

计算机编程语言课程设计丛书

Visual C++ 课程设计

——案例精选与编程指导

陈清华 朱 红 主编

· 最新版 ·

东南大学出版社

· 南 京 ·

内 容 提 要

学习程序设计语言的主要目的是能够利用简洁的语句编写出高效、完整的实用程序。本书正是为了满足广大读者的这一实际需要而编写的。书中精选出大量涉及各个领域的优秀案例,使读者通过阅读、分析优秀程序以及在样例程序的引导下编写程序等手段,掌握编写大规模实用程序的方法和技巧。

本书共分 8 部分,内容包括域名服务器的仿真程序样例及样例详解,经典有趣的游戏程序,简单实用的数学问题求解程序,清晰易懂的矩阵及向量处理程序,实用有趣、可扩充的小型管理程序,获取及利用时间和日期的实用程序,有趣、可扩充的多媒体应用程序,有一定难度的数据结构应用程序。书中精选的程序实用性强,语言简练,注释通俗易懂。程序组织结构及对问题的处理方法多种多样,覆盖面广,对读者编写大型实用程序有一定的借鉴作用。

本书是专门为大专院校理工科学生进行 C++ 程序设计语言课程设计而编写的教材,对广大的计算机编程爱好者而言也是一本难得的实用参考书。

图书在版编目(CIP)数据

Visual C++ 课程设计——案例精选与编程指导 / 陈清华,
朱红主编. —南京:东南大学出版社,2003.6
(计算机编程语言课程设计丛书)
ISBN 7-81089-275-4

I. V... II. ①陈... ②朱... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 041923 号

东南大学出版社出版发行
(南京四牌楼 2 号 邮编 210096)
出版人:宋增民
江苏省新华书店经销 南京玉河印刷厂印刷
开本:787 mm×1092 mm 1/16 印张:21.75 字数:557 千字
2005 年 2 月第 1 版修订本第 1 次印刷
本书定价:30.00 元(随书附赠光盘一张)
(凡因印装质量问题,可直接向发行部调换。电话:025-83795801)

计算机编程语言课程设计丛书

编委会名单

主任:李文忠

副主任:朱 敏 陈清华

编委:(以姓氏笔画为序)

丁晟春	王必友	王姗姗	王 栋
冯继生	卢雪松	叶锡君	李文忠
江国华	朱 敏	朱敦铭	邵定宏
宗 平	林 庆	陈利群	陈清华
张基温	周 群	张 煦	顾元刚
郑成增	施 诚	赵明生	唐 全
秦 军	徐逢辰	夏 斐	傅德胜

前 言

学了汉语的字、词、成语及语法,你未必能写出好的文章。但是优秀的文章读多了,你就能写出好的文章。而写出好文章的人,未必都是从文学院科班毕业的。

如果你不会游泳,即使请来国家游泳队的总教练,在岸上将蛙泳的每个动作要领都讲解得清清楚楚,你也把每个动作铭记在心,做得准确无误,而你正要跃跃欲试时,将你扔到深水里,你一定会被淹死。

本书主要目的是通过引导读者在阅读、分析、理解精选的优秀样例程序的基础上,学会一步一步设计完整程序的基本方法。提高读者用 C++ 程序设计语言来解决问题的能力,进而掌握编写高效、完整、面向对象的实用程序的设计方法,及编写大规模实用程序的方法和技巧。

本书中的优秀样例程序是从近 10 万行程序代码中精心挑选出来的。全书共分 8 部分,内容包括域名服务器的仿真程序样例及样例详解,6 个经典有趣的游戏程序,7 个简单实用的数学问题求解程序,6 个清晰易懂的矩阵及向量处理程序,5 个实用有趣的小型管理程序,2 个简单实用的时间和日期处理程序,5 个多媒体应用程序,以及 6 个实用但有一定难度的数据结构应用程序。

书中精选的程序实用性强,语言简练,注释通俗易懂。程序组织结构及对问题的处理方法多种多样,覆盖面广,对读者编写大型实用程序有一定的借鉴作用。

如果能把本书中的样例程序研读一遍,你就知道别人是如何写出优秀的程序的。如果你能在读懂三个以上样例程序的前提下,并按照题目中提出的“课程设计要求”去做,且能很好完成,你已经成为编程高手,这将使你在工作中受益无穷。

