师的人数比（保留 2 位小数），按学生教师人数比值降序排序后，输出统计结果。
2. 统计某年度各院系科研项目数，973 项目数，863 项目数，及科研总经费，科研项目数降序排序后输出。
3. 统计历年来类别为国家自然科学基金的科研项目数最多的 10 个科研团队，按项目数降序排序后输出科研团队名称，类别为国家自然科学基金的科研项目数，以及项目经费总数。
4. 统计科研项目数和教师人数的比值最高的 5 个科研团队，按比值（保留两位小数）降序排序后，输出科研团队名称，教师人数，科研项目数，项目数和教师人数比值。

有关科研项目其他方面的数据统计。

* 1. 运行环境

存储程序的媒体形式：硬盘

运行条件：winxp以上的pc

* 1. 基本设计概念

基本设计概念：该软件由用户界面，程序，文件三个模块功能。其中用户界面负责人机交互，程序负责完成相应功能。文件负责保存已有数据

## 接口设计

3.1 用户接口

交互界面:

文件：保存文件 载入数据 退出

数据: 对数据进行录入删除修改

查询：按相应的要求查找相关信息

统计：按要求统计数据并打印在屏幕上

* 1. 外部接口

无

* 1. 内部接口

文件与程序接口

用户交互界面和程序接口

程序内各函数调用

3.4

## 系统数据结构设计

* 1. 院系

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 院系名称 | char[20] | “Computer” |  |
| 负责人 | char[12] | “张三” |  |
| 联系电话 | char[15] | “13312345678” |  |

* 1. 团队

|  |  |  |  |
| --- | --- | --- | --- |
| 中文字段名 | 类型及长度 |  | 举例 |
| 团队名称 | char[30] | “网络安全” |  |
| 负责人 | char[12] | “李四” |  |
| 教师人数 | int | 5 |  |
| 研究生人数 | int | 20 |  |
| 所属院系 | char[20] | “Computer” |  |

4.3项目

|  |  |  |
| --- | --- | --- |
| 中文字段名 | 类型及长度 | 举例 |
| 项目编号 | char[15] | “0121210044” |
| 项目类别 | char | ‘1’-‘5’分别表示 973 计划项目、国家自然科学基金项目、863 计划项目、国际合作项目、横向项目 |
| 起始时间 | char[8] | “2015/03” |
| 项目经费 | float | 23.5 单位： 万元 |
| 项目负责人 | char[12] | “王五” |
| 所属团队 | char[30] | “网络安全” |

* 1. 数据间关系

…

head

院系

1

院系

2

院系n

NULL

…

团队

1

基本信息

团队

2

基本信息

团队

n

基本信息

科研项目信息链

1

科研项目信息链

2

科研项目信息链

n

head1

团队

1

信息

团队

2

信息

团队

n

信息

团队

1

项目

1

团队

1

项目

2

团队

1

项目

n

团队

2

项目

1

团队

2

项目

2

团队

2

项目

n

团队

n

项目

1

团队

n

项目

2

团队

n

项目

n

head2

院系

1

院系

2

院系

3

# 详细设计

## 引言

1.1 编写目的

这份软件详细设计说明书主要是在之前建立的软件需求分析和概要分析的基础上解释的。主要着力于解决实现该系统需求的程序模块设计文图。决定各个模块简单接口。各个模块间传递的信息，一级有关数据就够的设计问题

预期读者：开发的程序员

* 1. 定义

学院

|  |
| --- |
| char name[20]; //院系名称 |
|  |

|  |
| --- |
| char person\_in\_charge[12]; //负责人 |
|  |

|  |
| --- |
| char phone[15]; //联系号码 |
|  |

|  |
| --- |
| struct team\* Thead; |
|  |

struct school\* next;

学院数据

|  |
| --- |
| char name[20]; |
|  |

|  |
| --- |
| int Sch\_prof\_total; |
|  |

|  |
| --- |
| int Stu\_Amount; |
|  |

|  |
| --- |
| float Pro2Stu; |
|  |

|  |
| --- |
| int \_973\_num; |
|  |

|  |
| --- |
| int \_863\_num; |
|  |

|  |
| --- |
| int sch\_pro\_num; |
|  |

|  |
| --- |
| float total\_fund; |
|  |

struct sch\_data\* next;

团队

|  |
| --- |
| char name[20]; //团队名称 |
|  |

|  |
| --- |
| char person\_in\_charge[12]; //负责人 |
|  |

|  |
| --- |
| int tea\_amount; //教授人数 |
|  |

|  |
| --- |
| int ug\_amount; //研究生人数 |
|  |

|  |
| --- |
| char school[20]; //所属院系 |
|  |

|  |
| --- |
| struct team \*next; |
|  |

struct subject \*Shead;

团队数据

|  |
| --- |
| char name[30]; |
|  |

|  |
| --- |
| int t\_pro\_amount; |
|  |

|  |
| --- |
| int t\_tea\_amount; |
|  |

|  |
| --- |
| float Pro2Prof; |
|  |

|  |
| --- |
| float t\_total\_fund; |
|  |

|  |
| --- |
| int NA\_pro; |
|  |

struct team\_data\* next;

项目

|  |
| --- |
| char ID[15]; //项目编号 |
|  |

|  |
| --- |
| char type; //range from 1 to 5; |
|  |

|  |
| --- |
| char begin\_time[8]; //起始时间 |
|  |

|  |
| --- |
| float fund; //经费 |
|  |

|  |
| --- |
| char person\_in\_charge[12]; //负责人 |
|  |

|  |
| --- |
| char team[30]; //所属团队 |
|  |

struct subject \*next;

