

附件 1:

学 号:	0121213640126
------	---------------

武汉理工大学

《面向对象程序设计》
大作业

题 目	学生成绩管理系统
学 院	文法学院
专 业	教育学
班 级	教育学 1201
姓 名	杨欣
指导教师	鄢红国

2013 年 12 月 20 日

目 录

一	设计目的.....	1
二	大作业的内容.....	2
三	大作业的要求与数据.....	3
四	大作业应完成的工作.....	4
五	总体设计（包含几大功能模块）.....	5
六	详细设计（各功能模块的具体实现算法——流程图）.....	6
七	调试分析（包含各模块的测试用例，及测试结果）.....	7
八	总结.....	8
十	参考资料.....	9

一 大作业的目的

《面向对象程序设计》是一门实践性很强的课程，通过大作业不仅可以全方位检验学生知识掌握程度和综合能力，而且还可以进一步加深、巩固所学课程的基本理论知识，理论联系实际，进一步培养自己综合分析问题和解决问题的能力。更好地掌握运用 C++ 语言独立地编写、调试应用程序和进行其它相关设计的技能。

二 大作业的内容

对图书信息(包括编号、书名、总入库数量、当前库存量、已借出本数等)进行管理,包括图书信息的输入、输出、查询、删除、排序、统计、退出.将图书的信息进行记录,信息内容包含:(1)图书的编号(2)图书的书名(3)图书的库存量。假设,现收集到了一个图书馆的所有图书信息,要求用 C 语言编写一个简单的图书管理系统,可进行录入、查询、修改和浏览等功能。学习相关开发工具和应用软件,熟悉系统建设过程。

三 大作业的要求与数据

1、用C语言实现系统;

2、对图书信息(包括编号、书名、总入库数量、当前库存量、已借出本数)进行管理,包括图书信息的输入、输出、查询、删除、排序、统计、退出.

3、图书信息包括:其内容较多,为了简化讨论,要求设计的管理系统能够完成以下功能:

(1) 每一条记录包括一本图书的编号、书名、库存量

(2) 图书信息录入功能:(图书信息用文件保存,可以一次完成若干条记录的输入。)

(3) 图书信息显示浏览功能:完成全部图书记录的显示。

(4) 查询功能:完成按书名查找图书记录,并显示。

(5) 图书信息的删除:按编号进行图书某图书的库存量.

(6) 借书登记系统:可以输入读者编号和所借书号来借书。

(7) 还书管理系统:可以输入读者编号和所借书号来还书。

(8)、应提供一个界面来调用各个功能,调用界面和各个功能的操作界面应尽可能清晰美观!

四 大作业应完成的工作

- (1) 编写算法;
- (2) 算法测试, 并有具体的测试结果和结果分析;
- (3) 撰写大作业报告。内容包括:
 - 1) 封面扉页
 - 2) 设计任务书
 - 3) 目录
 - 4) 总体设计方案(包括流程图)
 - 5) 使用说明及运行实例(包括界面)
 - 6) 成员设计内容简介
 - 7) 项目源程序代码(注意注释的使用)
 - 8) 指导教师评语

五 总体设计(包含几大功能模块)

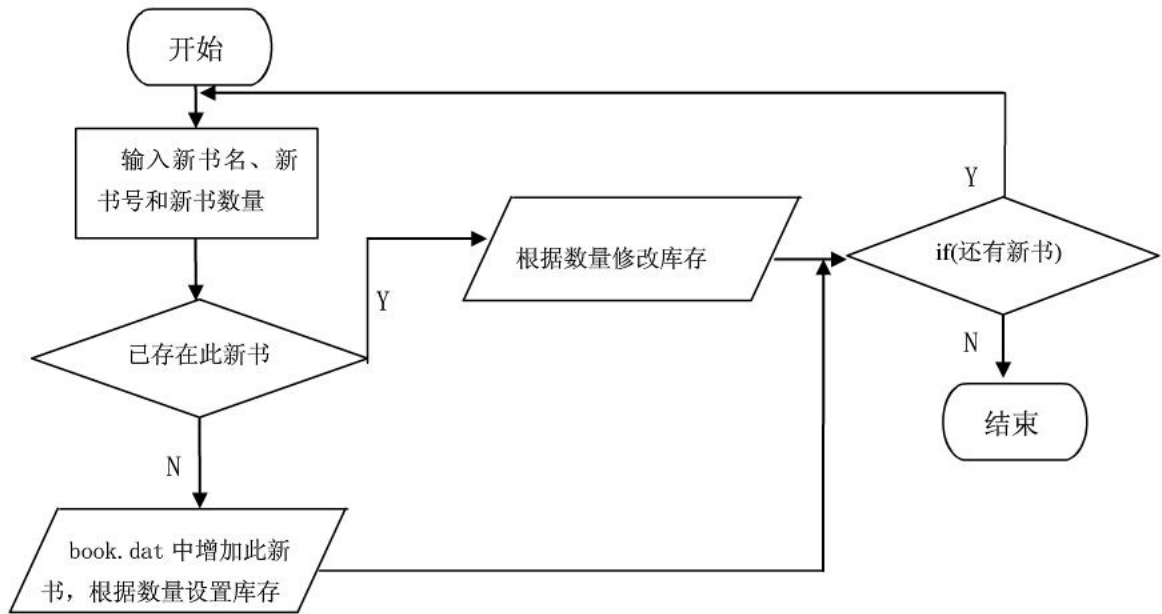
- 1、新书入库模块;
- 2、查询模块;
- 3、借阅管理模块;
- 4、借书登记模块;
- 5、还书管理模块;
- 6、输出信息模块;
- 7、退出系统模块;

六 详细设计(各功能模块的具体实现算法——流程图)