本书是专门为大专院校理工科学生进行 C++ 程序设计语言课程设计而编写的教材,对广大的计算机编程爱好者也是一本难得的实用参考书。

本书主编为陈清华和朱红,主要参编人员有朱近、靳丛、刘永、戴益民、王庆宝、刘明、钱芸生、赵琦、张峰、闫玉德。

本书能够正式出版,首先要感谢“计算机编程语言课程设计丛书编委会”的所有教授及专家,特别是东南大学的朱敏教授、南京邮电学院的秦军教授、中国药科大学的唐全教授以及南京航空航天大学的王珊珊教授,更要感谢南京理工大学张岳新教授、李晓梅、梅景春、陆汉栋、李涛、宗士增等相关领导,他们为本书的前期策划及编写提出了很多宝贵意见。

本书是编者总结多年来从事计算机教学的经验,结合实际科研开发的成果,几经修订编写成的。但由于时间仓促,书中难免有不当之处,恳请读者不吝指正。

陈清华

Cqh608@yahoo.com

2003 年 4 月于南京理工大学

使 用 说 明

本书可以用作“Visual C++课程设计”的教材,也可以用作“Visual C++程序设计”课程大作业及上机编程参考书。如果作为课程设计教材使用,以下使用说明可供参考。

一、各学校根据课程进度和学时数来安排学生选题。每个设计程序按难易程度被分为 A 或 B 两个等级,原则上 B 级题为单选题,A 级题可合做。建议由学生自由选题,但原则上不能重复。如果是二人合做一题,则教师要给学生规定好各自的工作,保证学生课程设计的重复。所附光盘中“补充部分”文件夹中的题也可选做。

二、原则上学生在进行课程设计时应独立完成,以培养其分析问题和解决问题的能力。教师可以安排一定的时间答疑。

三、要求学生必须利用一定的时间(可以利用假期)针对所选题目进行分析、注释。分析清楚程序主要完成的功能,如果有类,注释清楚类的功能、类中的数据变量用途、类中的函数定义;注释清楚每个函数的功能、函数的参数、函数的返回值含义;注释清楚整个程序中,各代码段之间的前后运行关系、函数的前后调用关系;注释清楚程序中的每一行语句的具体含义。

四、学生在完成课程设计的基本功能要求之后,可根据各自的兴趣对所选设计题目的功能进行扩充,对程序结构进行改写。

五、本书和光盘上提供的代码程序仅作参考。所附光盘中为每个题目提供了一个代码程序。在进行课程设计时,可将包含设计题目的文件夹复制到硬盘中,去掉所有文件的只读属性,如果是第 1,2,3,4,5,6,8 部分的题目建议在 VC6.0 环境下编译运行,如果是第 7 部分的题目建议在 Turbo C++ 2.0 以上环境下编译及运行。

六、编程时,要求变量名、函数名、类名规范,语句嵌套采用缩进格式,尽量采用面向对象的设计方法。请参见附录 1 和附录 2。

七、课程设计结束时,学生应提交完成的所有源程序及相关文件。同时撰写《课程设计报告》,其中主要包括以下内容:对题目的分析、注释,所增加功能模块的设计及设计中遇到的主要问题及解决方法,设计中尚存的不足,对设计的感想和心得体会等。

八、教师在考核学生的设计时,应增加面试过程。这样可以全面准确地了解学生对基础知识的掌握和对设计题目所做的工作,同时杜绝作弊现象的发生。因为题目的难易程度差别较大,第 8 部分设计题目相对较难,考核时应予以充分考虑。

九、如果有程序代码不在随书光盘中,请访问东南大学出版社网站 <http://press.seu.edu.cn>。

目 录

1	样例及样例解答	(1)
	样例 域名服务器的仿真程序	(2)
	样例解答 域名服务器的仿真程序	(9)
2	游戏程序	(49)
	第 1 题 扑克牌游戏	(50)
	第 2 题 TICTACTOE 游戏(三子连珠)	(58)
	第 3 题 21 点的扑克牌游戏	(64)
	第 4 题 猜单词游戏	(71)
	第 5 题 福彩游戏	(83)
	第 6 题 对抗游戏	(88)
3	数学问题求解程序	(93)
	第 1 题 计算器	(94)
	第 2 题 分数计算器	(100)
	第 3 题 复数计算器	(108)
	第 4 题 八皇后问题	(110)
	第 5 题 求解方程	(115)
	第 6 题 迭代法求解线性方程	(126)
	第 7 题 字符串操作	(132)
	第 8 题 四则运算	(140)
4	矩阵及向量处理程序	(145)
	第 1 题 矩阵旋转反射	(146)
	第 2 题 矩阵乘法计算	(151)
	第 3 题 矩阵数乘	(153)
	第 4 题 矩阵转置与乘法计算	(160)
5	小型管理程序	(161)
	第 1 题 工资管理	(162)
	第 2 题 物品库存管理	(170)
	第 3 题 股票交易系统	(179)
	第 4 题 个人财政支出管理	(193)
	第 5 题 电话簿管理	(200)
6	时间和日期处理程序	(213)
	第 1 题 时间和日期	(214)
	第 2 题 日期格式处理	(234)
7	多媒体应用程序	(245)