## 程序系统的结构



## 程序设计说明

3.1 程序描述

模块设计各个功能均有函数试下，通过不同的请求实现获取信息增删查改统计的功能

。非常住于内存，是子程序，串行处理

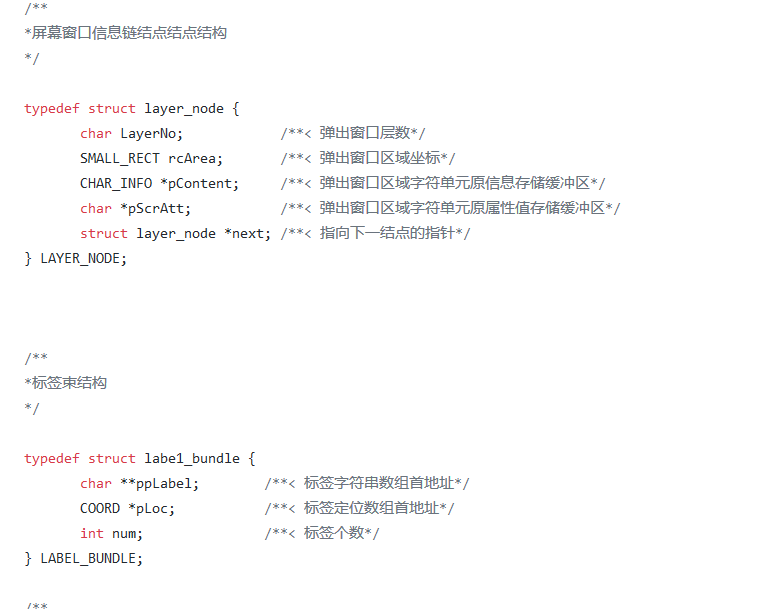
3.2性能

相应时间：任何系统相应时间<0.5s

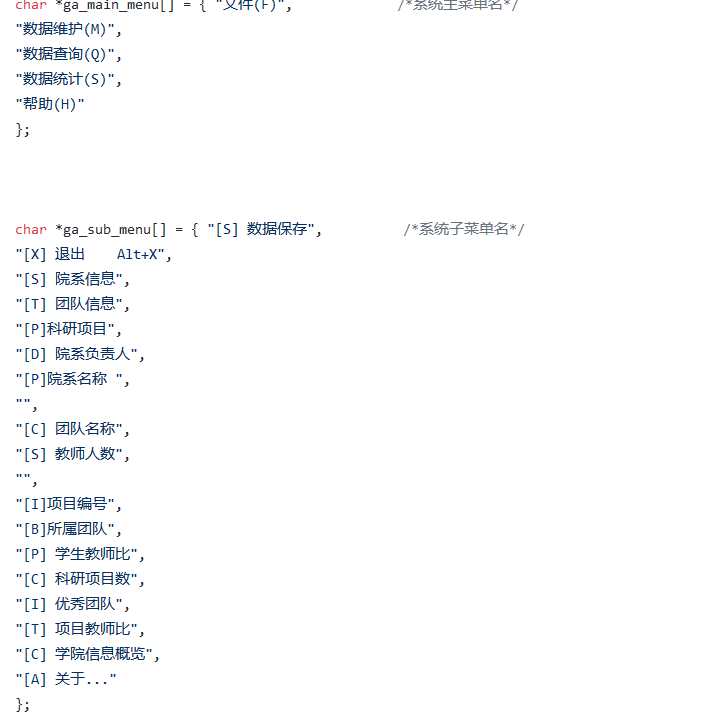
运行时间：全天候

提示信息： 立刻提示

3.3 用户界面



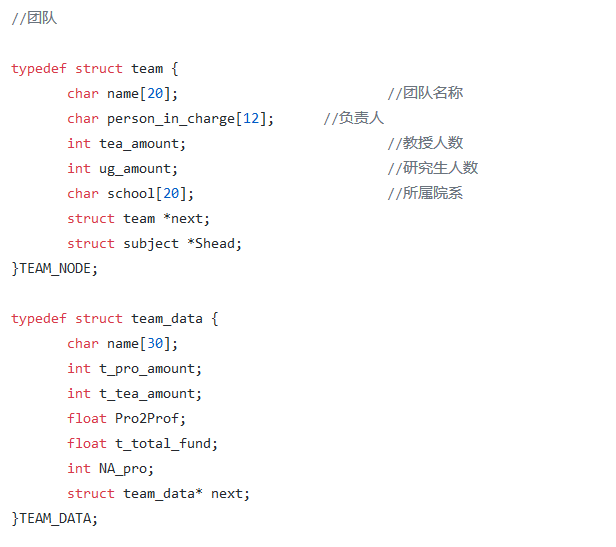




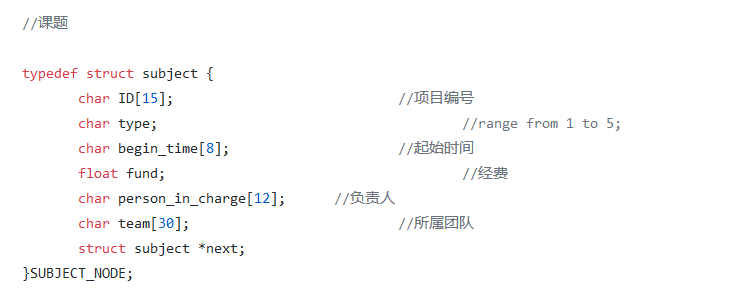
3.4 院系



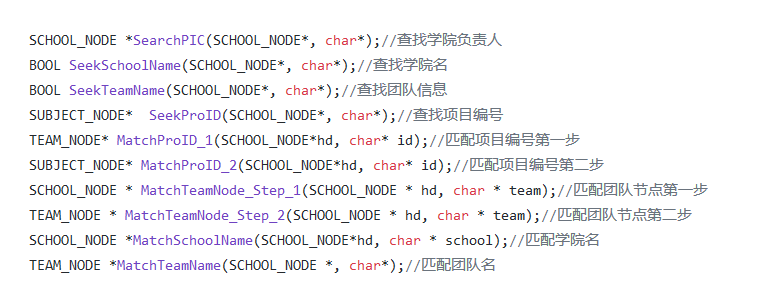
3.6团队



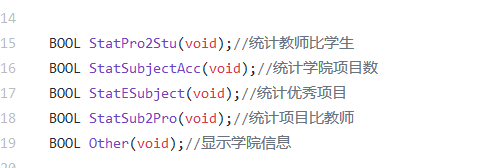
3.7 项目



3.8 查询



3.9 统计



3.10 存储分配

院系信息，团队信息，项目信息存储在三个不同的二进制文件中

3.11 限制条件

Winxp以上操作系统

3.12 测试计划

对测试技术的要求 了解输入规范

预期结果 程序运行良好

# 测试方案

## 1,引言

实行测试计划目的是为了检测程序是否可以顺利运行，并且在运行的过程中找到一些不合理的地方和Bug,减少正式投入后出现错误的可能

## 计划

2.1测试内容

用户界面，信息录入，查询，统计功能

2.2 测试环境

Win10操作系统pc一台

2.3 测试方法

黑盒测试

## 3.测试数据说明

3.1 测试1：读取数据

对于原有数据进行读取分为读取成功和读取失败两种结果

3.2测试2：按不同关键字进行查询

3.3 测试3：对增删查改功能进行测试

3.4 测试4 对保存文件进行测试

## 运行环境

硬件设备 i5 win10 pc

外存 1TB

脱机

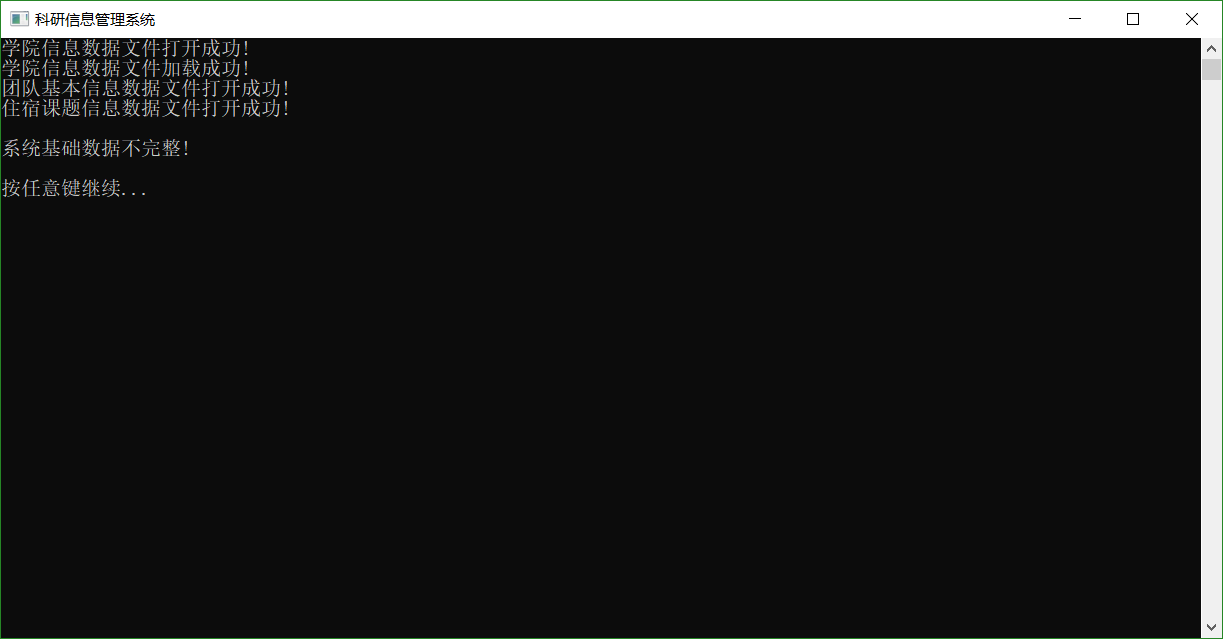
程序语言C++

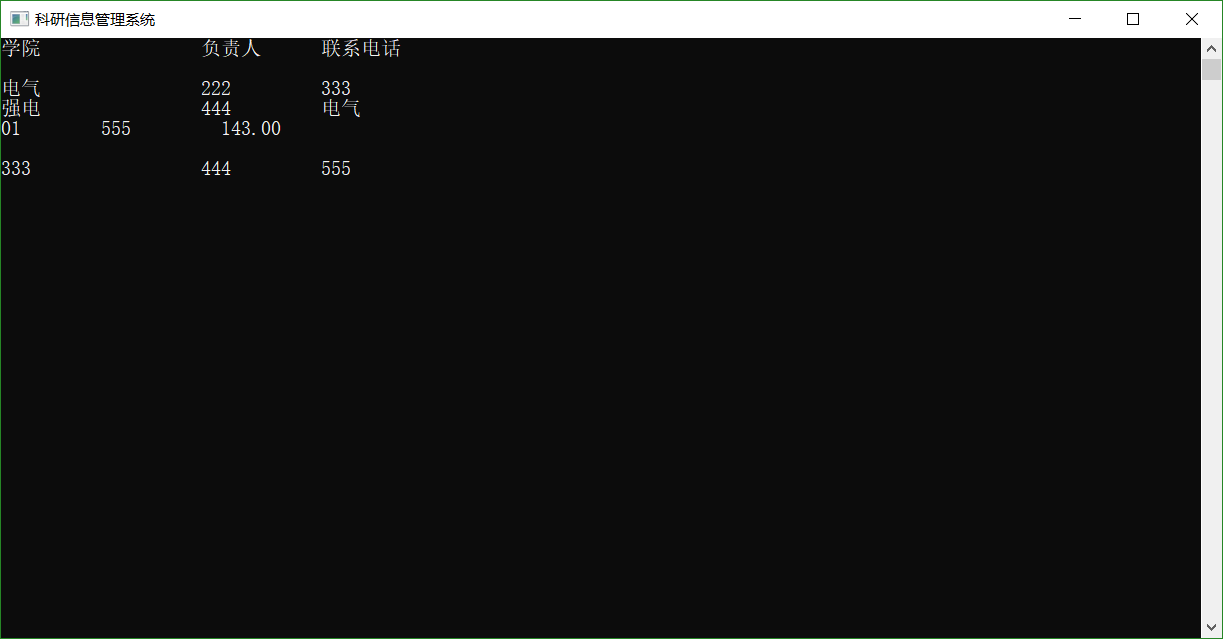
4.1 数据结构

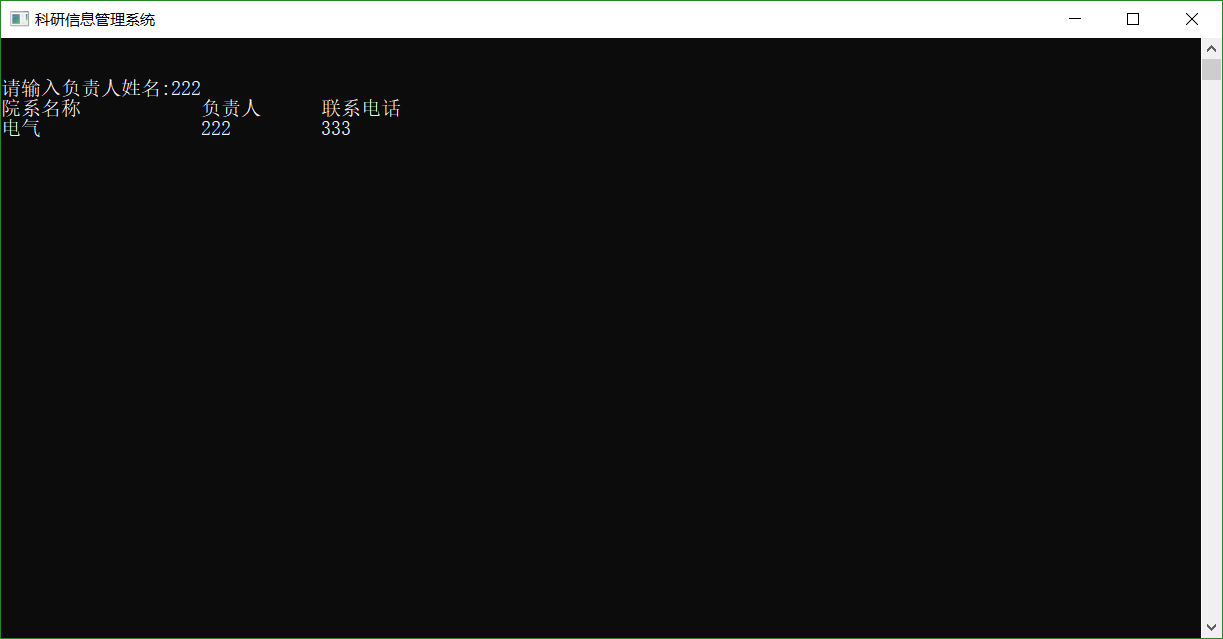


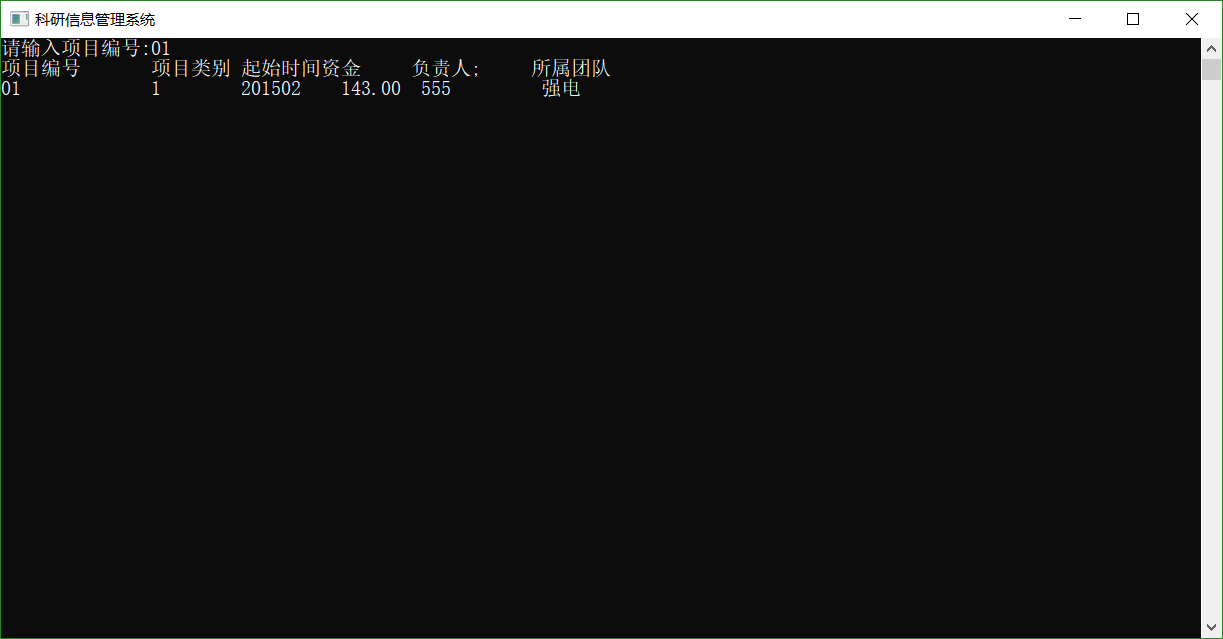


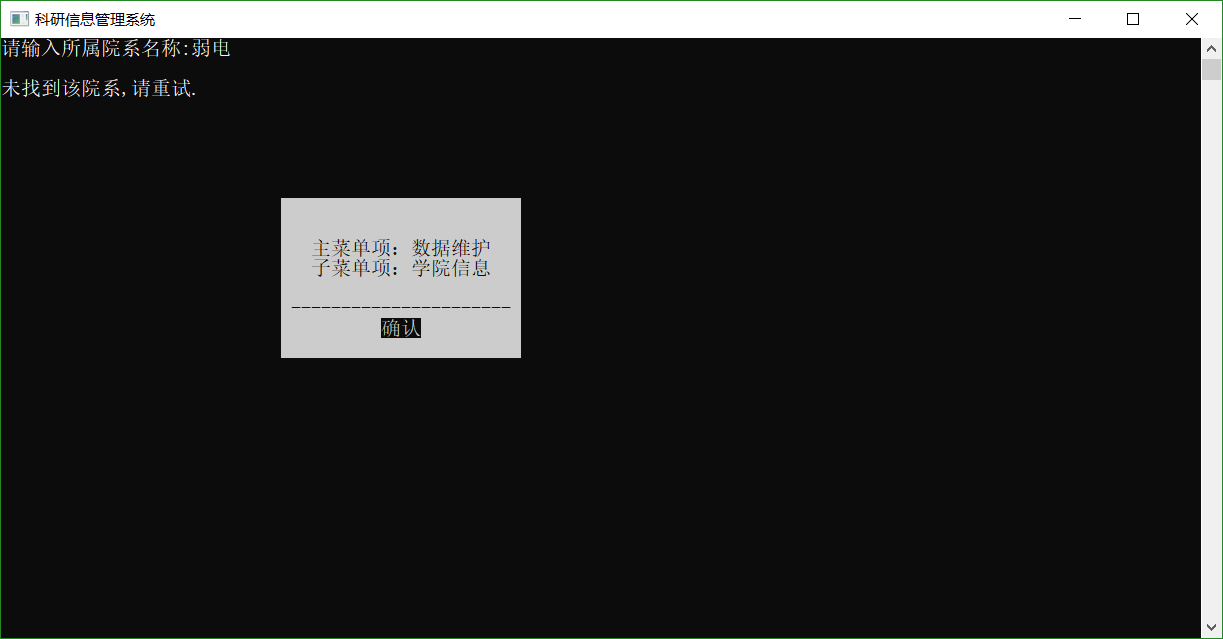
## 使用过程





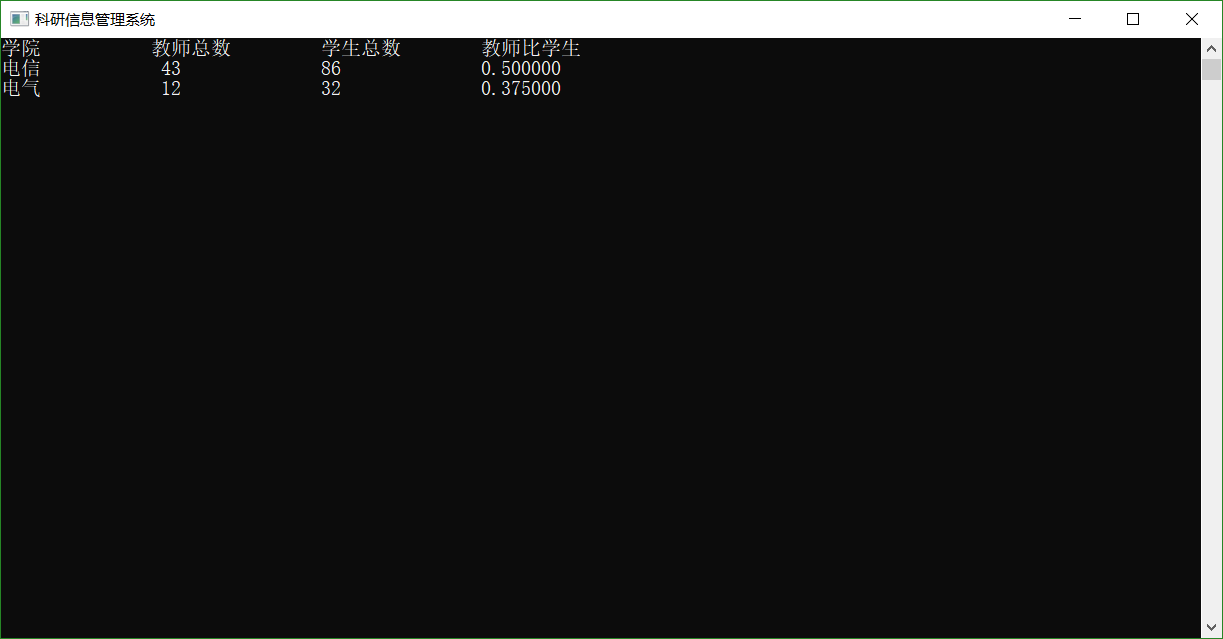


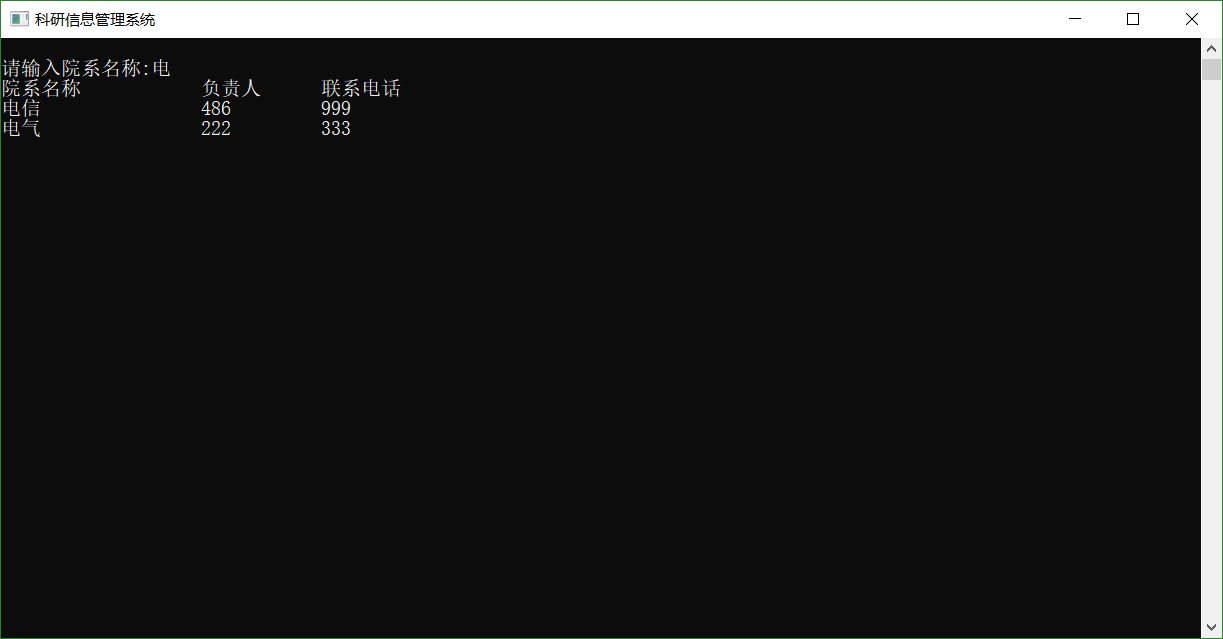








v



## 结论

软件大致能实现相应功能

# 项目管理

根据估算，代码行数约为3000行，所以L=3

E=a\*L^b

软件为组织型其中a=2.4 b=1.05 由此算出E=7.60

D=c\*E^d

其中c为2.5，d为0.38 由此算出 D=5.40又

E/D=1.4

可以卡出项目有一个人开发完成，工期5个月成本7万元

# 七、体会

学会了如何系统的思考，以及养成良好的编码习惯，了解了软件工程的目标、过程和原则：

生产一个最终能满足需求且达到工程目标的软件产品所需要的步骤。软件工程过程主要包括开发过程、运作过程、维护过程。它们覆盖了需求、设计、实现、确认以及维护等活动。需求活动包括问题分析和需求分析。问题分析获取需求定义，又称软件需求规约。需求分析生成功能规约。设计活动一般包括概要设计和详细设计。概要设计建立整个软件系统结构，包括子系统、模块以及相关层次的说明、每一模块的接口定义。详细设计产生程序员可用的模块说明，包括每一模块中数据结构说明及加工描述。实现活动把设计结果转换为可执行的程序代码。确认活动贯穿于整个开发过程，实现完成后的确认，保证最终产品满足用户的要求。维护活动包括使用过程中的扩充、修改与完善。伴随以上过程，还有管理过程、支持过程、培训过程等。

# 八、附录

#include "p.h"

unsigned long ul;

int main()

{

COORD size = { SCR\_COL, SCR\_ROW }; /\*窗口缓冲区大小\*/

gh\_std\_out = GetStdHandle(STD\_OUTPUT\_HANDLE); /\* 获取标准输出设备句柄\*/

gh\_std\_in = GetStdHandle(STD\_INPUT\_HANDLE); /\* 获取标准输入设备句柄\*/

SetConsoleTitle(gp\_sys\_name); /\*设置窗口标题\*/

SetConsoleScreenBufferSize(gh\_std\_out, size); /\*设置窗口缓冲区大小80\*25\*/

LoadData(); /\*数据加载\*/

InitInterface(); /\*界面初始化\*/

RunSys(&gp\_head); /\*系统功能模块的选择及运行\*/

CloseSys(gp\_head); /\*退出系统\*/

return 0;

}

/\*\*

\* 函数名称: LoadData

\* 函数功能: 将代码表和三类基础数据从数据文件载入到内存缓冲区和十字链表中.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: BOOL类型, 功能函数中除了函数ExitSys的返回值可以为FALSE外,

\* 其他函数的返回值必须为TRUE.

\*

\* 调用说明: 为了能够以统一的方式调用各功能函数, 将这些功能函数的原型设为

\* 一致, 即无参数且返回值为BOOL. 返回值为FALSE时, 结束程序运行.

\*/

BOOL LoadData()

{

int Re = 0;

Re = CreatList(&gp\_head);

gc\_sys\_state |= Re;

gc\_sys\_state &= ~(4 + 8 + 16 - Re);

if (gc\_sys\_state < (1 | 2 | 4 | 8 | 16))

{ /\*数据加载提示信息\*/

printf("\n系统基础数据不完整!\n");

printf("\n按任意键继续...\n");

getch();

}

return TRUE;

}

/\*\*

\* 函数名称: LoadCode

\* 函数功能: 将代码表从数据文件载入到内存缓冲区, 并进行排序和去除空格.

\* 输入参数: FileName 存放代码表的数据文件名.

\* 输出参数: pBuffer 指向内存缓冲区的指针变量的地址.

\* 返 回 值: 存放代码表的内存缓冲区大小(以字节为单位).

\*

\* 调用说明:

\*/

int LoadCode(char \*FileName, char \*\*pBuffer)

{

char \*pTemp, \*pStr1, \*pStr2;

int handle;

int BufferLen, len, loc1, loc2, i;

long filelen;

if ((handle = open(FileName, O\_RDONLY | O\_TEXT)) == -1) /\*如果以只读方式打开失败 \*/

{

handle = open(FileName, O\_CREAT | O\_TEXT, S\_IREAD); /\*以创建方式打开\*/

}

filelen = filelength(handle); /\*数据文件的长度\*/

pTemp = (char \*)calloc(filelen + 1, sizeof(char)); /\*申请同样大小的动态存储区\*/

BufferLen = read(handle, pTemp, filelen); /\*将数据文件的内容全部读入到内存\*/

close(handle);

\*(pTemp + BufferLen) = '\0'; /\*在动态存储区尾存一个空字符，作为字符串结束标志\*/

BufferLen++;

for (i = 0; i<BufferLen; i++) /\*将动态存储区中的所有换行符替换成空字符\*/

{

if (\*(pTemp + i) == '\n')

{

\*(pTemp + i) = '\0';

}

}

/\*再申请一块同样大小的动态存储区，用于存放排序后的代码串\*/

\*pBuffer = (char \*)calloc(BufferLen, sizeof(char));

loc2 = 0;

pStr1 = pTemp;

len = strlen(pStr1);

while (BufferLen > len + 1) /\*选择法排序\*/

{

loc1 = len + 1;

while (BufferLen > loc1) /\*每趟找到序列中最小代码串，首地址存入pStr1\*/

{

pStr2 = pTemp + loc1;

if (strcmp(pStr1, pStr2) > 0)

{

pStr1 = pStr2;

}

loc1 += strlen(pStr2) + 1;

}

len = strlen(pStr1); /\*这一趟所找到的最小代码串长度\*/

/\*如果不是空串，则进行复制，loc2是下一个最小代码串存放地址的偏移量\*/

if (len > 0)

{

strcpy(\*pBuffer + loc2, pStr1);

loc2 += len + 1; /\*已复制的代码串所占存储空间大小\*/

}

/\*将最小代码串从序列中删除掉\*/

for (i = 0; i<BufferLen - (pStr1 - pTemp) - (len + 1); i++)

{

\*(pStr1 + i) = \*(pStr1 + i + len + 1);

}

BufferLen -= len + 1; /\*下一趟排序所处理序列的长度\*/

pStr1 = pTemp; /\*假定序列的第一个代码串为最小代码串\*/

len = strlen(pStr1);

} /\*序列中只剩下一个代码串时，排序结束\*/

/\*复制最后这个代码串\*/

len = strlen(pStr1);

strcpy(\*pBuffer + loc2, pStr1);

/\*修改动态存储区大小，使其正好放下排序后代码串\*/

loc2 += len + 1;

\*pBuffer = (char \*)realloc(\*pBuffer, loc2);

free(pTemp); /\*释放最先申请的动态存储区\*/

return loc2; /\*返回存放代码串的内存缓冲区实际大小\*/

}

/\*\*

\* 函数名称: CreatList

\* 函数功能: 从数据文件读取基础数据, 并存放到所创建的十字链表中.