6.1 各函数的功能和实现

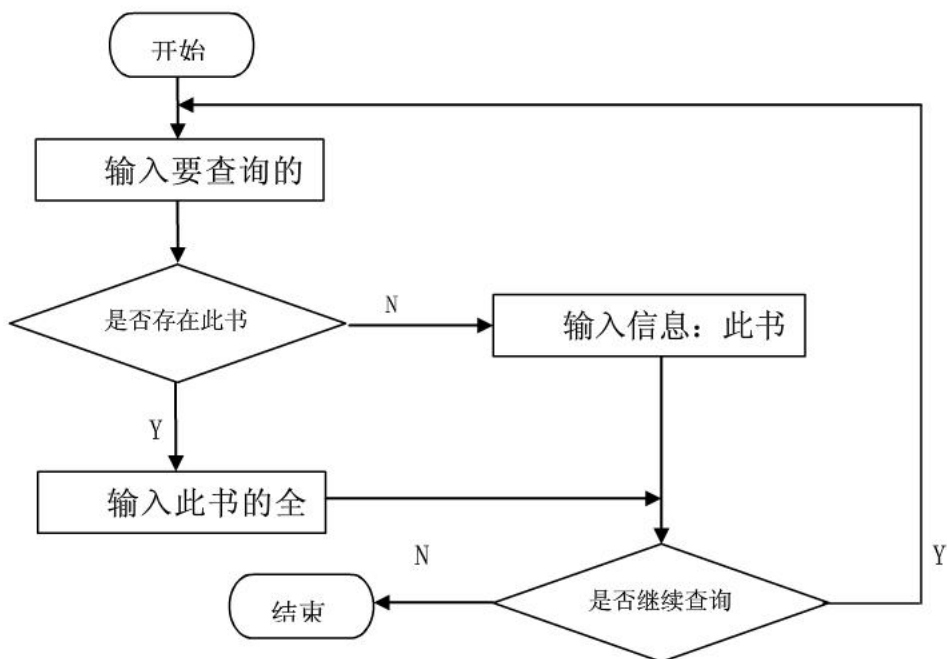
1、新书入库模块:

形参分别为: 书名, 书号, 以及有多少本书要入库, 再根据要求将信息写入文件 boot.dat 中。又由于最开始要输入三本新书, 故可以加一个初始化函数 init_book(), 但是初始化函数只执行一次, 也就是在系统第一次运行的时候执行, 此时可以根据判断是否存在 boot.dat 这个文件来决定是否执行初始化函数。



2、查询模块：

可用一个函数 `void find_book(char book_num[])` 来实现。找到就输出此书全部信息，找不到就输出查找失败信息。3、成绩查询功能



3、借阅管理模块：

用初始化函数 `init_reader()` 来实现对读者的初始化。因为读者初始化函数也只是在系统第一次运行的时候执行一次，故也可以根据是否存在 `reader.dat` 文件来确定是否执行此函数。（到现在可以发现在系统初始化函数里，至少已经需

要调用两个初始化函数了)

4、借书登记模块:

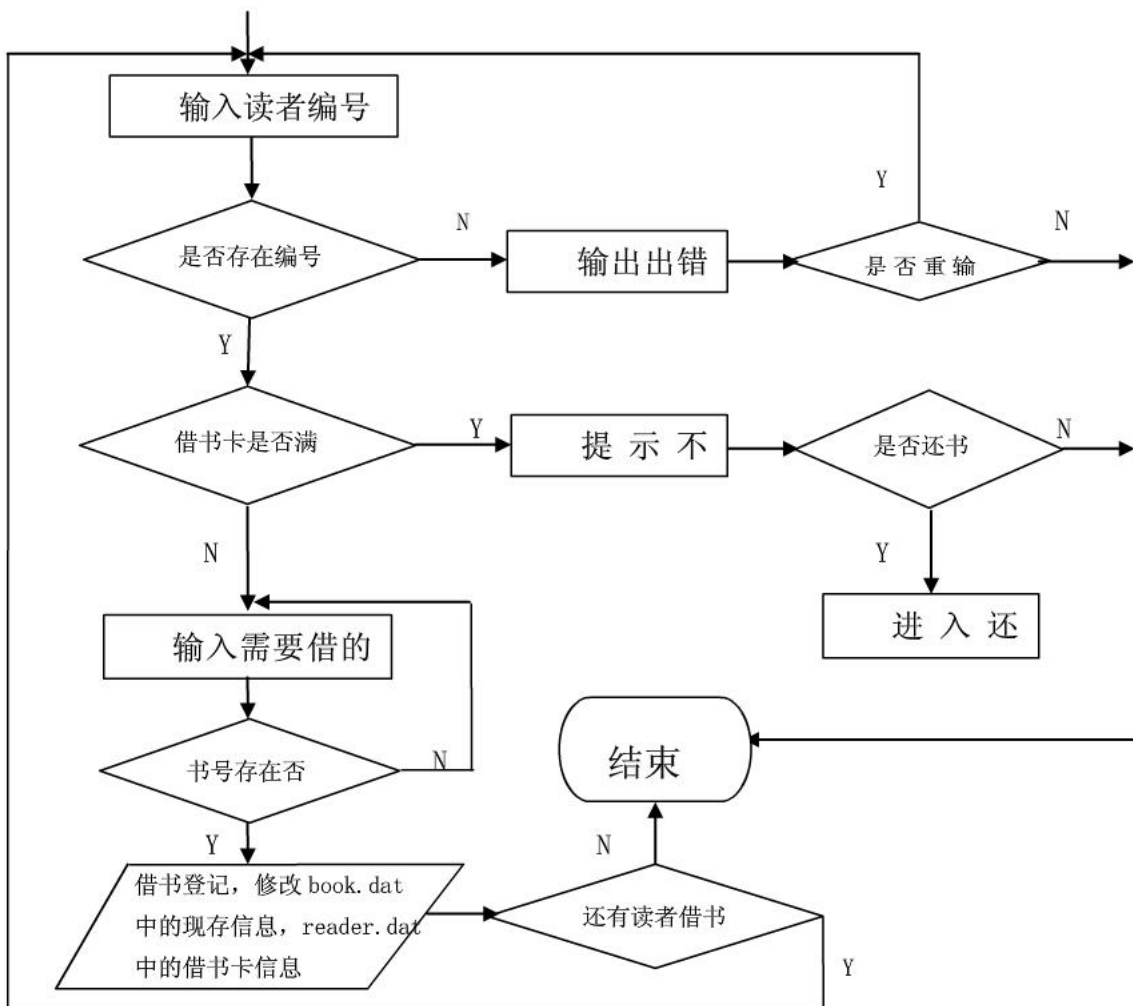
可用一个函数 `borrow_book(char reader_num[], char book_num[])` 来实现。在函数中要分别判断