第 1 题	屏保程序	(246)
第 2 题	牛顿法图解方程	(253)
第 3 题	图形运动程序	(256)
第 4 题	特殊图像效果	(260)
第 5 题	窗口菜单显示	(269)
8	数据结构应用程序	(275)
第 1 题	链表数组堆栈	(276)
第 2 题	双向链表	(289)
第 3 题	二叉树解决四则运算问题	(300)
第 4 题	二叉树操作	(308)
第 5 题	模板实现排行榜管理	(319)
第 6 题	模板实现电话簿管理	(328)
附录 1	变量命名应遵守的原则	(334)
附录 2	VC++ 编程应遵守的原则	(335)
	参考文献	(337)

样例 域名服务器的仿真程序

1.1.1 程序功能简介

一个域名服务器的仿真(模拟)程序。一个简单的仿真模拟程序,其中包括数据的输入、查询和删除功能,要求在该程序的基础上添加一些功能,使程序的功能更为全面,操作更加方便。

1.1.2 课程设计要求

(1) 本程序每个功能只能完成一次操作,若进行同样的操作需退回到主菜单中方可进行,增加可以多次进行同样操作的功能。

(2) 每次程序执行完后数据不能保存下来,应该将输入的数据存放在一个文件中,在每个数据记录格式中添加时间单元,存入添加数据时的系统时间,并可以根据不同的时间范围(某时间之前、某时间之后、某段时间之间)来查找域名。

(3) 将数据保存在文件中,增加打开原有文件和建立新文件的功能。

(4) 增加修改原有数据的功能,在修改数据时,应将新旧数据同时在屏幕上显示,并询问是否以新数据替换旧数据,按“确认”键后再将数据替换,否则放弃。

1.1.3 课程设计说明

(1) 本程序以链表的方式存储数据,基本结构比较合理,不需修改,应该在原有结构的基础上增加一些函数。

(2) 评定难易级别:A级。

1.1.4 关键源代码注解

```
#include <stdlib.h>
#include <time.h>
#include <iostream.h>
#include <ctype.h>
#include <string.h>
#include <fstream.h>
#include <stdio.h>

struct webaddress //地址簿格式
{
    char www[5];
    char name[9];
    char adm[5];
    char country[4];
    webaddress * next; //增加指针变量,作为链表进行操作
};
```

```

webaddress * head=NULL;    //全局变量,链表头
void Initial(void)
{
    cout<<" * * * * * \n";
    cout<<"    欢迎使用域名服务器查程序          \n";
    cout<<"        请选择以下功能                    \n";
    cout<<" \n";
    cout<<" 1. 查询数据                                \n";
    cout<<" 2. 插入数据                                \n";
    cout<<" 3. 删除数据                                \n";
    cout<<" 4. 打印全部数据                            \n";
    cout<<" 5. 退出                                    \n";
    cout<<" \n";
    cout<<" * * * * * \n";
}

void Change(webaddress * pt1,char * TempChar)
{
    int i,j;
    int dot=0;
    static char str[10];
    for(j=0;TempChar[j]! =\0;j++)
    {
        TempChar[j]=tolower(TempChar[j]);    //将输入的字符串变成小写
        if(TempChar[j]='.')
            dot++;        //统计数
    }
    if(dot==0)
    {
        cout<<"数据格式错误,未插入数据! \n";
        return;
    }
    for(i=0;i<4;i++)
        str[i]=TempChar[i];
    str[i]=\0;
    if(strcmp(str,"www.")==0)
    {
        strcpy(pt1->www,str);
        dot--;
        j=i;
    }
    else