\* 输入参数: 无

\* 输出参数: phead 主链头指针的地址, 用来返回所创建的十字链.

\* 返 回 值: int型数值, 表示链表创建的情况.

\* 0 空链, 无数据

\* 4 已加载学院信息数据，无团队基本信息和课题信息数据

\* 12 已加载学院信息和团队基本信息数据，无课题信息数据

\* 28 三类基础数据都已加载

\*

\* 调用说明:

\*/

int CreatList(SCHOOL\_NODE \*\*phead)

{

SCHOOL\_NODE \*hd = NULL, \*pSchNode, tmp1;

TEAM\_NODE \*pTeamNode, tmp2;

SUBJECT\_NODE \*pSubjectNode, tmp3;

FILE \*pFile;

int find;

int re = 0;

if ((pFile = fopen(gp\_school\_info\_filename, "rb")) == NULL)

{

printf("学院信息数据文件打开失败!\n");

return re;

}

printf("学院信息数据文件打开成功!\n");

/\*从数据文件中读学院信息数据，存入以后进先出方式建立的主链中\*/

while (fread(&tmp1, sizeof(SCHOOL\_NODE), 1, pFile) == 1)

{

pSchNode = (SCHOOL\_NODE \*)malloc(sizeof(SCHOOL\_NODE));

\*pSchNode = tmp1;

pSchNode->Thead = NULL;

pSchNode->next = hd;

hd = pSchNode;

}

fclose(pFile);

if (hd == NULL)

{

printf("学院信息数据文件加载失败!\n");

return re;

}

printf("学院信息数据文件加载成功!\n");

\*phead = hd;

re += 4;

if ((pFile = fopen(gp\_team\_info\_filename, "rb")) == NULL)

{

printf("团队基本信息数据文件打开失败!\n");

return re;

}

printf("团队基本信息数据文件打开成功!\n");

re += 8;

/\*从数据文件中读取团队基本信息数据，存入主链对应结点的团队基本信息支链中\*/

while (fread(&tmp2, sizeof(TEAM\_NODE), 1, pFile) == 1)

{

/\*创建结点，存放从数据文件中读出的团队基本信息\*/

pTeamNode = (TEAM\_NODE \*)malloc(sizeof(TEAM\_NODE));

\*pTeamNode = tmp2;

pTeamNode->Shead = NULL;

/\*在主链上查找该团队所住学院对应的主链结点\*/

pSchNode = hd;

while (pSchNode != NULL

&& strcmp(pSchNode->name, pTeamNode->school) != 0)

{

pSchNode = pSchNode->next;

}

if (pSchNode != NULL) /\*如果找到，则将结点以后进先出方式插入团队信息支链\*/

{

pTeamNode->next = pSchNode->Thead;

pSchNode->Thead = pTeamNode;

}

else /\*如果未找到，则释放所创建结点的内存空间\*/

{

free(pTeamNode);

}

}

fclose(pFile);

if ((pFile = fopen(gp\_subject\_info\_filename, "rb")) == NULL)

{

printf("住宿课题信息数据文件打开失败!\n");

return re;

}

printf("住宿课题信息数据文件打开成功!\n");

re += 16;

/\*从数据文件中读取团队课题信息数据，存入团队基本信息支链对应结点的课题支链中\*/

while (fread(&tmp3, sizeof(SUBJECT\_NODE), 1, pFile) == 1)

{

/\*创建结点，存放从数据文件中读出的团队课题信息\*/

pSubjectNode = (SUBJECT\_NODE \*)malloc(sizeof(SUBJECT\_NODE));

\*pSubjectNode = tmp3;

/\*查找团队信息支链上对应团队信息结点\*/

pSchNode = hd;

find = 0;

while (pSchNode != NULL && find == 0)

{

pTeamNode = pSchNode->Thead;

while (pTeamNode != NULL && find == 0)

{

if (strcmp(pTeamNode->name, pSubjectNode->team) == 0)

{

find = 1;

break;

}

pTeamNode = pTeamNode->next;

}

pSchNode = pSchNode->next;

}

if (find) /\*如果找到，则将结点以后进先出方式插入团队课题信息支链中\*/

{

pSubjectNode->next = pTeamNode->Shead;

pTeamNode->Shead = pSubjectNode;

}

else /\*如果未找到，则释放所创建结点的内存空间\*/

{

free(pSubjectNode);

}

}

fclose(pFile);

return re;

}

/\*\*

\* 函数名称: InitInterface

\* 函数功能: 初始化界面.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void InitInterface()

{

WORD att = FOREGROUND\_INTENSITY

| FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE;/\*白色前景和绿色背景\*/

SetConsoleTextAttribute(gh\_std\_out, att); /\*设置控制台屏幕缓冲区字符属性\*/

ClearScreen(); /\* 清屏\*/

/\*创建弹出窗口信息堆栈，将初始化后的屏幕窗口当作第一层弹出窗口\*/

gp\_scr\_att = (char \*)calloc(SCR\_COL \* SCR\_ROW, sizeof(char));/\*屏幕字符属性\*/

gp\_top\_layer = (LAYER\_NODE \*)malloc(sizeof(LAYER\_NODE));

gp\_top\_layer->LayerNo = 0; /\*弹出窗口的层号为0\*/

gp\_top\_layer->rcArea.Left = 0; /\*弹出窗口的区域为整个屏幕窗口\*/

gp\_top\_layer->rcArea.Top = 0;

gp\_top\_layer->rcArea.Right = SCR\_COL - 1;

gp\_top\_layer->rcArea.Bottom = SCR\_ROW - 1;

gp\_top\_layer->pContent = NULL;

gp\_top\_layer->pScrAtt = gp\_scr\_att;

gp\_top\_layer->next = NULL;

ShowMenu(); /\*显示菜单栏\*/

ShowState(); /\*显示状态栏\*/

return;

}

/\*\*

\* 函数名称: ClearScreen

\* 函数功能: 清除屏幕信息.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void ClearScreen(void)

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

COORD home = { 0, 0 };

unsigned long size;

GetConsoleScreenBufferInfo(gh\_std\_out, &bInfo);/\*取屏幕缓冲区信息\*/

size = bInfo.dwSize.X \* bInfo.dwSize.Y; /\*计算屏幕缓冲区字符单元数\*/

/\*将屏幕缓冲区所有单元的字符属性设置为当前屏幕缓冲区字符属性\*/

FillConsoleOutputAttribute(gh\_std\_out, bInfo.wAttributes, size, home, &ul);

/\*将屏幕缓冲区所有单元填充为空格字符\*/

FillConsoleOutputCharacter(gh\_std\_out, ' ', size, home, &ul);

return;

}

/\*\*

\* 函数名称: ShowMenu

\* 函数功能: 在屏幕上显示主菜单, 并设置热区, 在主菜单第一项上置选中标记.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void ShowMenu()

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

CONSOLE\_CURSOR\_INFO lpCur;

COORD size;

COORD pos = { 0, 0 };

int i, j;

int PosA = 2, PosB;

char ch;

GetConsoleScreenBufferInfo(gh\_std\_out, &bInfo);

size.X = bInfo.dwSize.X;

size.Y = 1;

SetConsoleCursorPosition(gh\_std\_out, pos);

for (i = 0; i < 5; i++) /\*在窗口第一行第一列处输出主菜单项\*/

{

printf(" %s ", ga\_main\_menu[i]);

}

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = FALSE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur); /\*隐藏光标\*/

/\*申请动态存储区作为存放菜单条屏幕区字符信息的缓冲区\*/

gp\_buff\_menubar\_info = (CHAR\_INFO \*)malloc(size.X \* size.Y \* sizeof(CHAR\_INFO));

SMALL\_RECT rcMenu = { 0, 0, size.X - 1, 0 };

/\*将窗口第一行的内容读入到存放菜单条屏幕区字符信息的缓冲区中\*/

ReadConsoleOutput(gh\_std\_out, gp\_buff\_menubar\_info, size, pos, &rcMenu);

/\*将这一行中英文字母置为红色，其他字符单元置为白底黑字\*/

for (i = 0; i<size.X; i++)

{

(gp\_buff\_menubar\_info + i)->Attributes = BACKGROUND\_BLUE | BACKGROUND\_GREEN

| BACKGROUND\_RED;

ch = (char)((gp\_buff\_menubar\_info + i)->Char.AsciiChar);

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_menubar\_info + i)->Attributes |= FOREGROUND\_RED;

}

}

/\*修改后的菜单条字符信息回写到窗口的第一行\*/

WriteConsoleOutput(gh\_std\_out, gp\_buff\_menubar\_info, size, pos, &rcMenu);

COORD endPos = { 0, 1 };

SetConsoleCursorPosition(gh\_std\_out, endPos); /\*将光标位置设置在第2行第1列\*/

/\*将菜单项置为热区，热区编号为菜单项号，热区类型为0(按钮型)\*/

i = 0;

do

{

PosB = PosA + strlen(ga\_main\_menu[i]); /\*定位第i+1号菜单项的起止位置\*/

for (j = PosA; j<PosB; j++)

{

gp\_scr\_att[j] |= (i + 1) << 2; /\*设置菜单项所在字符单元的属性值\*/

}

PosA = PosB + 4;

i++;

} while (i<5);

TagMainMenu(gi\_sel\_menu); /\*在选中主菜单项上做标记，gi\_sel\_menu初值为1\*/

return;

}

/\*\*

\* 函数名称: ShowState

\* 函数功能: 显示状态条.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明: 状态条字符属性为白底黑字, 初始状态无状态信息.

\*/

void ShowState()

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

COORD size;

COORD pos = { 0, 0 };

int i;

GetConsoleScreenBufferInfo(gh\_std\_out, &bInfo);

size.X = bInfo.dwSize.X;

size.Y = 1;

SMALL\_RECT rcMenu = { 0, bInfo.dwSize.Y - 1, size.X - 1, bInfo.dwSize.Y - 1 };

if (gp\_buff\_stateBar\_info == NULL)

{

gp\_buff\_stateBar\_info = (CHAR\_INFO \*)malloc(size.X \* size.Y \* sizeof(CHAR\_INFO));

ReadConsoleOutput(gh\_std\_out, gp\_buff\_stateBar\_info, size, pos, &rcMenu);

}

for (i = 0; i<size.X; i++)

{

(gp\_buff\_stateBar\_info + i)->Attributes = BACKGROUND\_BLUE | BACKGROUND\_GREEN

| BACKGROUND\_RED;

/\*

ch = (char)((gp\_buff\_stateBar\_info+i)->Char.AsciiChar);

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_stateBar\_info+i)->Attributes |= FOREGROUND\_RED;

}

\*/

}

WriteConsoleOutput(gh\_std\_out, gp\_buff\_stateBar\_info, size, pos, &rcMenu);

return;

}

/\*\*

\* 函数名称: TagMainMenu

\* 函数功能: 在指定主菜单项上置选中标志.

\* 输入参数: num 选中的主菜单项号

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void TagMainMenu(int num)

{

CONSOLE\_SCREEN\_BUFFER\_INFO bInfo;

COORD size;

COORD pos = { 0, 0 };

int PosA = 2, PosB;

char ch;

int i;

if (num == 0) /\*num为0时，将会去除主菜单项选中标记\*/

{

PosA = 0;

PosB = 0;

}

else /\*否则，定位选中主菜单项的起止位置: PosA为起始位置, PosB为截止位置\*/

{

for (i = 1; i<num; i++)

{

PosA += strlen(ga\_main\_menu[i - 1]) + 4;

}

PosB = PosA + strlen(ga\_main\_menu[num - 1]);

}

GetConsoleScreenBufferInfo(gh\_std\_out, &bInfo);

size.X = bInfo.dwSize.X;

size.Y = 1;

/\*去除选中菜单项前面的菜单项选中标记\*/

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

{

(gp\_buff\_menubar\_info + i)->Attributes = BACKGROUND\_BLUE | BACKGROUND\_GREEN

| BACKGROUND\_RED;

ch = (gp\_buff\_menubar\_info + i)->Char.AsciiChar;

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_menubar\_info + i)->Attributes |= FOREGROUND\_RED;

}

}

/\*在选中菜单项上做标记，黑底白字\*/

for (i = PosA; i<PosB; i++)

{

(gp\_buff\_menubar\_info + i)->Attributes = FOREGROUND\_BLUE | FOREGROUND\_GREEN

| FOREGROUND\_RED;

}

/\*去除选中菜单项后面的菜单项选中标记\*/

for (i = PosB; i<bInfo.dwSize.X; i++)

{

(gp\_buff\_menubar\_info + i)->Attributes = BACKGROUND\_BLUE | BACKGROUND\_GREEN

| BACKGROUND\_RED;

ch = (char)((gp\_buff\_menubar\_info + i)->Char.AsciiChar);

if ((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

(gp\_buff\_menubar\_info + i)->Attributes |= FOREGROUND\_RED;

}

}

/\*将做好标记的菜单条信息写到窗口第一行\*/

SMALL\_RECT rcMenu = { 0, 0, size.X - 1, 0 };

WriteConsoleOutput(gh\_std\_out, gp\_buff\_menubar\_info, size, pos, &rcMenu);

return;

}

/\*\*

\* 函数名称: CloseSys

\* 函数功能: 关闭系统.

\* 输入参数: hd 主链头指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void CloseSys(SCHOOL\_NODE \*hd)

{

SCHOOL\_NODE \*pSchNode1 = hd, \*pSchNode2;

TEAM\_NODE \*pTeamNode1, \*pTeamNode2;

SUBJECT\_NODE \*pSubjectNode1, \*pSubjectNode2;

while (pSchNode1 != NULL) /\*释放十字交叉链表的动态存储区\*/

{

pSchNode2 = pSchNode1->next;

pTeamNode1 = pSchNode1->Thead;

while (pTeamNode1 != NULL) /\*释放团队基本信息支链的动态存储区\*/

{

pTeamNode2 = pTeamNode1->next;

pSubjectNode1 = pTeamNode1->Shead;

while (pSubjectNode1 != NULL) /\*释放课题信息支链的动态存储区\*/

{

pSubjectNode2 = pSubjectNode1->next;

free(pSubjectNode1);

pSubjectNode1 = pSubjectNode2;

}

free(pTeamNode1);

pTeamNode1 = pTeamNode2;

}

free(pSchNode1); /\*释放主链结点的动态存储区\*/

pSchNode1 = pSchNode2;

}

ClearScreen(); /\*清屏\*/

/\*释放存放菜单条、状态条、性别代码和团队类别代码等信息动态存储区\*/

free(gp\_buff\_menubar\_info);

free(gp\_buff\_stateBar\_info);

free(gp\_sub\_code);

//free(gp\_type\_code);

/\*关闭标准输入和输出设备句柄\*/

CloseHandle(gh\_std\_out);

CloseHandle(gh\_std\_in);

/\*将窗口标题栏置为运行结束\*/

SetConsoleTitle("运行结束");

return;

}

/\*\*

\* 函数名称: RunSys

\* 函数功能: 运行系统, 在系统主界面下运行用户所选择的功能模块.