A 该读者的编号是否已经存在, 如果不存在, 则提示错误。

B 该读者的借阅书是否已满 (即最多只能借 1 本书) 如果是, 则提醒该用户先去还书

C 输出的书号对应的书是否存在, 如果不存在, 则提醒读者不存在此书。

借书过程完成后, 则对读者信息里的借书信息写上所借书号, 并修改库存量。

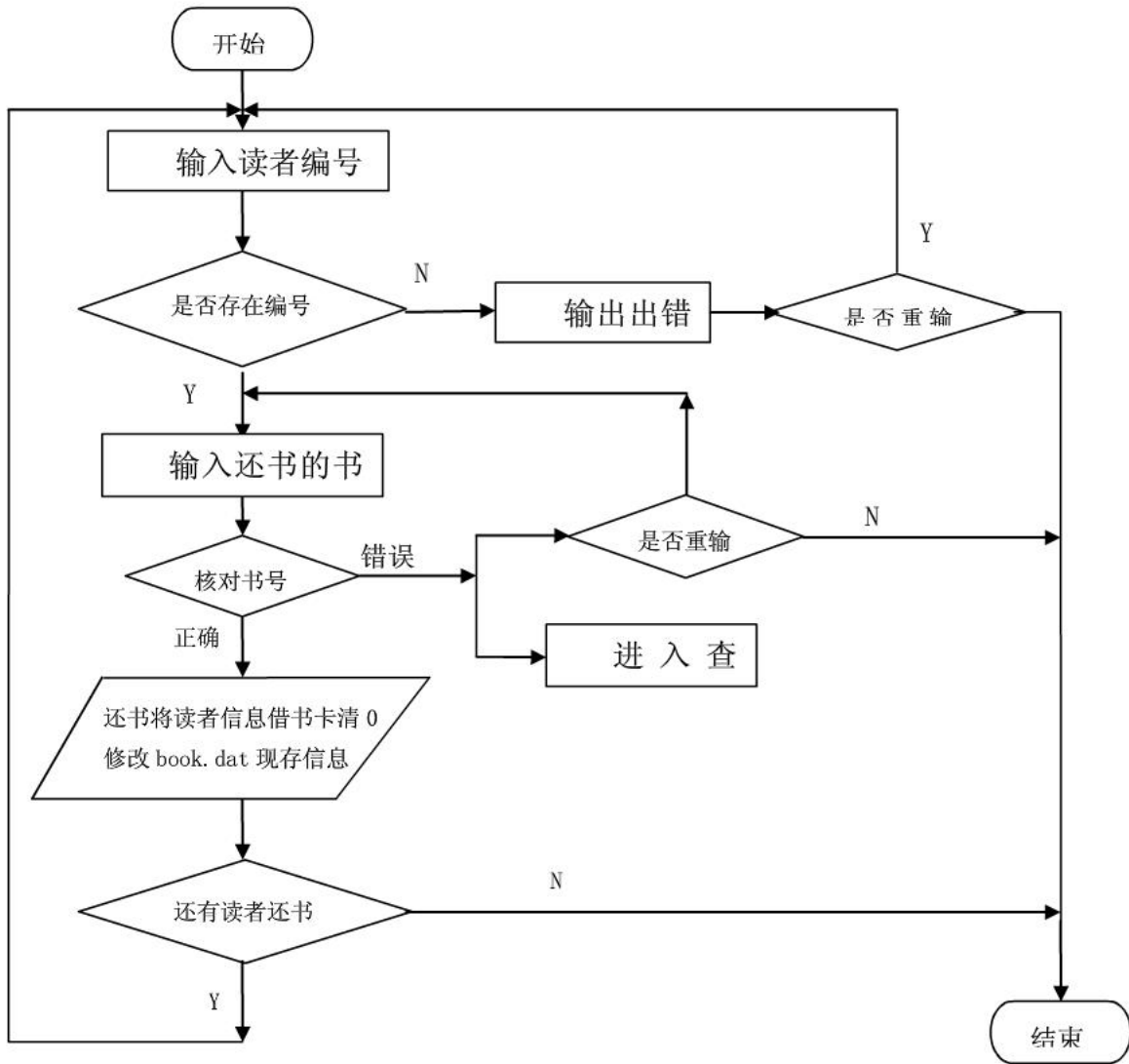


5、还书管理模块:

可用一个函数 `return_book(char reader_num[], char book_num[])` 来实现。在实现此函数时要判断:

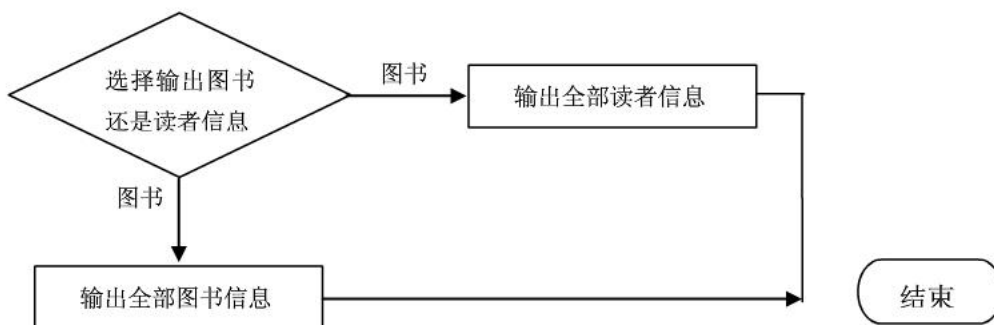
A 该读者的姓名是否已经存在, 如果不存在, 则提示错误。

- B 该读者是否已经借阅此书。如果没有，则提示错误。
- C 还书过程完成后，则对读者信息里的借书信息清零，并修改库存量。



6、输出信息模块：

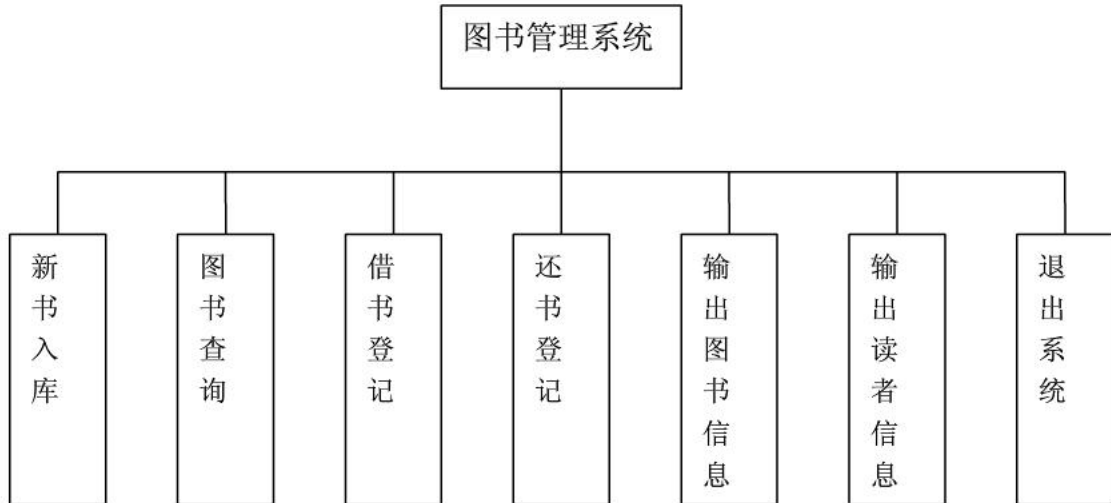
可分别用模块 print_book() 和 print_reader() 来实现直接从 book.dat 文件和 reader.dat 文件中读取信息并输出。



7、退出系统模块：

可用一个函数 `exit()` 来实现。首先将信息保存到文件，释放动态创建的内存空间，再退出此程序。

如图 6.1



七 调试分析（包含各模块的测试用例，及测试结果）

7.1 源程序

```
#include <stdio.h>
#include <math.h>
#include <string.h>
typedef struct bk {
    char book_num[11];
    char book_name[41];
    int book_kc;
    int book_xc;
    struct bk * next;
}book; /* 图书信息*/
typedef struct rd {
    char reader_num[11];
    char reader_name[11];
    char reader_book_num[11];
    struct rd * next ;
}reader; /*读者信息*/
struct bk * h_book,* h_reader; /*全局变量，分别为指向图书信息链表和读者信息链表的头指针*/
void Form_Insert_New_Book(); /*新书入库菜单*/
void Form_Find_Book(); /*查询图书菜单 */
void Form_Borrow_Book(); /*借书菜单*/
```



```

void Form_Return_Book(); /*还书菜单*/
void Find_Book(char *); /*查询图书*/
void Insert_New_Book(char *, char *, int ); /*新书入库, 即在链表中插入
一个新节点*/
void Borrow_Book(char *, char *); /*借书, 修改读者借阅信息和图书的现存信
息*/
void Return_Book(char *, char *); /*还书, 修改读者借阅信息和图书的现存信
息*/
void print_book(struct bk *); /*输出全部图书信息*/
void print_reader(struct rd *); /*输出全部读者信息*/
void load_data(); /*从文件中读取数据, 链表头指针指向第一个节点*/
void init_reader(); /*第一次运行程序时对三个读者的初始化*/
void init_book(); /*第一次运行程序时对三本图书的初始化*/
void save(); /*将链表中的信息保存到文件中*/
void M_Form() /*主菜单*/
{ char ch;
  printf("\n                图书管理系统主窗口");
  printf("\n                0:退出系统");
  printf("\n                1:新书入库");
  printf("\n                2:图书查询");
  printf("\n                3:借书登记");
  printf("\n                4:还书管理");
  printf("\n                5:输出全部信息");
do { /*此循环用来保证选择正确的数字键执行相应的功能*/
  printf("\n\n                请选择相应的功能代码: ");
  scanf("%c",&ch);
  fflush(stdin); /*用于清除键盘缓冲区中的多余字符*/
  }while(ch!='0' &&ch!='1' &&ch!='2' &&ch!='3' &&ch!='4' &&ch!='5');
  switch(ch) /*根据输入的字符进行选择执行相应的模块*/
  { case '0':exit(0);
    case '1':Form_Insert_New_Book();break;
    case '2':Form_Find_Book();break;
    case '3':Form_Borrow_Book();break;
    case '4':Form_Return_Book();break;
    case
'5':load_data();print_book(h_book);print_reader(h_reader);M_Form();
    default:printf("error\n");exit(0);
  }
}
void Form_Find_Book()
{ char ch;
  char book_num[11];
  printf("\n                图书查询功能模块, 请选择相应的功能代码:
\n");