```

```

        {
            j=0;
            strcpy(pt1->www,"");
        }
        for(i=0;TempChar[j]! ='\0'&&TempChar[j]! ='\0';i++,j++)
        {
            str[i]=TempChar[j];
        }
        str[i]='\0';
        j=j+1;
        dot--;
        strcpy(pt1->name,str);
        for(i=0;TempChar[j]! ='\0'&&TempChar[j]! ='\0';i++,j++)
        {
            str[i]=TempChar[j];
        }
        str[i]='\0';
        j=j+1;
        strcpy(pt1->adm,str);
        for(i=0;TempChar[j]! ='\0'&&.dot;i++,j++)
            str[i]=TempChar[j];
        str[i]='\0';
        strcpy(pt1->country,str);
        return;
    }
}

void Insert(webaddress * pt)
{
    webaddress * pt1, * pt2;
    pt1=pt2=head;
    if (head == NULL )
    {
        head = pt;
        pt->next = 0;
        return ;
    }
    if (strcmp( head->name, pt->name)>=0 )
    {
        pt->next=head;
        head=pt;
        return;
    }
    pt2=pt1 = head;
    while ( pt2->next && strcmp(pt2->name, pt->name)<0 )

```

```
{
    pt1 = pt2 ;
    pt2 = pt2->next;
}
if (strcmp( pt2->name, pt->name)<0 )
{
    pt2->next = pt;
    pt->next =0;
}
else
{
    pt->next = pt2;
    pt1->next = pt;
}
return ;
}

void Print(webaddress * pt)
{
    cout<<pt->www<<pt->name<<'. '<<pt->adm<<'. '<<pt->country<<endl;
}

void LookUpData(void)
{
    webaddress * pt;
    char webname[10];
    cout<<"输入域名的名称\n";
    cin>>webname;
    pt= head;
    while(pt)
    {
        if(strcmp(webname,pt->name)==0)
        {
            cout<<"查找出的域名为:\n";
            Print(pt);
            break;
        }
        pt=pt->next;
    }
    if(pt==NULL)
        cout<<"数据文件中没有该地址\n";
    cin.get();
    cin.get();
}
```

```

return;
}

void InsertData(void)
{
    webaddress * pt;
    char Tempchar[22];
    cout<<"请输入要插入的域名\n";
    cin.get();
    cin.getline(Tempchar,22);
    pt=new webaddress;
    Fill(pt,Tempchar);
    Insert(pt);
    return;
}

void Delete(char * Tempchar)
{
    webaddress * p1, * p2;
    char choice;
    if ( head == NULL )
    {
        cout <<"数据表为空,无域名可删! \n";
        return ;
    }
    if ( strcmp(head->name,Tempchar)==0 )
    {
        p1=head;
        cout<<"存在,该域名为:\n";
        Print(p1);
        cout<<"请确定是否删除该域名(y/n)? \n";
        cin>>choice;
        if(choice=='Y' || choice=='y')
        {
            head = head->next;
            delete p1;
            cout <<"删除了该域名! \n";
        }
        else
        {
            cout<<"未删除域名! \n";
        }
    }
}

```

```
else
{
    p2=p1 = head;
    while (strcmp(p2->name,Tempchar) !=0 && p2->next != NULL )
    {
        p1 = p2 ;
        p2 = p2->next;
    }
    if (strcmp( p2->name,Tempchar) ==0 )
    {
        cout<<"存在,该域名为:\n";
        Print(p2);
        cout<<"请确定是否删除该域名(y/n)? \n";
        cin>>choice;
        if(choice=='Y' || choice=='y')
        {
            p1->next = p2->next;
            delete p2;
            cout <<"删除了该域名! \n";
        }
        else
        {
            cout<<"未删除域名! \n";
        }
    }
    else cout <<"数据表上没有找到要删除的域名! \n";
}
return;
}

void DeleteData(void)
{
    char Tempchar[22];
    cout<<"请输入要删除的域名的名称(第二个字段)\n";
    cin.get();
    cin.getline(Tempchar,22);
    Delete(Tempchar);
    cin.get();
    cin.get();
    return;
}
```



```

void PrintData(void)
{
    webaddress * pt;
    pt = head;
    if(head != 0)
        cout<<"数据表中的域名有:\n";
    while(pt)
    {
        Print(pt);
        pt = pt->next;
    }
}

void main(void)
{
    int choice;
    do
    {
        Initial();
        cin>>choice;
        switch(choice)
        {
            case 1:
                LookUpData();    break;
            case 2:
                InsertData();    break;
            case 3:
                DeleteData();    break;
            case 4:
                PrintData();    getchar();break;
            case 5:
                break;
            default:
                cout<<"选择错误! \n";
        }
    }while(choice != 5);
}