\* 输入参数: 无

\* 输出参数: phead 主链头指针的地址

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void RunSys(SCHOOL\_NODE \*\*phead)

{

INPUT\_RECORD inRec;

DWORD res;

COORD pos = { 0, 0 };

BOOL bRet = TRUE;

int i, loc, num;

int cNo, cAtt; /\*cNo:字符单元层号, cAtt:字符单元属性\*/

char vkc, asc; /\*vkc:虚拟键代码, asc:字符的ASCII码值\*/

while (bRet)

{

/\*从控制台输入缓冲区中读一条记录\*/

ReadConsoleInput(gh\_std\_in, &inRec, 1, &res);

if (inRec.EventType == MOUSE\_EVENT) /\*如果记录由鼠标事件产生\*/

{

pos = inRec.Event.MouseEvent.dwMousePosition; /\*获取鼠标坐标位置\*/

cNo = gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] & 3; /\*取该位置的层号\*/

cAtt = gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] >> 2;/\*取该字符单元属性\*/

if (cNo == 0) /\*层号为0，表明该位置未被弹出子菜单覆盖\*/

{

/\* cAtt > 0 表明该位置处于热区(主菜单项字符单元)

\* cAtt != gi\_sel\_menu 表明该位置的主菜单项未被选中

\* gp\_top\_layer->LayerNo > 0 表明当前有子菜单弹出

\*/

if (cAtt > 0 && cAtt != gi\_sel\_menu && gp\_top\_layer->LayerNo > 0)

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

PopMenu(cAtt); /\*弹出鼠标所在主菜单项对应的子菜单\*/

}

}

else if (cAtt > 0) /\*鼠标所在位置为弹出子菜单的菜单项字符单元\*/

{

TagSubMenu(cAtt); /\*在该子菜单项上做选中标记\*/

}

if (inRec.Event.MouseEvent.dwButtonState

== FROM\_LEFT\_1ST\_BUTTON\_PRESSED) /\*如果按下鼠标左边第一键\*/

{

if (cNo == 0) /\*层号为0，表明该位置未被弹出子菜单覆盖\*/

{

if (cAtt > 0) /\*如果该位置处于热区(主菜单项字符单元)\*/

{

PopMenu(cAtt); /\*弹出鼠标所在主菜单项对应的子菜单\*/

}

/\*如果该位置不属于主菜单项字符单元，且有子菜单弹出\*/

else if (gp\_top\_layer->LayerNo > 0)

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

}

}

else /\*层号不为0，表明该位置被弹出子菜单覆盖\*/

{

if (cAtt > 0) /\*如果该位置处于热区(子菜单项字符单元)\*/

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

/\*执行对应功能函数:gi\_sel\_menu主菜单项号,cAtt子菜单项号\*/

bRet = ExeFunction(gi\_sel\_menu, cAtt);

}

}

}

else if (inRec.Event.MouseEvent.dwButtonState

== RIGHTMOST\_BUTTON\_PRESSED) /\*如果按下鼠标右键\*/

{

if (cNo == 0) /\*层号为0，表明该位置未被弹出子菜单覆盖\*/

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

}

}

}

else if (inRec.EventType == KEY\_EVENT /\*如果记录由按键产生\*/

&& inRec.Event.KeyEvent.bKeyDown) /\*且键被按下\*/

{

vkc = inRec.Event.KeyEvent.wVirtualKeyCode; /\*获取按键的虚拟键码\*/

asc = inRec.Event.KeyEvent.uChar.AsciiChar; /\*获取按键的ASC码\*/

/\*系统快捷键的处理\*/

if (vkc == 112) /\*如果按下F1键\*/

{

if (gp\_top\_layer->LayerNo != 0) /\*如果当前有子菜单弹出\*/

{

PopOff(); /\*关闭弹出的子菜单\*/

gi\_sel\_sub\_menu = 0; /\*将选中子菜单项的项号置为0\*/

}

bRet = ExeFunction(5, 1); /\*运行帮助主题功能函数\*/

}

else if (inRec.Event.KeyEvent.dwControlKeyState

& (LEFT\_ALT\_PRESSED | RIGHT\_ALT\_PRESSED))

{ /\*如果按下左或右Alt键\*/

switch (vkc) /\*判断组合键Alt+字母\*/

{

case 88: /\*Alt+X 退出\*/

if (gp\_top\_layer->LayerNo != 0)

{

PopOff();

gi\_sel\_sub\_menu = 0;

}

bRet = ExeFunction(1, 4);

break;

case 70: /\*Alt+F\*/

PopMenu(1);

break;

case 77: /\*Alt+M\*/

PopMenu(2);

break;

case 81: /\*Alt+Q\*/

PopMenu(3);

break;

case 83: /\*Alt+S\*/

PopMenu(4);

break;

case 72: /\*Alt+H\*/

PopMenu(5);

break;

}

}

else if (asc == 0) /\*其他控制键的处理\*/

{

if (gp\_top\_layer->LayerNo == 0) /\*如果未弹出子菜单\*/

{

switch (vkc) /\*处理方向键(左、右、下)，不响应其他控制键\*/

{

case 37:

gi\_sel\_menu--;

if (gi\_sel\_menu == 0)

{

gi\_sel\_menu = 5;

}

TagMainMenu(gi\_sel\_menu);

break;

case 39:

gi\_sel\_menu++;

if (gi\_sel\_menu == 6)

{

gi\_sel\_menu = 1;

}

TagMainMenu(gi\_sel\_menu);

break;

case 40:

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

}

}

else /\*已弹出子菜单时\*/

{

for (loc = 0, i = 1; i<gi\_sel\_menu; i++)

{

loc += ga\_sub\_menu\_count[i - 1];

} /\*计算该子菜单中的第一项在子菜单字符串数组中的位置(下标)\*/

switch (vkc) /\*方向键(左、右、上、下)的处理\*/

{

case 37:

gi\_sel\_menu--;

if (gi\_sel\_menu < 1)

{

gi\_sel\_menu = 5;

}

TagMainMenu(gi\_sel\_menu);

PopOff();

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

case 38:

num = gi\_sel\_sub\_menu - 1;

if (num < 1)

{

num = ga\_sub\_menu\_count[gi\_sel\_menu - 1];

}

if (strlen(ga\_sub\_menu[loc + num - 1]) == 0)

{

num--;

}

TagSubMenu(num);

break;

case 39:

gi\_sel\_menu++;

if (gi\_sel\_menu > 5)

{

gi\_sel\_menu = 1;

}

TagMainMenu(gi\_sel\_menu);

PopOff();

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

case 40:

num = gi\_sel\_sub\_menu + 1;

if (num > ga\_sub\_menu\_count[gi\_sel\_menu - 1])

{

num = 1;

}

if (strlen(ga\_sub\_menu[loc + num - 1]) == 0)

{

num++;

}

TagSubMenu(num);

break;

}

}

}

else if ((asc - vkc == 0) || (asc - vkc == 32)) { /\*按下普通键\*/

if (gp\_top\_layer->LayerNo == 0) /\*如果未弹出子菜单\*/

{

switch (vkc)

{

case 70: /\*f或F\*/

PopMenu(1);

break;

case 77: /\*m或M\*/

PopMenu(2);

break;

case 81: /\*q或Q\*/

PopMenu(3);

break;

case 83: /\*s或S\*/

PopMenu(4);

break;

case 72: /\*h或H\*/

PopMenu(5);

break;

case 13: /\*回车\*/

PopMenu(gi\_sel\_menu);

TagSubMenu(1);

break;

}

}

else /\*已弹出子菜单时的键盘输入处理\*/

{

if (vkc == 27) /\*如果按下ESC键\*/

{

PopOff();

gi\_sel\_sub\_menu = 0;

}

else if (vkc == 13) /\*如果按下回车键\*/

{

num = gi\_sel\_sub\_menu;

PopOff();

gi\_sel\_sub\_menu = 0;

bRet = ExeFunction(gi\_sel\_menu, num);

}

else /\*其他普通键的处理\*/

{

/\*计算该子菜单中的第一项在子菜单字符串数组中的位置(下标)\*/

for (loc = 0, i = 1; i<gi\_sel\_menu; i++)

{

loc += ga\_sub\_menu\_count[i - 1];

}

/\*依次与当前子菜单中每一项的代表字符进行比较\*/

for (i = loc; i<loc + ga\_sub\_menu\_count[gi\_sel\_menu - 1]; i++)

{

if (strlen(ga\_sub\_menu[i])>0 && vkc == ga\_sub\_menu[i][1])

{ /\*如果匹配成功\*/

PopOff();

gi\_sel\_sub\_menu = 0;

bRet = ExeFunction(gi\_sel\_menu, i - loc + 1);

}

}

}

}

}

}

}

}

/\*\*

\* 函数名称: PopMenu

\* 函数功能: 弹出指定主菜单项对应的子菜单.

\* 输入参数: num 指定的主菜单项号

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void PopMenu(int num)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

char \*pCh;

int i, j, loc = 0;

if (num != gi\_sel\_menu) /\*如果指定主菜单不是已选中菜单\*/

{

if (gp\_top\_layer->LayerNo != 0) /\*如果此前已有子菜单弹出\*/

{

PopOff();

gi\_sel\_sub\_menu = 0;

}

}

else if (gp\_top\_layer->LayerNo != 0) /\*若已弹出该子菜单，则返回\*/

{

return;

}

gi\_sel\_menu = num; /\*将选中主菜单项置为指定的主菜单项\*/

TagMainMenu(gi\_sel\_menu); /\*在选中的主菜单项上做标记\*/

LocSubMenu(gi\_sel\_menu, &rcPop); /\*计算弹出子菜单的区域位置, 存放在rcPop中\*/

/\*计算该子菜单中的第一项在子菜单字符串数组中的位置(下标)\*/

for (i = 1; i<gi\_sel\_menu; i++)

{

loc += ga\_sub\_menu\_count[i - 1];

}

/\*将该组子菜单项项名存入标签束结构变量\*/

labels.ppLabel = ga\_sub\_menu + loc; /\*标签束第一个标签字符串的地址\*/

labels.num = ga\_sub\_menu\_count[gi\_sel\_menu - 1]; /\*标签束中标签字符串的个数\*/

COORD aLoc[100];/\*定义一个坐标数组，存放每个标签字符串输出位置的坐标\*/

for (i = 0; i<labels.num; i++) /\*确定标签字符串的输出位置，存放在坐标数组中\*/

{

aLoc[i].X = rcPop.Left + 2;

aLoc[i].Y = rcPop.Top + i + 1;

}

labels.pLoc = aLoc; /\*使标签束结构变量labels的成员pLoc指向坐标数组的首元素\*/

/\*设置热区信息\*/

areas.num = labels.num; /\*热区的个数，等于标签的个数，即子菜单的项数\*/

SMALL\_RECT aArea[100];// [areas.num]; /\*定义数组存放所有热区位置\*/

char aSort[100];// [areas.num]; /\*定义数组存放所有热区对应类别\*/

char aTag[100];// [areas.num]; /\*定义数组存放每个热区的编号\*/

for (i = 0; i<areas.num; i++)

{

aArea[i].Left = rcPop.Left + 2; /\*热区定位\*/

aArea[i].Top = rcPop.Top + i + 1;

aArea[i].Right = rcPop.Right - 2;

aArea[i].Bottom = aArea[i].Top;

aSort[i] = 0; /\*热区类别都为0(按钮型)\*/

aTag[i] = i + 1; /\*热区按顺序编号\*/

}

areas.pArea = aArea;/\*使热区结构变量areas的成员pArea指向热区位置数组首元素\*/

areas.pSort = aSort;/\*使热区结构变量areas的成员pSort指向热区类别数组首元素\*/

areas.pTag = aTag; /\*使热区结构变量areas的成员pTag指向热区编号数组首元素\*/

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

PopUp(&rcPop, att, &labels, &areas);

DrawBox(&rcPop); /\*给弹出窗口画边框\*/

pos.X = rcPop.Left + 2;

for (pos.Y = rcPop.Top + 1; pos.Y<rcPop.Bottom; pos.Y++)

{ /\*此循环用来在空串子菜项位置画线形成分隔，并取消此菜单项的热区属性\*/

pCh = ga\_sub\_menu[loc + pos.Y - rcPop.Top - 1];

if (strlen(pCh) == 0) /\*串长为0，表明为空串\*/

{ /\*首先画横线\*/

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right - rcPop.Left - 3, pos, &ul);

for (j = rcPop.Left + 2; j<rcPop.Right - 1; j++)

{ /\*取消该区域字符单元的热区属性\*/

gp\_scr\_att[pos.Y\*SCR\_COL + j] &= 3; /\*按位与的结果保留了低两位\*/

}

}

}

/\*将子菜单项的功能键设为白底红字\*/

pos.X = rcPop.Left + 3;

att = FOREGROUND\_RED | BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED;

for (pos.Y = rcPop.Top + 1; pos.Y<rcPop.Bottom; pos.Y++)

{

if (strlen(ga\_sub\_menu[loc + pos.Y - rcPop.Top - 1]) == 0)

{

continue; /\*跳过空串\*/

}

FillConsoleOutputAttribute(gh\_std\_out, att, 1, pos, &ul);

}

return;

}

/\*\*

\* 函数名称: PopUp

\* 函数功能: 在指定区域输出弹出窗口信息, 同时设置热区, 将弹出窗口位置信息入栈.

\* 输入参数: pRc 弹出窗口位置数据存放的地址

\* att 弹出窗口区域字符属性

\* pLabel 弹出窗口中标签束信息存放的地址

pHotArea 弹出窗口中热区信息存放的地址

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void PopUp(SMALL\_RECT \*pRc, WORD att, LABEL\_BUNDLE \*pLabel, HOT\_AREA \*pHotArea)

{

LAYER\_NODE \*nextLayer;

COORD size;

COORD pos = { 0, 0 };

char \*pCh;

int i, j, row;

/\*弹出窗口所在位置字符单元信息入栈\*/

size.X = pRc->Right - pRc->Left + 1; /\*弹出窗口的宽度\*/

size.Y = pRc->Bottom - pRc->Top + 1; /\*弹出窗口的高度\*/

/\*申请存放弹出窗口相关信息的动态存储区\*/

nextLayer = (LAYER\_NODE \*)malloc(sizeof(LAYER\_NODE));

nextLayer->next = gp\_top\_layer;

nextLayer->LayerNo = gp\_top\_layer->LayerNo + 1;

nextLayer->rcArea = \*pRc;

nextLayer->pContent = (CHAR\_INFO \*)malloc(size.X\*size.Y \* sizeof(CHAR\_INFO));

nextLayer->pScrAtt = (char \*)malloc(size.X\*size.Y \* sizeof(char));

pCh = nextLayer->pScrAtt;

/\*将弹出窗口覆盖区域的字符信息保存，用于在关闭弹出窗口时恢复原样\*/

ReadConsoleOutput(gh\_std\_out, nextLayer->pContent, size, pos, pRc);

for (i = pRc->Top; i <= pRc->Bottom; i++)

{ /\*此二重循环将所覆盖字符单元的原先属性值存入动态存储区，便于以后恢复\*/

for (j = pRc->Left; j <= pRc->Right; j++)

{

\*pCh = gp\_scr\_att[i\*SCR\_COL + j];

pCh++;

}

}

gp\_top\_layer = nextLayer; /\*完成弹出窗口相关信息入栈操作\*/

/\*设置弹出窗口区域字符的新属性\*/

pos.X = pRc->Left;

pos.Y = pRc->Top;

for (i = pRc->Top; i <= pRc->Bottom; i++)

{

FillConsoleOutputAttribute(gh\_std\_out, att, size.X, pos, &ul);

pos.Y++;

}

/\*将标签束中的标签字符串在设定的位置输出\*/

for (i = 0; i<pLabel->num; i++)

{

pCh = pLabel->ppLabel[i];

if (strlen(pCh) != 0)

{

WriteConsoleOutputCharacter(gh\_std\_out, pCh, strlen(pCh),

pLabel->pLoc[i], &ul);

}

}

/\*设置弹出窗口区域字符单元的新属性\*/

for (i = pRc->Top; i <= pRc->Bottom; i++)

{ /\*此二重循环设置字符单元的层号\*/

for (j = pRc->Left; j <= pRc->Right; j++)

{

gp\_scr\_att[i\*SCR\_COL + j] = gp\_top\_layer->LayerNo;

}

}

for (i = 0; i<pHotArea->num; i++)

{ /\*此二重循环设置所有热区中字符单元的热区类型和热区编号\*/

row = pHotArea->pArea[i].Top;

for (j = pHotArea->pArea[i].Left; j <= pHotArea->pArea[i].Right; j++)

{

gp\_scr\_att[row\*SCR\_COL + j] |= (pHotArea->pSort[i] << 6)

| (pHotArea->pTag[i] << 2);

}

}

return;

}

/\*\*

\* 函数名称: PopOff

\* 函数功能: 关闭顶层弹出窗口, 恢复覆盖区域原外观和字符单元原属性.

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void PopOff(void)

{

LAYER\_NODE \*nextLayer;

COORD size;

COORD pos = { 0, 0 };

char \*pCh;

int i, j;

if ((gp\_top\_layer->next == NULL) || (gp\_top\_layer->pContent == NULL))

{ /\*栈底存放的主界面屏幕信息，不用关闭\*/

return;

}

nextLayer = gp\_top\_layer->next;

/\*恢复弹出窗口区域原外观\*/

size.X = gp\_top\_layer->rcArea.Right - gp\_top\_layer->rcArea.Left + 1;

size.Y = gp\_top\_layer->rcArea.Bottom - gp\_top\_layer->rcArea.Top + 1;

WriteConsoleOutput(gh\_std\_out, gp\_top\_layer->pContent, size, pos, &(gp\_top\_layer->rcArea));

/\*恢复字符单元原属性\*/

pCh = gp\_top\_layer->pScrAtt;

for (i = gp\_top\_layer->rcArea.Top; i <= gp\_top\_layer->rcArea.Bottom; i++)

{

for (j = gp\_top\_layer->rcArea.Left; j <= gp\_top\_layer->rcArea.Right; j++)

{

gp\_scr\_att[i\*SCR\_COL + j] = \*pCh;

pCh++;

}

}

free(gp\_top\_layer->pContent); /\*释放动态存储区\*/

free(gp\_top\_layer->pScrAtt);

free(gp\_top\_layer);

gp\_top\_layer = nextLayer;

gi\_sel\_sub\_menu = 0;

return;

}

/\*\*

\* 函数名称: DrawBox

\* 函数功能: 在指定区域画边框.

\* 输入参数: pRc 存放区域位置信息的地址

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void DrawBox(SMALL\_RECT \*pRc)

{

char chBox[] = { '+','-','|' }; /\*画框用的字符\*/

COORD pos = { pRc->Left, pRc->Top }; /\*定位在区域的左上角\*/

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框左上角\*/

for (pos.X = pRc->Left + 1; pos.X < pRc->Right; pos.X++)

{ /\*此循环画上边框横线\*/

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[1], 1, pos, &ul);

}

pos.X = pRc->Right;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框右上角\*/

for (pos.Y = pRc->Top + 1; pos.Y < pRc->Bottom; pos.Y++)

{ /\*此循环画边框左边线和右边线\*/

pos.X = pRc->Left;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[2], 1, pos, &ul);

pos.X = pRc->Right;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[2], 1, pos, &ul);

}

pos.X = pRc->Left;

pos.Y = pRc->Bottom;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框左下角\*/

for (pos.X = pRc->Left + 1; pos.X < pRc->Right; pos.X++)

{ /\*画下边框横线\*/

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[1], 1, pos, &ul);

}

pos.X = pRc->Right;

WriteConsoleOutputCharacter(gh\_std\_out, &chBox[0], 1, pos, &ul);/\*画边框右下角\*/

return;

}

/\*\*

\* 函数名称: TagSubMenu

\* 函数功能: 在指定子菜单项上做选中标记.