```

```

printf("                0:返回到主菜单\n");
printf("                1:图书查询");
do{
    printf("\n\n                请选择相应的功能代码：");
    scanf("%c",&ch);
    fflush(stdin);
}while(ch!='0' &&ch!='1');
switch(ch)
{ case '0':M_Form();break; /*选择0, 则返回到上一级目录, 即主菜单*/
  case '1':
    { printf("                \n 请输入需要需要查询的书的编号:");
      scanf("%s",book_num);
      fflush(stdin);
      Find_Book(book_num); /*执行查询*/
      Form_Find_Book();/*查询完毕后继续返回到查询窗口, 执行下一个查询*/
    }
  default:printf("error\n");exit(0);
}
}
void Find_Book(char book_num[]) /*查询图书模块, 根据图书编号查询*/
{ struct bk *p;
  p=h_book;
  while(p!=NULL&&strcmp(p->book_num,book_num)!=0)
p=p->next;/*此循环用于查找, 如果在中间找到, 则 p 就不可能到末尾指向空*/
  if(p==NULL) printf("\n\t 此编号对应的图书不存在! \n");
  else { printf("\n\t 图书编号\t 图书名\t 库存量\t 现存量    \n");

printf(" %10s\t %10s\t %d\t %d\t\n",p->book_num,p->book_name,p->b
ook_kc,p->book_xc);
    }
  getch();
}
void init_book() /*初始化图书模块*/
{ book ba[3];
  FILE *fp;
  int i;
  printf("\n    系统即将进行初始化, 图书初始化模块:\n");
  printf("\n    请分别输入三本书的书号, 书名, 数量 (以空格隔开): \n
");
  for(i=0;i<3;i++) /*输入 3 本图书进行初始化*/
    scanf("%s%s%d",ba[i].book_num,ba[i].book_name,&ba[i].book_kc);
  for(i=0;i<3;i++) {ba[i].book_xc=ba[i].book_kc;} /*开始时库存=现存*/
  if((fp=fopen("book.dat","wb"))==NULL)/*建立新文件 book.dat 准备写入
数据*/

```



```

{ printf(" %10s\t %10s %10d\t%10d\n", p->book_num, p->book_name, p
->book_kc, p->book_xc);
    p=p->next;
}
}
void print_reader(struct rd *h)
/*输出读者信息*/
{ struct rd *p;
    p=h;
    printf("\n\t读者编号 \t读者姓名 \t所借书号 \n");
    while(p!=NULL)

{ printf(" %10s\t %10s %10s\n", p->reader_num, p->reader_name, p
->reader_book_num);
    p=p->next;
}
    getch();
}
void Form_Borrow_Book()
/*借书菜单界面*/
{ char ch;
    char book_num[11], reader_num[11];
    printf("\n 图书借阅功能模块，请选择相应的功能代码：
\n");
    printf(" 0: 返回到主菜单\n");
    printf(" 1: 图书借阅\n");
    do{
        printf("\n\n 请选择相应的功能代码：");
        scanf("%c",&ch);
        fflush(stdin);
    }while(ch!='0' &&ch!='1');
    switch(ch)
    { case '0':M_Form();break;
      case '1':
        { printf(" \n 请输入读者编号与所借书编号：(以空格分隔)");
          scanf("%s%s", reader_num, book_num);
          fflush(stdin);
          Borrow_Book(reader_num, book_num);/*调用借书函数，执行借书功能*/
          Form_Borrow_Book();/*借书完成后，程序返回到借书菜单*/
        }
        default:printf("error\n");exit(0);
    }
}
}
void Borrow_Book(char reader_num[], char book_num[])

```



```

{ case '0':M_Form();break;
  case '1':
    { printf("      \n 请输入读者编号与所还书编号:(以空格分隔)");
      scanf("%s%s",reader_num,book_num);
      fflush(stdin);
      Return_Book(reader_num,book_num);/*调用还书函数*/
      Form_Return_Book();
    }
  default:printf("error\n");exit(0);
}

}

void Return_Book(char reader_num[],char book_num[])
/*输入读者编号和所还书编号执行还书*/
{ struct bk *book_p;
  struct rd *reader_p;
  book_p=h_book;
  while(book_p!=NULL&&strcmp(book_p->book_num,book_num)!=0)
    book_p=book_p->next;
  if(book_p==NULL)
    { printf("\n\t 此编号对应的图书不存在! \n");
      Form_Return_Book();
    }
  reader_p=h_reader;

while(reader_p!=NULL&&(strcmp(reader_p->reader_num,reader_num)!=0))
  reader_p=reader_p->next;
if(reader_p==NULL)
  { printf("\n\t 此编号对应的读者不存在! \n");
    Form_Return_Book();
  }
  else if(strcmp(reader_p->reader_book_num,book_num)!=0)
    { printf("\n\t 读者没有借阅此书\n");
      Form_Return_Book();
    }
    else {
book_p->book_xc=book_p->book_xc+1;/*还书后,修改图书库存量*/
strcpy(reader_p->reader_book_num,"0");/*还书后,修改读者借阅信息*/
save();/*保存到文件*/
load_data();/*读入文件中的数据*/}
  Form_Return_Book();
}

void init_reader()
/*系统第一次运行时,初始化3位读者信息,从键盘读入*/