```

样例解答 域名服务器的仿真程序

1.2.1 原程序各个功能说明

1) 原程序的结构

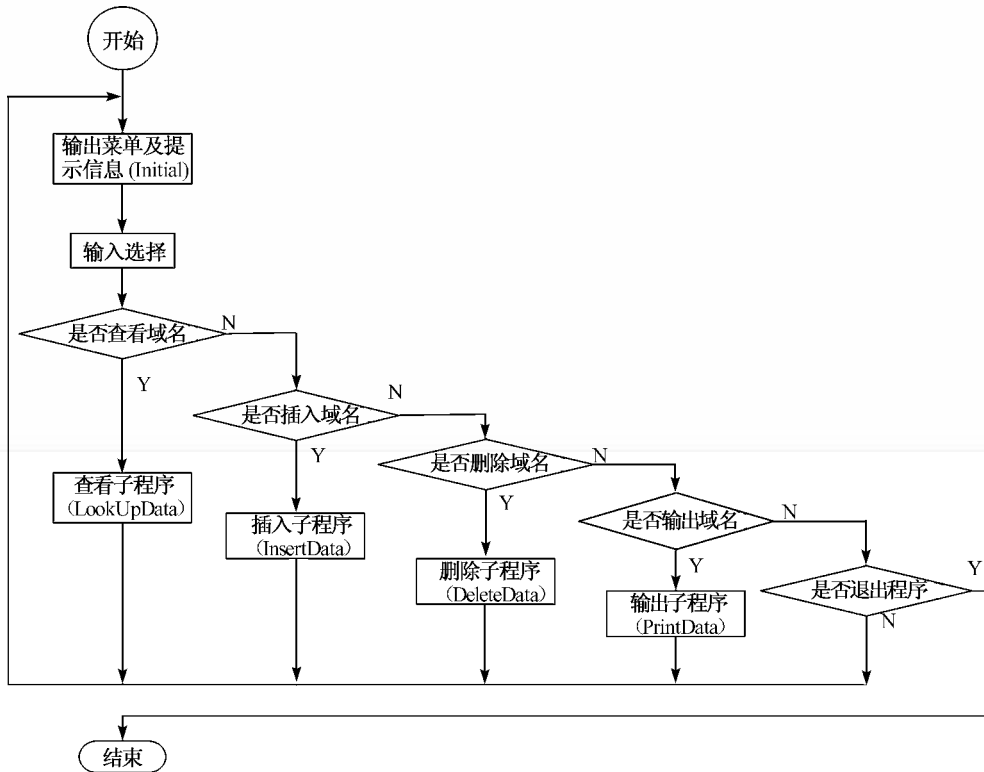


图 1.1 域名服务器结构框图

这是原程序的框图,可以看出,域名服务器程序是一个大的循环结构,每种功能都用菜单选项列出,用户可以根据需要选择相应的菜单项,从而执行不同的子程序以完成对应的操作。

比如插入子程序的结构框图如下:

可以看出,插入子程序调用了两个函数,一个是将域名(字符串)转换为结点的 Change 函数,另一个是单向有序链表的通用插入函数 Insert。在对数据的其他操作功能中还多次调用了 Change 函数,这个函数在后面再进行详细分析。

2) 链表说明

原程序用链表的方式存储数据,各个子程序都是针对同一个链表的不同操作。链表存储数据有诸多好处,它可以根据需要动态开辟存储空间,有较大的灵活性,且算法固定通用。查找、建立、删除、

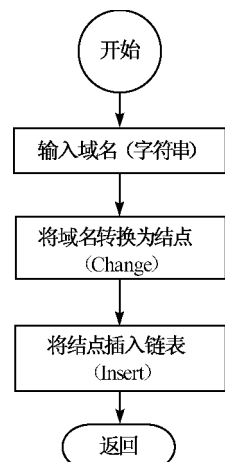


图 1.2 插入子程序框图

输出等操作都比较方便,下面着重分析本程序涉及的链表的具体操作。由于链表的操作结合了指针和结构体类型,所以对于初学者而言,可能不太好掌握,其实链表操作的算法比较固定,无论我们编写程序或阅读程序,有关链表操作的要点是:先解决中间情况,再考虑一头一尾。在这里用链表的插入操作来举例说明,首先考虑结点插入到链表中间(除去一头一尾)的情况,这是一般操作,将这个一般的算法掌握后,再考虑两种特殊情况,即插入到头结点前或插入到尾结点后,这样,插入的算法就基本上掌握了。当然,根据具体的操作可能还有一些特殊情况,如遇到的是空链表等,但这些特殊情况一般来说相对容易理解些。同理,删除、查找等操作也是这种思路。