\* 输入参数: num 选中的子菜单项号

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void TagSubMenu(int num)

{

SMALL\_RECT rcPop;

COORD pos;

WORD att;

int width;

LocSubMenu(gi\_sel\_menu, &rcPop); /\*计算弹出子菜单的区域位置, 存放在rcPop中\*/

if ((num<1) || (num == gi\_sel\_sub\_menu) || (num>rcPop.Bottom - rcPop.Top - 1))

{ /\*如果子菜单项号越界，或该项子菜单已被选中，则返回\*/

return;

}

pos.X = rcPop.Left + 2;

width = rcPop.Right - rcPop.Left - 3;

if (gi\_sel\_sub\_menu != 0) /\*首先取消原选中子菜单项上的标记\*/

{

pos.Y = rcPop.Top + gi\_sel\_sub\_menu;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

FillConsoleOutputAttribute(gh\_std\_out, att, width, pos, &ul);

pos.X += 1;

att |= FOREGROUND\_RED;/\*白底红字\*/

FillConsoleOutputAttribute(gh\_std\_out, att, 1, pos, &ul);

}

/\*在制定子菜单项上做选中标记\*/

pos.X = rcPop.Left + 2;

pos.Y = rcPop.Top + num;

att = FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_RED; /\*黑底白字\*/

FillConsoleOutputAttribute(gh\_std\_out, att, width, pos, &ul);

gi\_sel\_sub\_menu = num; /\*修改选中子菜单项号\*/

return;

}

/\*\*

\* 函数名称: LocSubMenu

\* 函数功能: 计算弹出子菜单区域左上角和右下角的位置.

\* 输入参数: num 选中的主菜单项号

\* 输出参数: rc 存放区域位置信息的地址

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void LocSubMenu(int num, SMALL\_RECT \*rc)

{

int i, len, loc = 0;

rc->Top = 1; /\*区域的上边定在第2行，行号为1\*/

rc->Left = 1;

for (i = 1; i<num; i++)

{ /\*计算区域左边界位置, 同时计算第一个子菜单项在子菜单字符串数组中的位置\*/

rc->Left += strlen(ga\_main\_menu[i - 1]) + 4;

loc += ga\_sub\_menu\_count[i - 1];

}

rc->Right = strlen(ga\_sub\_menu[loc]);/\*暂时存放第一个子菜单项字符串长度\*/

for (i = 1; i<ga\_sub\_menu\_count[num - 1]; i++)

{ /\*查找最长子菜单字符串，将其长度存放在rc->Right\*/

len = strlen(ga\_sub\_menu[loc + i]);

if (rc->Right < len)

{

rc->Right = len;

}

}

rc->Right += rc->Left + 3; /\*计算区域的右边界\*/

rc->Bottom = rc->Top + ga\_sub\_menu\_count[num - 1] + 1;/\*计算区域下边的行号\*/

if (rc->Right >= SCR\_COL) /\*右边界越界的处理\*/

{

len = rc->Right - SCR\_COL + 1;

rc->Left -= len;

rc->Right = SCR\_COL - 1;

}

return;

}

/\*\*

\* 函数名称: DealInput

\* 函数功能: 在弹出窗口区域设置热区, 等待并相应用户输入.

\* 输入参数: pHotArea

\* piHot 焦点热区编号的存放地址, 即指向焦点热区编号的指针

\* 输出参数: piHot 用鼠标单击、按回车或空格时返回当前热区编号

\* 返 回 值:

\*

\* 调用说明:

\*/

int DealInput(HOT\_AREA \*pHotArea, int \*piHot)

{

INPUT\_RECORD inRec;

DWORD res;

COORD pos = { 0, 0 };

int num, arrow, iRet = 0;

int cNo, cTag, cSort;/\*cNo:层号, cTag:热区编号, cSort: 热区类型\*/

char vkc, asc; /\*vkc:虚拟键代码, asc:字符的ASCII码值\*/

SetHotPoint(pHotArea, \*piHot);

while (TRUE)

{ /\*循环\*/

ReadConsoleInput(gh\_std\_in, &inRec, 1, &res);

if ((inRec.EventType == MOUSE\_EVENT) &&

(inRec.Event.MouseEvent.dwButtonState

== FROM\_LEFT\_1ST\_BUTTON\_PRESSED))

{

pos = inRec.Event.MouseEvent.dwMousePosition;

cNo = gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] & 3;

cTag = (gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] >> 2) & 15;

cSort = (gp\_scr\_att[pos.Y \* SCR\_COL + pos.X] >> 6) & 3;

if ((cNo == gp\_top\_layer->LayerNo) && cTag > 0)

{

\*piHot = cTag;

SetHotPoint(pHotArea, \*piHot);

if (cSort == 0)

{

iRet = 13;

break;

}

}

}

else if (inRec.EventType == KEY\_EVENT && inRec.Event.KeyEvent.bKeyDown)

{

vkc = inRec.Event.KeyEvent.wVirtualKeyCode;

asc = inRec.Event.KeyEvent.uChar.AsciiChar;;

if (asc == 0)

{

arrow = 0;

switch (vkc)

{ /\*方向键(左、上、右、下)的处理\*/

case 37: arrow = 1; break;

case 38: arrow = 2; break;

case 39: arrow = 3; break;

case 40: arrow = 4; break;

}

if (arrow > 0)

{

num = \*piHot;

while (TRUE)

{

if (arrow < 3)

{

num--;

}

else

{

num++;

}

if ((num < 1) || (num > pHotArea->num) ||

((arrow % 2) && (pHotArea->pArea[num - 1].Top

== pHotArea->pArea[\*piHot - 1].Top)) || ((!(arrow % 2))

&& (pHotArea->pArea[num - 1].Top

!= pHotArea->pArea[\*piHot - 1].Top)))

{

break;

}

}

if (num > 0 && num <= pHotArea->num)

{

\*piHot = num;

SetHotPoint(pHotArea, \*piHot);

}

}

}

else if (vkc == 27)

{ /\*ESC键\*/

iRet = 27;

break;

}

else if (vkc == 13 || vkc == 32)

{ /\*回车键或空格表示按下当前按钮\*/

iRet = 13;

break;

}

}

}

return iRet;

}

void SetHotPoint(HOT\_AREA \*pHotArea, int iHot)

{

CONSOLE\_CURSOR\_INFO lpCur;

COORD pos = { 0, 0 };

WORD att1, att2;

int i, width;

att1 = FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_RED; /\*黑底白字\*/

att2 = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

for (i = 0; i<pHotArea->num; i++)

{ /\*将按钮类热区置为白底黑字\*/

pos.X = pHotArea->pArea[i].Left;

pos.Y = pHotArea->pArea[i].Top;

width = pHotArea->pArea[i].Right - pHotArea->pArea[i].Left + 1;

if (pHotArea->pSort[i] == 0)

{ /\*热区是按钮类\*/

FillConsoleOutputAttribute(gh\_std\_out, att2, width, pos, &ul);

}

}

pos.X = pHotArea->pArea[iHot - 1].Left;

pos.Y = pHotArea->pArea[iHot - 1].Top;

width = pHotArea->pArea[iHot - 1].Right - pHotArea->pArea[iHot - 1].Left + 1;

if (pHotArea->pSort[iHot - 1] == 0)

{ /\*被激活热区是按钮类\*/

FillConsoleOutputAttribute(gh\_std\_out, att1, width, pos, &ul);

}

else if (pHotArea->pSort[iHot - 1] == 1)

{ /\*被激活热区是文本框类\*/

SetConsoleCursorPosition(gh\_std\_out, pos);

GetConsoleCursorInfo(gh\_std\_out, &lpCur);

lpCur.bVisible = TRUE;

SetConsoleCursorInfo(gh\_std\_out, &lpCur);

}

}

/\*\*

\* 函数名称: ExeFunction

\* 函数功能: 执行由主菜单号和子菜单号确定的功能函数.

\* 输入参数: m 主菜单项号

\* s 子菜单项号

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE 或 FALSE

\*

\* 调用说明: 仅在执行函数ExitSys时, 才可能返回FALSE, 其他情况下总是返回TRUE

\*/

BOOL ExeFunction(int m, int s)

{

BOOL bRet = TRUE;

/\*函数指针数组，用来存放所有功能函数的入口地址\*/

BOOL(\*pFunction[19])(void);// [ga\_sub\_menu\_count[0] + ga\_sub\_menu\_count[1] + ga\_sub\_menu\_count[2] + ga\_sub\_menu\_count[3] + ga\_sub\_menu\_count[4]])(void);

int i, loc;

/\*将功能函数入口地址存入与功能函数所在主菜单号和子菜单号对应下标的数组元素\*/

pFunction[0] = SaveData;

pFunction[1] = ExitSys;

pFunction[2] = MaintainSchInfo;

pFunction[3] = MaintainTeamInfo;

pFunction[4] = MaintainSubjectInfo;

pFunction[5] = QurrySchoolP;

pFunction[6] = QurrySchoolName;

pFunction[7] = NULL;

pFunction[8] = QurryTeamName;

pFunction[9] = QurryProfAcc;

pFunction[10] = NULL;

pFunction[11] = QurrySubID;

pFunction[12] = QurryTeamBelong;

pFunction[13] = StatPro2Stu;

pFunction[14] = StatSubjectAcc;

pFunction[15] = StatESubject;

pFunction[16] = StatSub2Pro;

pFunction[17] = Other;

pFunction[18] = About;

for (i = 1, loc = 0; i<m; i++) /\*根据主菜单号和子菜单号计算对应下标\*/

{

loc += ga\_sub\_menu\_count[i - 1];

}

loc += s - 1;

if (pFunction[loc] != NULL)

{

bRet = (\*pFunction[loc])(); /\*用函数指针调用所指向的功能函数\*/

}

return bRet;

}

/\*\*

\* 函数名称: SaveData

\* 函数功能: 保存数据

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: 1

\*

\* 调用说明:调用SaveSysdata

\*/

BOOL SaveData(void)

{

system("cls");

BOOL bRet = TRUE;

char \*plabel\_name[] = { "主菜单项：文件",

"子菜单项：数据保存",

"确认"

};

ShowModule(plabel\_name, 3);

SaveSysData(gp\_head);

printf("保存成功!\n");

return bRet;

}

/\*\*

\* 函数名称: ExitSys

\* 函数功能: 推出系统

\* 输入参数: 无

\* 输出参数: 状态参量

\* 返 回 值: 1

\*

\* 调用说明:DealInput PopOff PopUp

\*/

BOOL ExitSys(void)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

BOOL bRet = TRUE;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

char \*pCh[] = { "确认退出系统吗？", "确定 取消" };

int iHot = 1;

pos.X = strlen(pCh[0]) + 6;

pos.Y = 7;

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = 2;

labels.ppLabel = pCh;

COORD aLoc[] = { { rcPop.Left + 3, rcPop.Top + 2 },

{ rcPop.Left + 5, rcPop.Top + 5 } };

labels.pLoc = aLoc;

areas.num = 2;

SMALL\_RECT aArea[] = { { rcPop.Left + 5, rcPop.Top + 5,

rcPop.Left + 8, rcPop.Top + 5 },

{ rcPop.Left + 13, rcPop.Top + 5,

rcPop.Left + 16, rcPop.Top + 5 } };

char aSort[] = { 0, 0 };

char aTag[] = { 1, 2 };

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 4;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right - rcPop.Left - 1, pos, &ul);

if (DealInput(&areas, &iHot) == 13 && iHot == 1)

{

bRet = FALSE;

}

else

{

bRet = TRUE;

}

PopOff();

return bRet;

}

/\*\*

\* 函数名称: MainTainSchinfo

\* 函数功能: 维护学院信息

\* 输入参数: 无

\* 输出参数: 状态参数

\* 返 回 值: TRUE

\*

\* 调用说明:DelSchoolNode InsertSchoolNode ModifySchool

\*/

BOOL MaintainSchInfo(void)

{

BOOL bRet = TRUE;

int model;

char \*plabel\_name[] = { "主菜单项：数据维护",

"子菜单项：学院信息",

"确认"

};

ShowModule(plabel\_name, 3);

system("cls");

printf("请输入想进行的操作:\n");

printf("(1)输入学院信息 (2)删除学院信息 （3）修改学院信息 (4)返回上一级\n");

scanf("%d", &model);

getchar();

switch (model)

{

case 1:

InsertSchoolNode();

break;

case 2:

DelSchoolNode();

break;

case 3:

ModifySchoolInfo();

break;

case 4:

system("cls");

return TRUE;

default:

printf("错误操作\n");

break;

}

return bRet;

}

/\*\*

\* 函数名称: MainTainTeaminfo

\* 函数功能: 维护团队信息

\* 输入参数: 无

\* 输出参数: 状态参数

\* 返 回 值: TRUE

\*

\* 调用说明:DelTeamNode InsertTeamNode ModifyTeaminfo

\*/

BOOL MaintainTeamInfo(void)

{

BOOL bRet = TRUE;

int model;

char \*plabel\_name[] = { "主菜单项：数据维护",

"子菜单项：学院信息",

"确认"

};

ShowModule(plabel\_name, 3);

system("cls");

printf("请输入想进行的操作:\n");

printf("(1)输入团队信息 (2)删除团队信息 （3）修改团队信息 (4)返回上一级\n");

scanf("%d", &model);

getchar();

switch (model)

{

case 1:

InsertTeamNode();

break;

case 2:

DelTeamNode();

break;

case 3:

ModifyTeamInfo();

break;

case 4:

system("cls");

return TRUE;

default:

printf("错误操作\n");

break;

}

ShowModule(plabel\_name, 3);

return bRet;

}

/\*\*

\* 函数名称: MainTainSubinfo

\* 函数功能: 维护项目信息

\* 输入参数: 无

\* 输出参数: 状态参数

\* 返 回 值: TRUE

\*

\* 调用说明:DelSubNode InsertSubNode ModifySubNode

\*/

BOOL MaintainSubjectInfo(void)

{

BOOL bRet = TRUE;

int model;

char \*plabel\_name[] = { "主菜单项：数据维护",

"子菜单项：学院信息",

"确认"

};

ShowModule(plabel\_name, 3);

system("cls");

printf("请输入想进行的操作:\n");

printf("(1)输入项目信息 (2)删除项目信息 （3）修改项目信息 (4)返回上一级\n");

printf("请选择操作模式:");

scanf("%d", &model);

getchar();

switch (model)

{

case 1:

InsertProInfo();

break;

case 2:

DelProInfo();

break;

case 3:

ModifyProInfo\_2();

break;

case 4:

system("cls");

return TRUE;

default:

printf("错误操作\n");

break;

}

return bRet;

}

/\*\*

\* 函数名称: InsertProNode

\* 函数功能: 在十字链表中插入一个课题信息结点.

\* 输入参数: void

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示插入成功, FALSE表示插入失败

\*

\* 调用说明:

\*/

BOOL InsertProInfo(void)

{

system("cls");

SUBJECT\_NODE\* new\_subject = (SUBJECT\_NODE\*)malloc(sizeof(SUBJECT\_NODE));

TEAM\_NODE \*temp;

printf("请输入项目所属团队:");

scanf("%s", new\_subject->team);

getchar();

if ((temp = MatchTeamName(gp\_head, new\_subject->team)) != NULL)

{

new\_subject->next = temp->Shead;

temp->Shead = new\_subject;

FullFillProInfo(new\_subject);

printf("录入成功\n");

return TRUE;

}

else

{

printf("\n未找到该团队.请重试.\n");

return FALSE;

}

}

/\*\*

\* 函数名称: DelSubjectNode

\* 函数功能: 从十字链表中删除指定的课题信息结点.

\* 输入参数: hd 主链头指针

\* Team\_id 课题团队学号

\* date 课题日期

\* 输出参数: 无

\* 返 回 值: BOOL类型, TRUE表示删除成功, FALSE表示删除失败

\*

\* 调用说明: 根据学号和课题日期可以确定唯一的课题信息

\*/

BOOL DelSubjectNode(void)

{

system("cls");

char ID[15];

printf("请输入项目编号：");

scanf("%s", ID);

getchar();

TEAM\_NODE\* tmp1 = SeekProID\_2(gp\_head, ID);

SUBJECT\_NODE\* pS1, \*pS2, \*tmp2;

if (tmp1 != NULL)

tmp2 = SeekProID(gp\_head, ID);

if (tmp2 != NULL)

{

pS1 = NULL;

pS2 = tmp1->Shead;

while (pS2 != NULL&&pS2 != tmp2)

{

pS1 = pS2;

pS2 = pS2->next;

}

if (pS2 == tmp2)

{

if (pS1 == NULL)

tmp1->Shead = pS2->next;

else

pS1->next = pS2->next;

printf("删除成功！\n");

free(pS2);

return TRUE;

}

}

return TRUE;

}

/\*\*

\* 函数名称: SeekProID

\* 函数功能: 在十字链表中寻找一个课题信息结点.

\* 输入参数: 主链头指针

\* 输出参数: 无

\* 返 回 值: 课题所属团队指针

\*

\* 调用说明:

\*/

TEAM\_NODE\* SeekProID\_2(SCHOOL\_NODE\*hd, char \*id)

{

SCHOOL\_NODE\* pSchool = hd;

TEAM\_NODE \* pTeam;

SUBJECT\_NODE\* pSubject;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

pSubject = pTeam->Shead;

while (pSubject != NULL)

{

if (strcmp(pSubject->ID, id) == 0)

return pTeam;

pSubject = pSubject->next;

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return NULL;

}

/\*\*

\* 函数名称: SaveSysData

\* 函数功能: 保存系统代码表和三类基础数据.