```

```

{ FILE *fp;
  reader ra[3];
  int i;
  printf("\n    系统即将进行读者初始化:\n");
  printf("\n    请分别输入三位读者的编号和姓名 (以空格隔开): \n
");
  for(i=0;i<3;i++)
    scanf("%s%s", ra[i].reader_num, ra[i].reader_name);
  for(i=0;i<3;i++)
    strcpy(ra[i].reader_book_num, "0");
  if((fp=fopen("reader.dat", "wb"))==NULL)
    { printf("不能建立读者文件, 初始化失败! 请重新启动系统! \n");
return; }
  for(i=0;i<3;i++)/*将 3 位读者信息写入文件*/
  { if(fwrite(&ra[i], sizeof(struct rd), 1, fp)!=1)
    { printf("\写数据错误! \n"); exit(0); }
  }
}
void Form_Insert_New_Book()
/*添加新书菜单*/
{ char ch;
  struct bk t;
  printf("\n                新书入库功能模块, 请选择相应的功能代码:
\n");
  printf("                0: 返回到主菜单\n");
  printf("                1: 新书入库");
  do{printf("\n\n                请选择相应的功能代码: ");
    scanf("%c", &ch);
    fflush(stdin);
  }while(ch!='0' &&ch!='1' &&ch!='2' &&ch!='3' &&ch!='4' &&ch!='5');
switch(ch)
{ case '0':M_Form();break;
  case '1':
  { printf(" \n 请按顺序输入新书的编号、名称、数量 (以空格隔开): ");
    scanf("%s%s%d", t.book_num, t.book_name, &t.book_kc);
    fflush(stdin);
    Insert_New_Book(t.book_num, t.book_name, t.book_kc);
    Form_Insert_New_Book();
  }
  default:printf("error\n");exit(0);
}
}
void load_data()
/*从文件 book.dat 和文件 reader.dat 中读取数据到内存*/

```

```

/*将读取的数据存放到 2 个链表中*/
{ struct bk *book_p1,*book_p2,*book_p3;
  struct rd *reader_p1,*reader_p2,*reader_p3;
  FILE *fp_book,*fp_reader;
  fp_book=fopen("book.dat","rb");
  book_p1=(struct bk *)malloc(sizeof(struct bk));
  fread(book_p1,sizeof(struct bk),1,fp_book);
  h_book=book_p3=book_p2=book_p1;
  while(! feof(fp_book))
  { book_p1=(struct bk *)malloc(sizeof(struct bk));
    fread(book_p1,sizeof(struct bk),1,fp_book);
    book_p2->next=book_p1;book_p3=book_p2;book_p2=book_p1;
  }
  book_p3->next=NULL;
  free(book_p1);
  fclose(fp_book);
  fp_reader=fopen("reader.dat","rb");
  reader_p1=(struct rd *)malloc(sizeof(struct rd));
  fread(reader_p1,sizeof(struct rd),1,fp_reader);
  h_reader=reader_p3=reader_p2=reader_p1;
  while(! feof(fp_reader))
  { reader_p1=(struct rd *)malloc(sizeof(struct rd));
    fread(reader_p1,sizeof(struct rd),1,fp_reader);

reader_p2->next=reader_p1;reader_p3=reader_p2;reader_p2=reader_p1;
  }
  reader_p3->next=NULL;
  free(reader_p1);
  fclose(fp_reader);
}
void save()
/*将链表中的数据保存到内存中*/
{ FILE *book_fp,*reader_fp;
  struct bk *book_p;
  struct rd *reader_p;
  book_p=h_book; reader_p=h_reader;
  if((book_fp=fopen("book.dat","wb"))==NULL)
  { printf("不能建立图书文件，初始化失败！请重新启动系统！\n");
return; }
  while(book_p!=NULL)
  { if(fwrite(book_p,sizeof(struct bk),1,book_fp)!=1)
    { printf("\写数据错误！\n"); exit(0); }
    book_p=book_p->next;
  }
}

```



```

fclose(book_fp);
if((reader_fp=fopen("reader.dat", "wb"))==NULL)
    { printf("不能建立图书文件, 初始化失败! 请重新启动系统! \n");
return; }
while(reader_p!=NULL)
    { if(fwrite(reader_p, sizeof(struct rd), 1, reader_fp)!=1)
        { printf("\写数据错误! \n"); exit(0); }
        reader_p=reader_p->next;
    }
fclose(reader_fp);
}
main()
{ FILE * fp;
  struct bk temp;
  clrscr();
  h_book=NULL;h_reader=NULL;
  if((fp=fopen("book.dat", "r"))==NULL)
    init();/*第 1 次运行系统时的初始化*/
  fclose(fp);
  load_data();/*从文件中读入数据*/
  M_Form();/*显示主菜单*/
  save();/*保存内存中的数据(即链表中的数据)到文件*/
  getch();
}

```

7.2.1 算法调试、测试过程中出现的问题及解决方法

本系统调试过程中遇到问题、原因和解决方法如下:

- (1) 程序在储存图书信息时只能一次存入, 并且读取。再次添加新的程序时将会把原来添加的图书信息覆盖。原因是存储函数只能只写 2 进制文件不能追加填写 2 进制文件。
- (2) 运行完程序一次有错误提醒
原因是上次运行程序后没有关闭操作界面当点击编译时会有一个错误提示, 解决方法: 将上一次运行时的操作界面关闭。
- (3) 程序不能运行
程序运行到添加新图书时自动提示错误需要调试, 其原因是指针指向的位置出现错误导致程序传参错乱无法运行, 解决办法: 使程序进入调试状态在传参出现错误的位置一句一句查找传递的参数是否是你所

希望的，如若不是就加以修正。

问题：借书管理模块中，当正确输入存在的读者编号和所借的书的编号，系统进行判断时，提示不存在此读者。

解决办法及步骤：

- 1、一个个输出所有的读者的编号，检查文件中是否有此读者，发现有。
- 2、既然有此读者，那么检查循环判断是否有此读者的语句发现没有错
- 3、输出用于循环检查语句中的读者信息，发现乱码
- 4、仔细分析乱码的原因，最后发现是变量的类型错误，错将读者类型的结构体指针变量定义为了图书类型的指针变量。