\* 输入参数: hd 主链头结点指针

\* 输出参数:

\* 返 回 值: BOOL类型, 总是为TRUE

\*

\* 调用说明:

\*/

BOOL SaveSysData(SCHOOL\_NODE \*hd)

{

SCHOOL\_NODE\* pSchool;

TEAM\_NODE \*pTeam;

SUBJECT\_NODE \*pSubject;

FILE \*pfout;

int handle;

pfout = fopen(gp\_school\_info\_filename, "wb");

for (pSchool = hd; pSchool != NULL; pSchool = pSchool->next)

fwrite(pSchool, sizeof(SCHOOL\_NODE), 1, pfout);

fclose(pfout);

pfout = fopen(gp\_team\_info\_filename, "wb");

for (pSchool = hd; pSchool != NULL; pSchool = pSchool->next)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

fwrite(pTeam, sizeof(TEAM\_NODE), 1, pfout);

pTeam = pTeam->next;

}

}

fclose(pfout);

pfout = fopen(gp\_subject\_info\_filename, "wb");

for (pSchool = hd; pSchool != NULL; pSchool = pSchool->next)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

pSubject = pTeam->Shead;

while (pSubject != NULL)

{

fwrite(pSubject, sizeof(SUBJECT\_NODE), 1, pfout);

pSubject = pSubject->next;

}

pTeam = pTeam->next;

}

}

fclose(pfout);

return TRUE;

}

BOOL ShowModule(char \*\*pString, int n)

{

LABEL\_BUNDLE labels;

HOT\_AREA areas;

BOOL bRet = TRUE;

SMALL\_RECT rcPop;

COORD pos;

WORD att;

int iHot = 1;

int i, maxlen, str\_len;

for (i = 0, maxlen = 0; i<n; i++) {

str\_len = strlen(pString[i]);

if (maxlen < str\_len) {

maxlen = str\_len;

}

}

pos.X = maxlen + 6;

pos.Y = n + 5;

rcPop.Left = (SCR\_COL - pos.X) / 2;

rcPop.Right = rcPop.Left + pos.X - 1;

rcPop.Top = (SCR\_ROW - pos.Y) / 2;

rcPop.Bottom = rcPop.Top + pos.Y - 1;

att = BACKGROUND\_BLUE | BACKGROUND\_GREEN | BACKGROUND\_RED; /\*白底黑字\*/

labels.num = n;

labels.ppLabel = pString;

COORD aLoc[100];

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

aLoc[i].X = rcPop.Left + 3;

aLoc[i].Y = rcPop.Top + 2 + i;

}

str\_len = strlen(pString[n - 1]);

aLoc[n - 1].X = rcPop.Left + 3 + (maxlen - str\_len) / 2;

aLoc[n - 1].Y = aLoc[n - 1].Y + 2;

labels.pLoc = aLoc;

areas.num = 1;

SMALL\_RECT aArea[] = { { aLoc[n - 1].X, aLoc[n - 1].Y,

aLoc[n - 1].X + 3, aLoc[n - 1].Y } };

char aSort[] = { 0 };

char aTag[] = { 1 };

areas.pArea = aArea;

areas.pSort = aSort;

areas.pTag = aTag;

PopUp(&rcPop, att, &labels, &areas);

pos.X = rcPop.Left + 1;

pos.Y = rcPop.Top + 2 + n;

FillConsoleOutputCharacter(gh\_std\_out, '-', rcPop.Right - rcPop.Left - 1, pos, &ul);

DealInput(&areas, &iHot);

PopOff();

return bRet;

}

/\*\*

\* 函数名称: QurrySchoolP

\* 函数功能: 查询学院负责人.

\* 输入参数: woid

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:SearchPIC

\*/

BOOL QurrySchoolP(void) {

system("cls");

char person[12];

SCHOOL\_NODE\* temp;

printf("\n\n请输入负责人姓名:");

scanf("%s", person);

if ((temp = SearchPIC(gp\_head, person)) != NULL)

{

printf("院系名称 负责人 联系电话 \n");

printf("%-20s%-12s%-15s\n", temp->name, temp->person\_in\_charge, temp->phone);

}

else

printf("\nNOT FOUND!\n");

return TRUE;

}

/\*\*

\* 函数名称: SearchSchoolPIC

\* 函数功能: 遍历链表查询学院负责人.

\* 输入参数: 主链头指针 负责人姓名

\* 输出参数: 所在院系指针

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:SearchPIC

\*/

SCHOOL\_NODE \*SearchPIC(SCHOOL\_NODE\* hd, char\*person)

{

SCHOOL\_NODE \*pSchool = hd;

while (pSchool != NULL)

{

if (strcmp(pSchool->person\_in\_charge, person) == 0)

return pSchool;

pSchool = pSchool->next;

}

return NULL;

}

/\*\*

\* 函数名称: QurryTeamName

\* 函数功能: 查询团队名称

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:SearchPIC

\*/

BOOL QurryTeamName(void)

{

system("cls");

char s1[] = "团队";

char s2[] = "负责人";

char s3[] = "所属学院";

char s4[] = "教师总数";

char s5[] = "学生总数";

char name[20];

printf("请输入团队名称:");

scanf("%s", name);

printf("%-30s%-12s%s %s %s\n", s1, s2, s4, s5, s3);

if (!SeekTeamName(gp\_head, name))

printf("\nNOT FOUND!\n");

return TRUE;

}/\*\*

\* 函数名称: QurryProfAcc

\* 函数功能: 查询学院教师人数，输出大于该值的所有团队

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:

\*/

BOOL QurryProfAcc(void)

{

char s1[] = "学院";

char s2[] = "负责人";

char s3[] = "所属学院";

char s4[] = "教师总数";

char s5[] = "学生总数";

system("cls");

int num, count = 0;

printf("教师人数:");

scanf("%d", &num);

printf("%-30s%-12s%s %s%s\n", s1, s2, s4, s5, s3);

SCHOOL\_NODE\*pSchool;

TEAM\_NODE \*pTeam;

pSchool = gp\_head;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

if (pTeam->tea\_amount >= num)

{

printf("%-30s%-12s %d %d %-20s\n", pTeam->name, pTeam->person\_in\_charge, pTeam->tea\_amount, pTeam->ug\_amount, pTeam->school);

count++;

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

if (count == 0)

printf("\n没有找到符合要求的团队信息!\n");

return TRUE;

}

/\*\*

\* 函数名称: QurrySubID

\* 函数功能: 查询项目编号.

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:

\*/

BOOL QurrySubID(void)

{

system("cls");

char ID[15];

SUBJECT\_NODE \*temp;

printf("请输入项目编号:");

scanf("%s", ID);

char s1[] = "项目编号";

char s2[] = "项目类别";

char s3[] = "资金";

char s4[] = "负责人;";

char s5[] = "所属团队";

char s6[] = "起始时间";

printf("%-15s%s %-8s%-8s %-12s%-30s\n",s1,s2,s6,s3,s4,s5);

if ((temp = SeekProID(gp\_head, ID)) != NULL)

printf("%-15s%c %-8s %.2f %-12s%-30s\n", temp->ID, temp->type, temp->begin\_time, temp->fund, temp->person\_in\_charge, temp->team);

else printf("\nNot Found!\n");

return TRUE;

}

/\*\*

\* 函数名称: QurryTeamBelonging

\* 函数功能: 查询团队所有项目信息

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\*/

BOOL QurryTeamBelong(void)

{

system("cls");

printf("输入团队名：");

char team[30];

scanf("%s", team);

getchar();

SUBJECT\_NODE \*pSubject;

TEAM\_NODE\* pTeam = MatchTeamName(gp\_head, team);

if (pTeam != NULL)

{

pSubject = pTeam->Shead;

printf("项目编号 项目类别 起始时间 资金 负责人 所属团队\n");

while (pSubject != NULL)

{

printf("%-15s%c %-8s %f %-12s%-30s\n", pSubject->ID, pSubject->type, pSubject->begin\_time, pSubject->fund, pSubject->person\_in\_charge, pSubject->team);

pSubject = pSubject->next;

}

}

return TRUE;

}

BOOL StatPro2Stu(void)

{

system("cls");

statistic\_1(gp\_head);

return TRUE;

}

BOOL StatSubjectAcc(void)

{

system("cls");

statistic\_2(gp\_head);

return TRUE;

}

BOOL StatESubject(void)

{

system("cls");

statistic\_3(gp\_head);

return TRUE;

}

BOOL StatSub2Pro(void)

{

system("cls");

statistic\_4(gp\_head);

return TRUE;

}

BOOL Other(void)

{

system("cls");

StatSchInfo(gp\_head);

return TRUE;

}

BOOL About(void)

{

printf("\n\n\n Developed By XXX\n If you find any problem,contact me on XXXX\n\n");

return TRUE;

}

BOOL QurrySchoolName(void)

{

system("cls");

char name[30];

printf("\n请输入院系名称:");

scanf("%s", name);

printf("院系名称 负责人 联系电话 \n");

if (!SeekSchoolName(gp\_head, name))

printf("\nNOT FOUND!\n");

return TRUE;

}

BOOL SeekSchoolName(SCHOOL\_NODE \*hd, char\* name)

{

BOOL signal = FALSE;

SCHOOL\_NODE \*pSchool = hd;

while (pSchool != NULL)

{

if (strstr(pSchool->name, name) != NULL)

{

signal = TRUE;

printf("%-20s%-12s%-15s\n", pSchool->name, pSchool->person\_in\_charge, pSchool->phone);

}

pSchool = pSchool->next;

}

return signal;

}

BOOL SeekTeamName(SCHOOL\_NODE\* hd, char \*name)

{

BOOL signal = FALSE;

SCHOOL\_NODE \*pSchool = hd;

TEAM\_NODE \*pTeam;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

if (strstr(pTeam->name, name) != NULL)

{

signal = TRUE;

printf("%-30s%-12s %d %d %-20s\n", pTeam->name, pTeam->person\_in\_charge, pTeam->tea\_amount, pTeam->ug\_amount, pTeam->school);

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return signal;

}

/\*\*

\* 函数名称: SeekProId

\* 函数功能: 遍历列表查询项目编号

\* 输入参数: 主链头指针 项目编号

\* 输出参数: 所在项目指针

\* 返 回 值:

\*

\* 调用说明:

\*/

SUBJECT\_NODE\* SeekProID(SCHOOL\_NODE \*hd, char \*ID)

{

SCHOOL\_NODE\*pSchool;

TEAM\_NODE \*pTeam;

SUBJECT\_NODE\* pSubject;

pSchool = hd;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

pSubject = pTeam->Shead;

while (pSubject != NULL)

{

if (strcmp(ID, pSubject->ID) == 0)

return pSubject;

pSubject = pSubject->next;

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return NULL;

}

/\*\*

\* 函数名称: MatchTeamName

\* 函数功能: 匹配团队信息负责人.

\* 输入参数: 主链头指针

\* 输出参数: 所在团队指针

\* 返 回 值:

\* 调用说明:

\*/

TEAM\_NODE \*MatchTeamName(SCHOOL\_NODE\*hd, char\* name)

{

SCHOOL\_NODE\* pSchool = hd;

TEAM\_NODE \*pTeam;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

if (strcmp(name, pTeam->name) == 0)

return pTeam;

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return NULL;

}

/\*\*

\* 函数名称: FullfillProInfo

\* 函数功能: 填写项目信息

\* 输入参数: 对于项目指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:SearchPIC

\*/

void FullFillProInfo(SUBJECT\_NODE \*sub)

{

printf("请输入项目编号;");

scanf("%s", sub->ID);

getchar();

printf("请输入项目类别:");

scanf("%c", &(sub->type));

getchar();

printf("请输入起始时间:");

scanf("%s", sub->begin\_time);

getchar();

printf("请输入项目经费:");

scanf("%f", &(sub->fund));

getchar();

printf("请输入负责人:");

scanf("%s", sub->person\_in\_charge);

getchar();

}

/\*\*

\* 函数名称: DelProInfo

\* 函数功能: 删除课题信息

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:SearchproID

\*/

BOOL DelProInfo(void)

{

system("cls");

char ID[15];

printf("请输入项目编号:");

scanf("%s", ID);

getchar();

SUBJECT\_NODE \*p1, \*p2, \*tmp1;

TEAM\_NODE\*tmp2 = MatchProID\_1(gp\_head, ID);

if (tmp2 != NULL)

tmp1 = MatchProID\_2(gp\_head, ID);

else {

printf("删除失败！\n");

return TRUE;

}

if (tmp1 != NULL)

{

p1 = NULL;

p2 = tmp2->Shead;

while (p2 != tmp1&&p2 != NULL)

{

p1 = p2;

p2 = p2->next;

}

if (p2 == tmp1)

{

if (p1 == NULL)

tmp2->Shead = p2->next;

else

p1->next = p2->next;

free(p2);

printf("删除成功！\n");

return TRUE;

}

}

else printf("删除失败！\n");

return TRUE;

}

TEAM\_NODE\* MatchProID\_1(SCHOOL\_NODE\*hd, char\* id)

{

SCHOOL\_NODE\* pSchool = hd;

TEAM\_NODE \*pTeam;

SUBJECT\_NODE \*pSubject;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

pSubject = pTeam->Shead;

while (pSubject != NULL)

{

if (strcmp(id, pSubject->ID) == 0)

return pTeam;

pSubject = pSubject->next;

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return NULL;

}

BOOL ModifyProInfo\_2(void)

{

system("cls");

char id[15];

printf("请输入项目编号:");

scanf("%s", id);

getchar();

SUBJECT\_NODE \*temp = MatchProID\_2(gp\_head, id);

if (temp != NULL)

{

FullFillProInfo(temp);

printf("修改成功!\n");

return TRUE;

}

else printf("\n未找到该项目,请重试!");

return FALSE;

}

/\*

\*\*

\* 函数名称: InsertTeamNode

\* 函数功能: 后进先出插入团队节点

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: TRUE代表正常

\*

\*/

SUBJECT\_NODE\* MatchProID\_2(SCHOOL\_NODE\*hd, char\* id)

{

SCHOOL\_NODE\* pSchool = hd;

TEAM\_NODE \*pTeam;

SUBJECT\_NODE \*pSubject;

//遍历链表查找所需信息

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

pSubject = pTeam->Shead;

while (pSubject != NULL)

{

if (strcmp(id, pSubject->ID) == 0)

return pSubject;

pSubject = pSubject->next;

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return NULL;

}

/\*

\*\*

\* 函数名称: InsertTeamNode

\* 函数功能: 后进先出插入团队节点

\* 输入参数: 无

\* 输出参数: 无

\* 返 回 值: TRUE代表正常

\*

\*/

BOOL InsertTeamNode(void)

{

system("cls");

printf("请输入所属院系名称:");

TEAM\_NODE\*new\_team = (TEAM\_NODE\*)malloc(sizeof(TEAM\_NODE));

new\_team->Shead = NULL;

scanf("%s", new\_team->school);

getchar();

SCHOOL\_NODE \*temp = MatchSchoolName(gp\_head, new\_team->school);

if (temp != NULL)

{

new\_team->next = temp->Thead;

temp->Thead = new\_team;

FullFillTeamInfo(new\_team);

return TRUE;

}

else

printf("\n未找到该院系,请重试.\n");

return FALSE;

}

/\*

\*\*

\* 函数名称: MatchSchoolName

\* 函数功能: 查找对应的团队节点返回该队所在学院节点

\* 输入参数: 主链头指针 团队名

\* 输出参数: pTeam

\* 返 回 值: 找到返回对应学院节点，未找到返回NULL

\*

\*/

SCHOOL\_NODE \*MatchSchoolName(SCHOOL\_NODE\*hd, char \* school)

{

SCHOOL\_NODE\* pSchool = hd;

while (pSchool != NULL)

{

if (strcmp(pSchool->name, school) == 0)

return pSchool;

pSchool = pSchool->next;

}

return NULL;

}

/\*\*

\* 函数名称: FullFillTeamInfo

\* 函数功能: 查询学院负责人.

\* 输入参数: 对应团队指针

\* 输出参数: 无

\* 返 回 值: 无

\*

\* 调用说明:

\*/

void FullFillTeamInfo(TEAM\_NODE\* team)

{

printf("请输入团队名称:");

scanf("%s", team->name);

getchar();

printf("请输入负责人姓名:");

scanf("%s", team->person\_in\_charge);

getchar();

printf("请输入教师人数:");

scanf("%d", &(team->tea\_amount));

getchar();

printf("请输入研究生人数：");

scanf("%d", &(team->ug\_amount));

getchar();

}

/\*\*

\* 函数名称: 删除团队信息

\* 函数功能: 删除团队节点.

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:MacthTeamNode\_step\_1,2

\*/

BOOL DelTeamNode(void)

{

system("cls");

char team[30];

TEAM\_NODE\* pT1, \*pT2, \*tmp2;

printf("请输入团队名称:");

scanf("%s", team);

getchar();

SCHOOL\_NODE \*tmp1 = MatchTeamNode\_Step\_1(gp\_head, team);

if (tmp1 != NULL)

tmp2 = MatchTeamNode\_Step\_2(tmp1, team);

else

{

printf("删除失败！\n");

return TRUE;

}

if (tmp2 != NULL)

{

pT1 = NULL;

pT2 = tmp1->Thead;

while (pT2 != NULL&&pT2 != tmp2)

{

pT1 = pT2;

pT2 = pT2->next;

}

if (pT2 == tmp2)

{

if (pT1 == NULL)

tmp1->Thead = pT2->next;

else pT1->next = pT2->next;

free(tmp2);

printf("删除成功\n");

return TRUE;

}

}

else printf("\n未找到该团队\n");

return FALSE;

}

/\*

\*\*

\* 函数名称: MatchTeamNode\_Step\_1

\* 函数功能: 查找对应的团队节点返回该团队所在学院节点

\* 输入参数: 主链头指针 团队名

\* 输出参数: pTeam

\* 返 回 值: 找到返回该团队所在学院节点，未找到返回NULL

\*

\*/

SCHOOL\_NODE \*MatchTeamNode\_Step\_1(SCHOOL\_NODE\* hd, char \* team)

{

SCHOOL\_NODE \*pSchool = hd;

TEAM\_NODE \*pTeam;

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

if (strcmp(pTeam->name, team) == 0)

return pSchool;

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

return NULL;

}

/\*

\*\*

\* 函数名称: MatchTeamNode\_Step\_2

\* 函数功能: 查找对应的团队节点返回该团队节点

\* 输入参数: 主链头指针 团队名

\* 输出参数: pTeam

\* 返 回 值: 找到返回对应节点，未找到返回NULL

\*

\*/

TEAM\_NODE \*MatchTeamNode\_Step\_2(SCHOOL\_NODE\* hd, char \*team)

{

TEAM\_NODE\* pTeam = hd->Thead;

while (pTeam != NULL)

{

if (strcmp(pTeam->name, team) == 0)

return pTeam;

pTeam = pTeam->next;

}

printf("未找到该团队\n");

return NULL;

}

/\*\*

\* 函数名称: ModifyTeamInfo

\* 函数功能: 查询学院负责人.