7.2.2 主要程序运行结果

点击运行，首先是进入菜单选择界面，可进行如图 1 所示。



图 1 进入菜单选择界面

输入 1 回车进入新书入库功能系统，可进行如图 2 所示。



图 2 进入新书入库选择界面

输入 1 回车输入新书功能系统，可进行如图 3 所示。

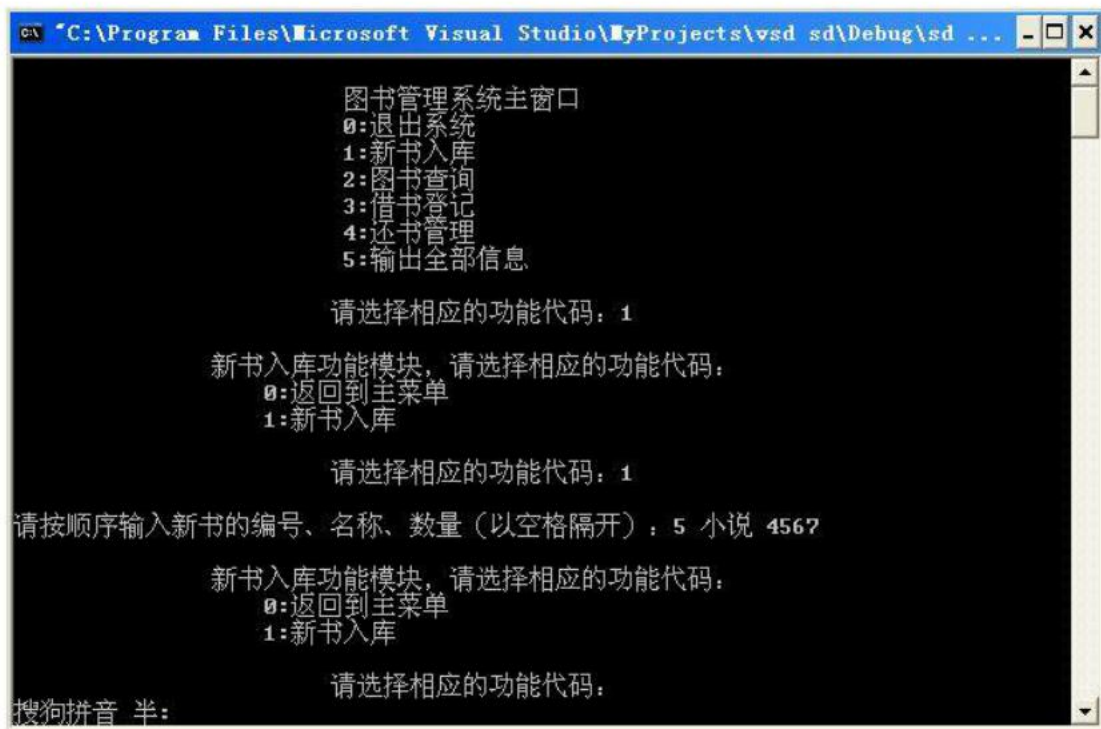


图 3 进入新书入库选择界面

输入 2 回车进入图书查询功能系统，可进行如图 4 所示。



图 4 进入新书入库选择界面

输入 1 回车开始时图书查询系统，可进行如图 5 所示。

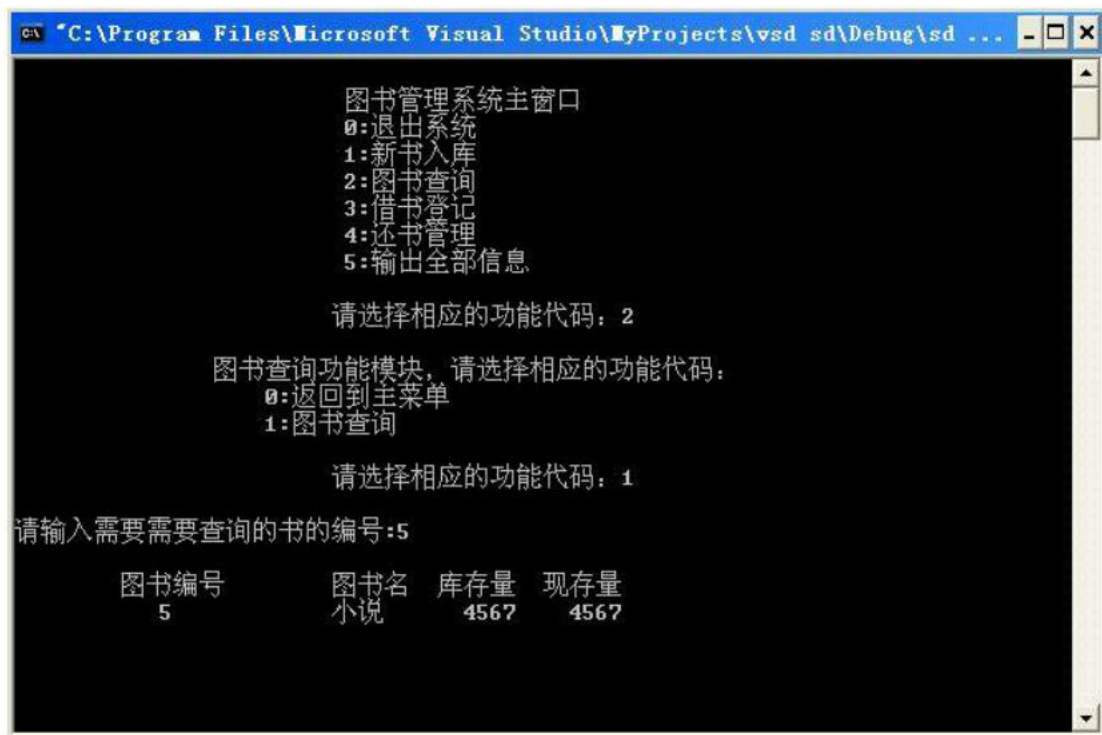


图 5 进入图书查询界面

输入 3 回车借书登记系统, 可进行如图 6 所示

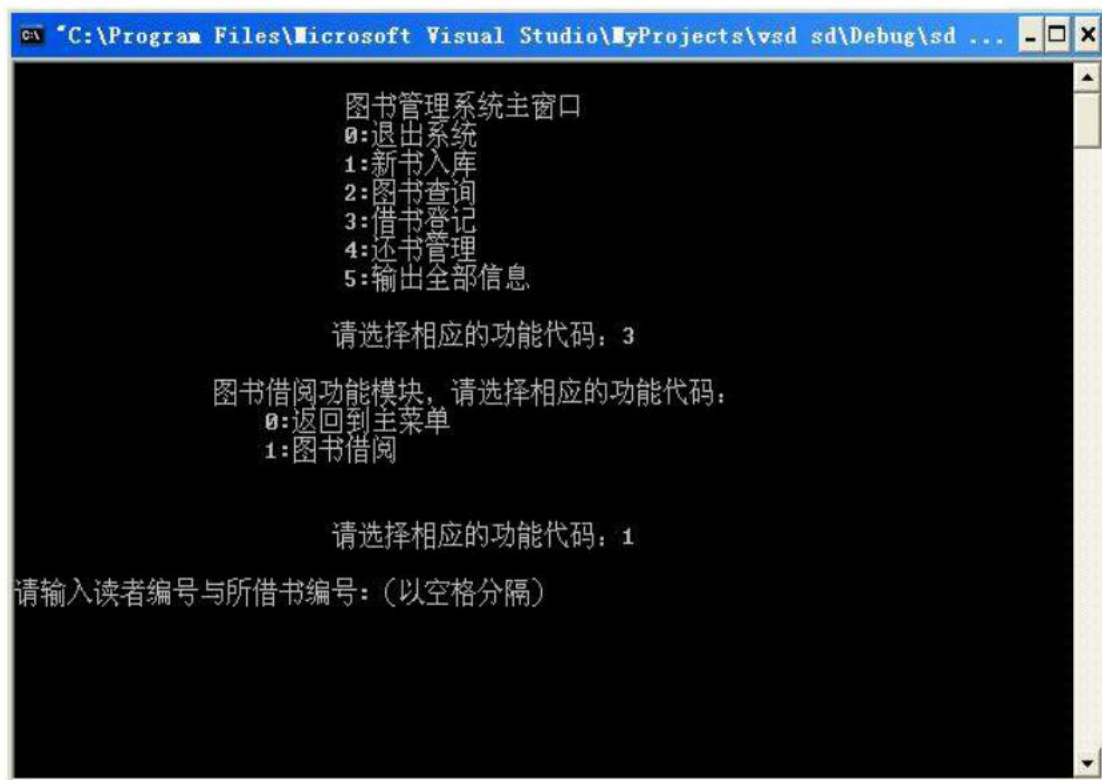


图 6 进入借书登记界面

输入 4 回车还书管理系统，可进行如图 7 所示



图 7 进入图书查询界面

输入 5 回车显示信息系统，可进行如图 8 所示



图 8 输出信息界面

八 总结

通过这次课程设计，我对复杂程序的开发有了一个深刻的认识，使我进一步明白了模块化思想在程序开发中的重要性。

由于对程序算法不是很熟练，开始设计出来的菜单不是预想的那样，而是整个窗中出现混乱。解决的这个问题的办法是调整。一个系统的菜单和提示信息非常重要。如果没有这些用户根本不知道怎么用你设计的这个系统。在设计的调试过程中也无法顺利的完成调试工作。有了一个清晰简单的菜单和一些提示信息这后，调试过程完成的非常顺利。