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:FullFillTeamInfo

\*/

BOOL ModifyTeamInfo(void)

{

system("cls");

char team[30];

SCHOOL\_NODE \*pSchool = gp\_head;

TEAM\_NODE \*pTeam;

printf("请输入团队名称:");

scanf("%s", team);

getchar();

while (pSchool != NULL)

{

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

if (strcmp(pTeam->name, team) == 0)

{

FullFillTeamInfo(pTeam);

return TRUE;

}

pTeam = pTeam->next;

}

pSchool = pSchool->next;

}

printf("\n未找到团队\n");

return FALSE;

}

/\*\*

\* 函数名称: InsertSchoolNode

\* 函数功能: 后进先出插入学院节点

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\* 调用说明:FullFillSchInfo

\*/

BOOL InsertSchoolNode(void)

{

system("cls");

SCHOOL\_NODE \* nSch = (SCHOOL\_NODE\*)malloc(sizeof(SCHOOL\_NODE));

nSch->Thead = NULL;

FullFillSchInfo(nSch);

nSch->next = gp\_head;

gp\_head = nSch;

printf("输入成功！\n");

return TRUE;

}

/\*\*

\* 函数名称: FullFillSchInfo

\* 函数功能: 填写学院信息

\* 输入参数: void

\* 输出参数: 无

\* 返 回 值: 无

\*

\*/

void FullFillSchInfo(SCHOOL\_NODE \* new\_sch)

{

printf("\n请输入院系名:");

scanf("%s", new\_sch->name);

getchar();

printf("请输入负责人姓名:");

scanf("%s", new\_sch->person\_in\_charge);

getchar();

printf("请输入联系电话:");

scanf("%s", new\_sch->phone);

getchar();

}

/\*\*

\* 函数名称: DelSchoolNode

\* 函数功能: 删除学院节点

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表统计正常

\*

\*/

BOOL DelSchoolNode(void)

{

char name[20];

printf("请输入学院名称：");

scanf("%s", name);

getchar();

SCHOOL\_NODE\* temp = MatchSchoolName(gp\_head, name), \*p1, \*p2;

if (temp != NULL)

{

p1 = NULL;

p2 = gp\_head;

while (p2 != NULL&&p2 != temp)

{

p1 = p2;

p2 = p2->next;

}

if (p2 == temp)

{

if (p1 == NULL)

gp\_head = p2->next;

else

p1->next = p2->next;

}

free(p2);

printf("删除成功");

return TRUE;

}

printf("删除失败!\n");

return FALSE;

}

/\*\*

\* 函数名称: SortSchoolInfo\_2

\* 函数功能: 按学院师生比对学院信息链表进行排序

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\*/

void SortSchoolInfo\_1(SCH\_DATA\* hd)

{

SCH\_DATA \*prior, \*after, \*current, \*temp, \*p;

prior = hd;

if (prior == NULL)

return;

temp = (SCH\_DATA\*)malloc(sizeof(SCH\_DATA));

while (prior->next != NULL)

{

current = prior;

after = prior->next;

while (after != NULL)

{

if (current->Pro2Stu <after->Pro2Stu)

current = after;

after = after->next;

}

if (current != prior)

{

\*temp = \*prior;

\*prior = \*current;

prior->next = temp->next;

temp->next = current->next;

\*current = \*temp;

}

prior = prior->next;

}

char str1[] = "学院";

char str2[] = "教师总数";

char str3[] = "学生总数";

char str4[] = "教师比学生";

printf("%s %s %s %s\n", str1, str2, str3, str4);

for (p = hd; p != NULL; p = p->next)

{

if(p->Stu\_Amount==0)

continue;

printf("%s %d %d %f\n", p->name, p->Sch\_prof\_total, p->Stu\_Amount, p->Pro2Stu);

}

free(temp);

return;

}

/\*\*

\* 函数名称: SortSchoolInfo\_2

\* 函数功能: 按学院项目数对学院信息链表进行排序

\* 输入参数: void

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\*/

void SortSchoolInfo\_2(SCH\_DATA\* hd)

{

SCH\_DATA \*prior, \*after, \*current, \*temp;

prior = hd;

if (prior == NULL)

return;

temp = (SCH\_DATA\*)malloc(sizeof(SCH\_DATA));

while (prior->next != NULL)

{

current = prior;

after = prior->next;

while (after != NULL)

{

if (current->sch\_pro\_num <after->sch\_pro\_num)

current = after;

after = after->next;

}

if (current != prior)

{

\*temp = \*prior;

\*prior = \*current;

prior->next = temp->next;

temp->next = current->next;

\*current = \*temp;

}

prior = prior->next;

}

SCH\_DATA \*p = hd;

char s1[] = "学院";

char s2[] = "973项目数";

char s3[] = "863项目数";

char s4[] = "总资金";

printf("%-20s%-10s%-10s%s\n", s1, s2, s3, s4);

for (p = hd; p != NULL; p = p->next)

{

if(p->total\_fund==0) continue;

printf("%-20s%-10d%-10d%-12.2f\n", p->name, p->\_973\_num, p->\_863\_num, p->total\_fund);

}

}

void SortTeamInfo\_1(TEAM\_DATA\* hd)

{

TEAM\_DATA \*prior, \*after, \*current, \*temp;

prior = hd;

if (prior == NULL)

return;

temp = (TEAM\_DATA\*)malloc(sizeof(TEAM\_DATA));

while (prior->next != NULL)

{

current = prior;

after = prior->next;

while (after != NULL)

{

if (current->t\_pro\_amount < after->t\_pro\_amount)

current = after;

after = after->next;

}

if (current != prior)

{

\*temp = \*prior;

\*prior = \*current;

prior->next = temp->next;

temp->next = current->next;

\*current = \*temp;

}

prior = prior->next;

}

int i;

TEAM\_DATA\* p;

char s1[] = "团队名";

char s2[] = "国家自然基金项目";

char s3[] = "团队总基金";

printf("%-30s%-s%s\n", s1, s2, s3);

for (i = 0, p = hd; i < 10 && p != NULL; i++, p = p->next)

printf("%-30s%-8d %-12.2f\n", p->name, p->NA\_pro, p->t\_total\_fund);

}

/\*\*

\* 函数名称: SortTeamInfo\_2

\* 函数功能: 按团队重点项目数对团队信息链表进行排序

\* 输入参数: voud

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表排序正常

\*

\*/

void SortTeamInfo\_2(TEAM\_DATA\* hd)

{

TEAM\_DATA \*prior, \*after, \*current, \*temp;

prior = hd;

if (prior == NULL)

return;

temp = (TEAM\_DATA\*)malloc(sizeof(TEAM\_DATA));

while (prior->next != NULL)

{

current = prior;

after = prior->next;

while (after != NULL)

{

if (current->Pro2Prof< after->Pro2Prof)

current = after;

after = after->next;

}

if (current != prior)

{

\*temp = \*prior;

\*prior = \*current;

prior->next = temp->next;

temp->next = current->next;

\*current = \*temp;

}

prior = prior->next;

}

int i;

TEAM\_DATA \*p;

char s1[] = "团队名称";

char s2[] = "项目总数";

char s3[] = "教师总数";

char s4[] = "项目比教师";

printf("%-30s%-8s%-8s%-12s\n", s1, s2, s3, s4);

for (p = hd, i = 0; i < 5 && p != NULL; p = p->next, i++)

printf("%-30s%-8d%-8d%-12f\n", p->name, p->t\_pro\_amount, p->t\_tea\_amount, p->Pro2Prof);

return;

}

void StatSchInfo(SCHOOL\_NODE\* hd)

{

SCHOOL\_NODE \*p = hd;

TEAM\_NODE\* pTeam;

SUBJECT\_NODE \*pSubject;

char s1[] = "学院";

char s2[] = "负责人";

char s3[] = "联系电话";

printf("%-20s%-12s%-15s\n", s1, s2, s3);

for (p = hd; p != NULL; p = p->next)

{

printf("\n");

printf("%-20s%-12s%-15s\n", p->name, p->person\_in\_charge, p->phone);

pTeam = p->Thead;

while (pTeam != NULL)

{

printf("%-0s%-12s%-15s\n", pTeam->name, pTeam->person\_in\_charge, pTeam->school);

pSubject = pTeam->Shead;

while (pSubject != NULL)

{

printf("%-10s%-12s%-10.2f\n", pSubject->ID, pSubject->person\_in\_charge, pSubject->fund);

pSubject = pSubject->next;

}

pTeam = pTeam->next;

}

}

}

/\*\*

\* 函数名称: statistic\_1

\* 函数功能: 统计相关数据按学院师生比对学院信息链表进行排序

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表统计正常

\*

\*/

void statistic\_1(SCHOOL\_NODE\*hd)

{

SCH\_DATA \*sdata\_hd = NULL;

TEAM\_DATA \*tdata\_hd = NULL;

SCHOOL\_NODE \*pSchool = hd;

TEAM\_NODE\* pTeam;

SUBJECT\_NODE\* pPro;

while (pSchool != NULL)

{

SCH\_DATA \*tmp1 = (SCH\_DATA\*)malloc(sizeof(SCH\_DATA));

tmp1->next = sdata\_hd;

sdata\_hd = tmp1;

//初始化新的学校信息节点

strcpy(sdata\_hd->name, pSchool->name);

sdata\_hd->Sch\_prof\_total = 0;

sdata\_hd->Stu\_Amount = 0;

sdata\_hd->\_863\_num = 0;

sdata\_hd->\_973\_num = 0;

sdata\_hd->total\_fund = 0;

sdata\_hd->sch\_pro\_num = 0;

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

TEAM\_DATA\* tmp2 = (TEAM\_DATA\*)malloc(sizeof(TEAM\_DATA));

tmp2->next = tdata\_hd;

tdata\_hd = tmp2;

//初始化新的团队信息节点

strcpy(tdata\_hd->name, pTeam->name);

tdata\_hd->t\_tea\_amount = pTeam->tea\_amount;

tdata\_hd->t\_pro\_amount = 0;

tdata\_hd->t\_total\_fund = 0;

tdata\_hd->NA\_pro = 0;

//计算学院信息

sdata\_hd->Stu\_Amount += pTeam->ug\_amount;

sdata\_hd->Sch\_prof\_total += pTeam->tea\_amount;

pPro = pTeam->Shead;

while (pPro != NULL)

{

//计算相关数据

sdata\_hd->total\_fund += pPro->fund;

tdata\_hd->t\_total\_fund += pPro->fund;

if (pPro->type == '1')

sdata\_hd->\_973\_num++;

if (pPro->type == '3')

sdata\_hd->\_863\_num++;

if (pPro->type == '2')

tdata\_hd->NA\_pro++;

sdata\_hd->sch\_pro\_num++;

tdata\_hd->t\_pro\_amount++;

pPro = pPro->next;

}

tdata\_hd->Pro2Prof = ((float)tdata\_hd->t\_pro\_amount) / ((float)tdata\_hd->t\_tea\_amount);

pTeam = pTeam->next;

}

sdata\_hd->Pro2Stu = ((float)sdata\_hd->Sch\_prof\_total) / ((float)sdata\_hd->Stu\_Amount);

pSchool = pSchool->next;

}

SortSchoolInfo\_1(sdata\_hd);

}

/\*\*

\* 函数名称: statistic\_2

\* 函数功能: 统计相关数据按学院项目数对学院信息链表进行排序

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表统计正常

\*

\*/

void statistic\_2(SCHOOL\_NODE\*hd)

{

SCH\_DATA \*sdata\_hd = NULL;

TEAM\_DATA \*tdata\_hd = NULL;

SCHOOL\_NODE \*pSchool = hd;

TEAM\_NODE\* pTeam;

SUBJECT\_NODE\* pPro;

while (pSchool != NULL)

{

SCH\_DATA \*tmp1 = (SCH\_DATA\*)malloc(sizeof(SCH\_DATA));

tmp1->next = sdata\_hd;

sdata\_hd = tmp1;

//初始化新的学校信息节点

strcpy(sdata\_hd->name, pSchool->name);

sdata\_hd->Sch\_prof\_total = 0;

sdata\_hd->Stu\_Amount = 0;

sdata\_hd->\_863\_num = 0;

sdata\_hd->\_973\_num = 0;

sdata\_hd->total\_fund = 0;

sdata\_hd->sch\_pro\_num = 0;

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

TEAM\_DATA\* tmp2 = (TEAM\_DATA\*)malloc(sizeof(TEAM\_DATA));

tmp2->next = tdata\_hd;

tdata\_hd = tmp2;

//初始化新的团队信息节点

strcpy(tdata\_hd->name, pTeam->name);

tdata\_hd->t\_tea\_amount = pTeam->tea\_amount;

tdata\_hd->t\_pro\_amount = 0;

tdata\_hd->t\_total\_fund = 0;

tdata\_hd->NA\_pro = 0;

//计算学院信息

sdata\_hd->Stu\_Amount += pTeam->ug\_amount;

sdata\_hd->Sch\_prof\_total += pTeam->tea\_amount;

pPro = pTeam->Shead;

while (pPro != NULL)

{

//计算相关数据

sdata\_hd->total\_fund += pPro->fund;

tdata\_hd->t\_total\_fund += pPro->fund;

if (pPro->type == '1')

sdata\_hd->\_973\_num++;

if (pPro->type == '3')

sdata\_hd->\_863\_num++;

if (pPro->type == '2')

tdata\_hd->NA\_pro++;

sdata\_hd->sch\_pro\_num++;

tdata\_hd->t\_pro\_amount++;

pPro = pPro->next;

}

tdata\_hd->Pro2Prof = ((float)tdata\_hd->t\_pro\_amount) / ((float)tdata\_hd->t\_tea\_amount);

pTeam = pTeam->next;

}

sdata\_hd->Pro2Stu = ((float)sdata\_hd->Sch\_prof\_total) / ((float)sdata\_hd->Stu\_Amount);

pSchool = pSchool->next;

}

SortSchoolInfo\_2(sdata\_hd);

}

/\*\*

\* 函数名称: statistic\_3

\* 函数功能: 统计相关数据,按团队重点项目数对团队信息链表进行排序

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表统计正常

\*

\*/

void statistic\_3(SCHOOL\_NODE\*hd)

{

SCH\_DATA \*sdata\_hd = NULL;

TEAM\_DATA \*tdata\_hd = NULL;

SCHOOL\_NODE \*pSchool = hd;

TEAM\_NODE\* pTeam;

SUBJECT\_NODE\* pPro;

while (pSchool != NULL)

{

SCH\_DATA \*tmp1 = (SCH\_DATA\*)malloc(sizeof(SCH\_DATA));

tmp1->next = sdata\_hd;

sdata\_hd = tmp1;

//初始化新的学校信息节点

strcpy(sdata\_hd->name, pSchool->name);

sdata\_hd->Sch\_prof\_total = 0;

sdata\_hd->Stu\_Amount = 0;

sdata\_hd->\_863\_num = 0;

sdata\_hd->\_973\_num = 0;

sdata\_hd->total\_fund = 0;

sdata\_hd->sch\_pro\_num = 0;

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

TEAM\_DATA\* tmp2 = (TEAM\_DATA\*)malloc(sizeof(TEAM\_DATA));

tmp2->next = tdata\_hd;

tdata\_hd = tmp2;

//初始化新的团队信息节点

strcpy(tdata\_hd->name, pTeam->name);

tdata\_hd->t\_tea\_amount = pTeam->tea\_amount;

tdata\_hd->t\_pro\_amount = 0;

tdata\_hd->t\_total\_fund = 0;

tdata\_hd->NA\_pro = 0;

//计算学院信息

sdata\_hd->Stu\_Amount += pTeam->ug\_amount;

sdata\_hd->Sch\_prof\_total += pTeam->tea\_amount;

pPro = pTeam->Shead;

while (pPro != NULL)

{

//计算相关数据

sdata\_hd->total\_fund += pPro->fund;

tdata\_hd->t\_total\_fund += pPro->fund;

if (pPro->type == '1')

sdata\_hd->\_973\_num++;

if (pPro->type == '3')

sdata\_hd->\_863\_num++;

if (pPro->type == '2')

tdata\_hd->NA\_pro++;

sdata\_hd->sch\_pro\_num++;

tdata\_hd->t\_pro\_amount++;

pPro = pPro->next;

}

tdata\_hd->Pro2Prof = ((float)tdata\_hd->t\_pro\_amount) / ((float)tdata\_hd->t\_tea\_amount);

pTeam = pTeam->next;

}

sdata\_hd->Pro2Stu = ((float)sdata\_hd->Sch\_prof\_total) / ((float)sdata\_hd->Stu\_Amount);

pSchool = pSchool->next;

}

SortTeamInfo\_1(tdata\_hd);

}

/\*\*

\* 函数名称: statistic\_1

\* 函数功能: 统计相关数据按学院师生比对学院信息链表进行排序

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表统计正常

\*

\*/

void statistic\_4(SCHOOL\_NODE\*hd)

{

SCH\_DATA \*sdata\_hd = NULL;

TEAM\_DATA \*tdata\_hd = NULL;

SCHOOL\_NODE \*pSchool = hd;

TEAM\_NODE\* pTeam;

SUBJECT\_NODE\* pPro;

while (pSchool != NULL)

{

SCH\_DATA \*tmp1 = (SCH\_DATA\*)malloc(sizeof(SCH\_DATA));

tmp1->next = sdata\_hd;

sdata\_hd = tmp1;

//初始化新的学校信息节点

strcpy(sdata\_hd->name, pSchool->name);

sdata\_hd->Sch\_prof\_total = 0;

sdata\_hd->Stu\_Amount = 0;

sdata\_hd->\_863\_num = 0;

sdata\_hd->\_973\_num = 0;

sdata\_hd->total\_fund = 0;

sdata\_hd->sch\_pro\_num = 0;

pTeam = pSchool->Thead;

while (pTeam != NULL)

{

TEAM\_DATA\* tmp2 = (TEAM\_DATA\*)malloc(sizeof(TEAM\_DATA));

tmp2->next = tdata\_hd;

tdata\_hd = tmp2;

//初始化新的团队信息节点

strcpy(tdata\_hd->name, pTeam->name);

tdata\_hd->t\_tea\_amount = pTeam->tea\_amount;

tdata\_hd->t\_pro\_amount = 0;

tdata\_hd->t\_total\_fund = 0;

tdata\_hd->NA\_pro = 0;

//计算学院信息

sdata\_hd->Stu\_Amount += pTeam->ug\_amount;

sdata\_hd->Sch\_prof\_total += pTeam->tea\_amount;

pPro = pTeam->Shead;

while (pPro != NULL)