回顾起此次大作业的完成过程，我感慨颇多，的确，从拿到题目到完成整个编程，从理论到实践，在整整两个礼拜的日子里，可以学到很多很多的东西，同时不仅可以巩固了以前所学过的知识，而且学到了很多在书本上所没有学到的知识。通过这次大作业使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。

第一次做难免会遇到过各种各样的问题，同时在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固。

这不仅是程序设计，更是锻炼我们处理问题的能力，同时也使我了解到编写程序是件细心活，稍不留神就会出错，这就必须要求对待事情要认真！在编写程序的过程中，错误不断出现，不同的类型（如少写了一个符号，写错了字母，用错了函数等等）层出不穷，这考验待事细心，耐心，能不能坚持到底，不能半途而废。

我总结出了一点点的经验如下：

- 1、要对系统的功能和要求做出详细的分析，并合理分解任务。
- 2、把分解出来的子任务，做给一个相对独立的模块。
- 3、在设计一个模块之前，要简单构想一下总界面的显视情况。
- 4、针对构想出来的界面进行程序的编写。

十 参考资料

- [1]谭浩强. C 程序设计(第三版). 北京: 清华大学出版社, 2005
- [2]谭浩强. C 程序设计题解与上机指导(第三版). 北京: 清华大学出版社, 2005
- [3]网上相关资料

本科生大作业成绩评定表

班级：

姓名：

学号：

序号	评分项目	满分	实得分
1	学习态度认真、遵守纪律	10	
2	设计分析合理性	10	
3	设计方案正确性、可行性、创造性	20	
4	设计结果正确性	40	
5	设计报告的规范性	10	
6	设计验收	10	
		总得分/等级	

评语：

注：最终成绩以五级分制记。优（90-100分）、良（80-89分）、中（70-79分）、及格（60-69分）、60分以下为不及格

指导教师签名：

2013 年 12 月 28 日