{

//计算相关数据

sdata\_hd->total\_fund += pPro->fund;

tdata\_hd->t\_total\_fund += pPro->fund;

if (pPro->type == '1')

sdata\_hd->\_973\_num++;

if (pPro->type == '3')

sdata\_hd->\_863\_num++;

if (pPro->type == '2')

tdata\_hd->NA\_pro++;

sdata\_hd->sch\_pro\_num++;

tdata\_hd->t\_pro\_amount++;

pPro = pPro->next;

}

tdata\_hd->Pro2Prof = ((float)tdata\_hd->t\_pro\_amount) / ((float)tdata\_hd->t\_tea\_amount);

pTeam = pTeam->next;

}

sdata\_hd->Pro2Stu = ((float)sdata\_hd->Sch\_prof\_total) / ((float)sdata\_hd->Stu\_Amount);

pSchool = pSchool->next;

}

SortTeamInfo\_2(tdata\_hd);

}

/\*\*

\* 函数名称: ModifySchoolInfo

\* 函数功能: 修改学院信息

\* 输入参数: BOOL

\* 输出参数: 状态参量

\* 返 回 值: TRUE代表查询正常

\*

\*/

BOOL ModifySchoolInfo()

{

char name[20];

printf("请输入学院名：");

scanf("%s", name);

getchar();

SCHOOL\_NODE\* temp = MatchSchoolName(gp\_head, name);

if (temp != NULL)

{

FullFillSchInfo(temp);

printf("修改成功！");

return TRUE;

}

else printf("未找到该团队！\n");

return FALSE;

}

|  |
| --- |
| #include <stdio.h> |
|  |

|  |
| --- |
| #include <stdlib.h> |
|  |

|  |
| --- |
| #include <windows.h> |
|  |

|  |
| --- |
| #include <wincon.h> |
|  |

|  |
| --- |
| #include <conio.h> |
|  |

|  |
| --- |
| #include <string.h> |
|  |

|  |
| --- |
| #include <io.h> |
|  |

|  |
| --- |
| #include <fcntl.h> |
|  |

|  |
| --- |
| #include <sys\stat.h> |
|  |

|  |
| --- |
| #include <ctype.h> |
|  |

|  |
| --- |
| #include <time.h> |
|  |

|  |
| --- |
| #include<stdbool.h> |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| #define SCR\_ROW 25 /\*屏幕行数\*/ |
|  |

|  |
| --- |
| #define SCR\_COL 80 /\*屏幕列数\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //data struct |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //院系 |
|  |

|  |
| --- |
| typedef struct school { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| char name[20]; //院系名称 |
|  |

|  |
| --- |
| char person\_in\_charge[12]; //负责人 |
|  |

|  |
| --- |
| char phone[15]; //联系号码 |
|  |

|  |
| --- |
| struct team\* Thead; |
|  |

|  |
| --- |
| struct school\* next; |
|  |

|  |
| --- |
| }SCHOOL\_NODE; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct sch\_data { |
|  |

|  |
| --- |
| char name[20]; |
|  |

|  |
| --- |
| int Sch\_prof\_total; |
|  |

|  |
| --- |
| int Stu\_Amount; |
|  |

|  |
| --- |
| float Pro2Stu; |
|  |

|  |
| --- |
| int \_973\_num; |
|  |

|  |
| --- |
| int \_863\_num; |
|  |

|  |
| --- |
| int sch\_pro\_num; |
|  |

|  |
| --- |
| float total\_fund; |
|  |

|  |
| --- |
| struct sch\_data\* next; |
|  |

|  |
| --- |
| }SCH\_DATA; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //团队 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct team { |
|  |

|  |
| --- |
| char name[20]; //团队名称 |
|  |

|  |
| --- |
| char person\_in\_charge[12]; //负责人 |
|  |

|  |
| --- |
| int tea\_amount; //教授人数 |
|  |

|  |
| --- |
| int ug\_amount; //研究生人数 |
|  |

|  |
| --- |
| char school[20]; //所属院系 |
|  |

|  |
| --- |
| struct team \*next; |
|  |

|  |
| --- |
| struct subject \*Shead; |
|  |

|  |
| --- |
| }TEAM\_NODE; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct team\_data { |
|  |

|  |
| --- |
| char name[30]; |
|  |

|  |
| --- |
| int t\_pro\_amount; |
|  |

|  |
| --- |
| int t\_tea\_amount; |
|  |

|  |
| --- |
| float Pro2Prof; |
|  |

|  |
| --- |
| float t\_total\_fund; |
|  |

|  |
| --- |
| int NA\_pro; |
|  |

|  |
| --- |
| struct team\_data\* next; |
|  |

|  |
| --- |
| }TEAM\_DATA; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| //课题 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct subject { |
|  |

|  |
| --- |
| char ID[15]; //项目编号 |
|  |

|  |
| --- |
| char type; //range from 1 to 5; |
|  |

|  |
| --- |
| char begin\_time[8]; //起始时间 |
|  |

|  |
| --- |
| float fund; //经费 |
|  |

|  |
| --- |
| char person\_in\_charge[12]; //负责人 |
|  |

|  |
| --- |
| char team[30]; //所属团队 |
|  |

|  |
| --- |
| struct subject \*next; |
|  |

|  |
| --- |
| }SUBJECT\_NODE; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \*屏幕窗口信息链结点结点结构 |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct layer\_node { |
|  |

|  |
| --- |
| char LayerNo; /\*\*< 弹出窗口层数\*/ |
|  |

|  |
| --- |
| SMALL\_RECT rcArea; /\*\*< 弹出窗口区域坐标\*/ |
|  |

|  |
| --- |
| CHAR\_INFO \*pContent; /\*\*< 弹出窗口区域字符单元原信息存储缓冲区\*/ |
|  |

|  |
| --- |
| char \*pScrAtt; /\*\*< 弹出窗口区域字符单元原属性值存储缓冲区\*/ |
|  |

|  |
| --- |
| struct layer\_node \*next; /\*\*< 指向下一结点的指针\*/ |
|  |

|  |
| --- |
| } LAYER\_NODE; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \*标签束结构 |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct labe1\_bundle { |
|  |

|  |
| --- |
| char \*\*ppLabel; /\*\*< 标签字符串数组首地址\*/ |
|  |

|  |
| --- |
| COORD \*pLoc; /\*\*< 标签定位数组首地址\*/ |
|  |

|  |
| --- |
| int num; /\*\*< 标签个数\*/ |
|  |

|  |
| --- |
| } LABEL\_BUNDLE; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*热区结构 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| typedef struct hot\_area { |
|  |

|  |
| --- |
| SMALL\_RECT \*pArea; /\*\*< 热区定位数组首地址\*/ |
|  |

|  |
| --- |
| char \*pSort; /\*\*< 热区类别(按键、文本框、选项框)数组首地址\*/ |
|  |

|  |
| --- |
| char \*pTag; /\*\*< 热区序号数组首地址\*/ |
|  |

|  |
| --- |
| int num; /\*\*< 热区个数\*/ |
|  |

|  |
| --- |
| } HOT\_AREA; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| LAYER\_NODE \*gp\_top\_layer = NULL; /\*弹出窗口信息链链头\*/ |
|  |

|  |
| --- |
| SCHOOL\_NODE\*gp\_head = NULL; /\*主链头指针\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| char \*gp\_sys\_name = "科研信息管理系统"; /\*系统名称\*/ |
|  |

|  |
| --- |
| char \*gp\_team\_info\_filename = "team.dat"; /\*学生基本信息数据文件\*/ |
|  |

|  |
| --- |
| char \*gp\_subject\_info\_filename = "subject.dat"; /\*住宿缴费信息数据文件\*/ |
|  |

|  |
| --- |
| char \*gp\_school\_info\_filename = "school.dat"; /\*宿舍楼信息数据文件\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| char \*ga\_main\_menu[] = { "文件(F)", /\*系统主菜单名\*/ |
|  |

|  |
| --- |
| "数据维护(M)", |
|  |

|  |
| --- |
| "数据查询(Q)", |
|  |

|  |
| --- |
| "数据统计(S)", |
|  |

|  |
| --- |
| "帮助(H)" |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| char \*ga\_sub\_menu[] = { "[S] 数据保存", /\*系统子菜单名\*/ |
|  |

|  |
| --- |
| "[X] 退出 Alt+X", |
|  |

|  |
| --- |
| "[S] 院系信息", |
|  |

|  |
| --- |
| "[T] 团队信息", |
|  |

|  |
| --- |
| "[P]科研项目", |
|  |

|  |
| --- |
| "[D] 院系负责人", |
|  |

|  |
| --- |
| "[P]院系名称 ", |
|  |

|  |
| --- |
| "", |
|  |

|  |
| --- |
| "[C] 团队名称", |
|  |

|  |
| --- |
| "[S] 教师人数", |
|  |

|  |
| --- |
| "", |
|  |

|  |
| --- |
| "[I]项目编号", |
|  |

|  |
| --- |
| "[B]所属团队", |
|  |

|  |
| --- |
| "[P] 学生教师比", |
|  |

|  |
| --- |
| "[C] 科研项目数", |
|  |

|  |
| --- |
| "[I] 优秀团队", |
|  |

|  |
| --- |
| "[T] 项目教师比", |
|  |

|  |
| --- |
| "[C] 学院信息概览", |
|  |

|  |
| --- |
| "[A] 关于..." |
|  |

|  |
| --- |
| }; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int ga\_sub\_menu\_count[] = { 2, 3, 8, 5,1 }; /\*各主菜单项下子菜单的个数\*/ |
|  |

|  |
| --- |
| int gi\_sel\_menu = 1; /\*被选中的主菜单项号,初始为1\*/ |
|  |

|  |
| --- |
| int gi\_sel\_sub\_menu = 0; /\*被选中的子菜单项号,初始为0,表示未选中\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| CHAR\_INFO \*gp\_buff\_menubar\_info = NULL; /\*存放菜单条屏幕区字符信息的缓冲区\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| CHAR\_INFO \*gp\_buff\_stateBar\_info = NULL; /\*存放状态条屏幕区字符信息的缓冲区\*/ |
|  |

|  |
| --- |
| char \*gp\_scr\_att = NULL; /\*存放屏幕上字符单元属性值的缓冲区\*/ |
|  |

|  |
| --- |
| char \*gp\_sub\_code = NULL; /\*存放课题代码表的数据缓冲区\*/ |
|  |

|  |
| --- |
| char gc\_sys\_state = '\0'; /\*用来保存系统状态的字符\*/ |
|  |

|  |
| --- |
| unsigned long gul\_sub\_code\_len = 0; /\*课题代码表长度\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| HANDLE gh\_std\_out; /\*标准输出设备句柄\*/ |
|  |

|  |
| --- |
| HANDLE gh\_std\_in; /\*标准输入设备句柄\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int LoadCode(char \*filename, char \*\*ppbuffer); /\*代码表加载\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int CreatList(SCHOOL\_NODE \*\*pphead); /\*数据链表初始化\*/ |
|  |

|  |
| --- |
| void InitInterface(void); /\*系统界面初始化\*/ |
|  |

|  |
| --- |
| void ClearScreen(void); /\*清屏\*/ |
|  |

|  |
| --- |
| void ShowMenu(void); /\*显示菜单栏\*/ |
|  |

|  |
| --- |
| void PopMenu(int num); /\*显示下拉菜单\*/ |
|  |

|  |
| --- |
| void PopPrompt(int num); /\*显示弹出窗口\*/ |
|  |

|  |
| --- |
| void PopUp(SMALL\_RECT \*, WORD, LABEL\_BUNDLE \*, HOT\_AREA \*); /\*弹出窗口屏幕信息维护\*/ |
|  |

|  |
| --- |
| void PopOff(void); /\*关闭顶层弹出窗口\*/ |
|  |

|  |
| --- |
| void DrawBox(SMALL\_RECT \*parea); /\*绘制边框\*/ |
|  |

|  |
| --- |
| void LocSubMenu(int num, SMALL\_RECT \*parea); /\*主菜单下拉菜单定位\*/ |
|  |

|  |
| --- |
| void ShowState(void); /\*显示状态栏\*/ |
|  |

|  |
| --- |
| void TagMainMenu(int num); /\*标记被选中的主菜单项\*/ |
|  |

|  |
| --- |
| void TagSubMenu(int num); /\*标记被选中的子菜单项\*/ |
|  |

|  |
| --- |
| void SetHotPoint(HOT\_AREA \*phot\_area, int hot\_num); /\*设置热区\*/ |
|  |

|  |
| --- |
| void RunSys(SCHOOL\_NODE \*\*pphd); /\*系统功能模块的选择和运行\*/ |
|  |

|  |
| --- |
| BOOL ExeFunction(int main\_menu\_num, int sub\_menu\_num); /\*功能模块的调用\*/ |
|  |

|  |
| --- |
| void CloseSys(SCHOOL\_NODE \*phd); /\*退出系统\*/ |
|  |

|  |
| --- |
| BOOL ShowModule(char \*\*pString, int n); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL LoadData(void); /\*数据加载\*/ |
|  |

|  |
| --- |
| BOOL SaveData(void); /\*保存数据\*/ |
|  |

|  |
| --- |
| BOOL BackupData(void); /\*备份数据\*/ |
|  |

|  |
| --- |
| BOOL RestoreData(void); /\*恢复数据\*/ |
|  |

|  |
| --- |
| BOOL ExitSys(void); /\*退出系统\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL MaintainSchInfo(void); |
|  |

|  |
| --- |
| BOOL MaintainTeamInfo(void); /\*维护学生基本信息\*/ |
|  |

|  |
| --- |
| BOOL MaintainSubjectInfo(void); /\*维护学生缴费信息\*/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL QurrySchoolP(void);//查询学校负责人 |
|  |

|  |
| --- |
| BOOL QurrySchoolName(void);//查询学院名 |
|  |

|  |
| --- |
| BOOL QurryTeamName(void);//查询团队名 |
|  |

|  |
| --- |
| BOOL QurryProfAcc(void);//查询教师数量 |
|  |

|  |
| --- |
| BOOL QurrySubID(void);//查询项目编号 |
|  |

|  |
| --- |
| BOOL QurryTeamBelong(void);//查询所属团队 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL StatPro2Stu(void);//统计教师比学生 |
|  |

|  |
| --- |
| BOOL StatSubjectAcc(void);//统计学院项目数 |
|  |

|  |
| --- |
| BOOL StatESubject(void);//统计优秀项目 |
|  |

|  |
| --- |
| BOOL StatSub2Pro(void);//统计项目比教师 |
|  |

|  |
| --- |
| BOOL Other(void);//显示学院信息 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL About(void);//其他 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| SCHOOL\_NODE \*SearchPIC(SCHOOL\_NODE\*, char\*);//查找学院负责人 |
|  |

|  |
| --- |
| BOOL SeekSchoolName(SCHOOL\_NODE\*, char\*);//查找学院名 |
|  |

|  |
| --- |
| BOOL SeekTeamName(SCHOOL\_NODE\*, char\*);//查找团队信息 |
|  |

|  |
| --- |
| SUBJECT\_NODE\* SeekProID(SCHOOL\_NODE\*, char\*);//查找项目编号 |
|  |

|  |
| --- |
| TEAM\_NODE\* MatchProID\_1(SCHOOL\_NODE\*hd, char\* id);//匹配项目编号第一步 |
|  |

|  |
| --- |
| SUBJECT\_NODE\* MatchProID\_2(SCHOOL\_NODE\*hd, char\* id);//匹配项目编号第二步 |
|  |

|  |
| --- |
| SCHOOL\_NODE \* MatchTeamNode\_Step\_1(SCHOOL\_NODE \* hd, char \* team);//匹配团队节点第一步 |
|  |

|  |
| --- |
| TEAM\_NODE \* MatchTeamNode\_Step\_2(SCHOOL\_NODE \* hd, char \* team);//匹配团队节点第二步 |
|  |

|  |
| --- |
| SCHOOL\_NODE \*MatchSchoolName(SCHOOL\_NODE\*hd, char \* school);//匹配学院名 |
|  |

|  |
| --- |
| TEAM\_NODE \*MatchTeamName(SCHOOL\_NODE \*, char\*);//匹配团队名 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void FullFillSchInfo(SCHOOL\_NODE\*);//填写学院信息 |
|  |

|  |
| --- |
| void FullFillTeamInfo(TEAM\_NODE\*);//填写团队信息 |
|  |

|  |
| --- |
| void FullFillProInfo(SUBJECT\_NODE\*);//填写项目信息 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL DelProInfo(void);//删除项目信息 |
|  |

|  |
| --- |
| BOOL DelTeamNode(void);//删除团队节点 |
|  |

|  |
| --- |
| BOOL DelSchoolNode(void);//删除学院节点 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| BOOL InsertProInfo(void);//插入项目节点 |
|  |

|  |
| --- |
| BOOL InsertTeamNode(void);//插入团队节点 |
|  |

|  |
| --- |
| BOOL ModifyProInfo\_2(void);//修改项目信息 |
|  |

|  |
| --- |
| BOOL ModifyTeamInfo(void);//修改学院信息; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void SortSchoolInfo\_1(SCH\_DATA\* hd);//教师比学生排序 |
|  |

|  |
| --- |
| void SortSchoolInfo\_2(SCH\_DATA\* hd);//项目数排序 |
|  |

|  |
| --- |
| void SortTeamInfo\_1(TEAM\_DATA\*);//优秀项目数排序 |
|  |

|  |
| --- |
| void SortTeamInfo\_2(TEAM\_DATA\*);//项目比老师排序 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| void StatSchInfo(SCHOOL\_NODE \* hd); |
|  |

|  |
| --- |
| void statistic\_1(SCHOOL\_NODE \* hd); |
|  |

|  |
| --- |
| void statistic\_2(SCHOOL\_NODE \* hd); |
|  |

|  |
| --- |
| void statistic\_3(SCHOOL\_NODE \* hd); |
|  |

|  |
| --- |
| void statistic\_4(SCHOOL\_NODE \* hd); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| TEAM\_NODE\* SeekProID\_2(SCHOOL\_NODE\*hd, char \*id); |
|  |

BOOL SaveSysData(SCHOOL\_NODE\*);