结点的构成:结点是一个结构体类型,由两部分构成的,第一部分是数据部分,可以由各种类型的数据组成,第二部分是指针部分,这部分存放下一个结点的地址。本程序的结点结构定义如下:

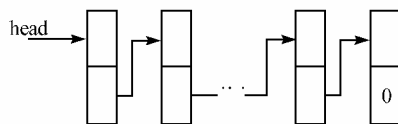
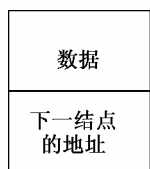


图 1.3.1 结点结构

图 1.3.2 链表结构

定义语句为:

```
struct webaddress //结构体,地址簿格式,结点类型
{
    char www[5]; //存放字符串"www.",若省略该字符串,则为空
    char name[9]; //存放机构名,排序等均以此为依据
    char adm[5]; //存放网络名,不可省略
    char country[4]; //存放国家名,可省略
    webaddress * next; //结构体指针变量,存放下一结点地址
};
```

链表头指针也是一个结构体类型的指针,指示链表头结点的地址,链表尾结点的指针部分内容为空,值为 0 表示空,即指针不再指向其他结点。这样,只要知道链表头指针就可以对整个链表进行操作。

(1) 链表结点的插入(排序)操作

所谓链表结点的插入是指对有序链表而言的,将结点按一定的顺序链接起来,首先考虑插入到链表中间的情况(而不是插入到头结点前或尾结点后)。首要的问题是要找到插入点,即新增结点插入到链表中的哪个结点后面,这是一个循环查找的过程。查找循环进行下去有两个条件,一是没有找到合适的插入点,一个是没有查找完全部链表,只有当这两个条件全部满足时,才能继续进行查找。在这里,用指针 pt 来表示指向欲插入链表的结点,假定链表的结点是以机构名的升序来排序的,因此,这个循环判断可用下面语句实现:

```
webaddress * pt1, * pt2;
pt1=pt2=head; //定义 pt1,pt2 是结构体的指针变量,并从链表头开始操作
while ( pt2->next && strcmp(pt2->name, pt->name)<0 ) /* pt2 沿着链表依次指
```

向各个结点,当 pt2 未指到尾结点且不满足插入条件时(即欲插入数据的机构名大于 pt2 指向结点的数据的机构名)循环。*/

```
{
    pt1 = pt2;    //将 pt2 指向的结点地址赋给 pt1
    pt2 = pt2->next; //pt2 指向链表的下一结点
}
```

该循环图示如下:

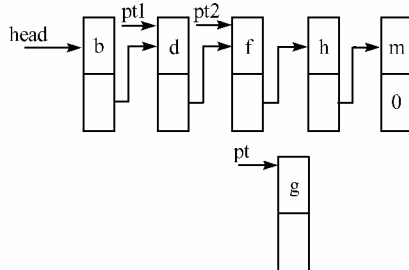


图 1.4 查找插入点循环示意图

由此可见,pt2 寻找第一个机构名大于 pt 所指结点的机构名的结点,如图 1.4 所示。当寻找到满足上述条件的结点时,链表及指针的分布情况如图 1.5 所示。

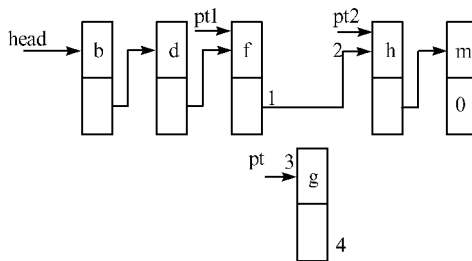


图 1.5 插入点在 pt1 和 pt2 之间

可以看出,pt 指向的结点应插入到 pt1 和 pt2 指向的结点之间,即将 1,2 间的连线断掉,即对原来 $pt1 \rightarrow next$ ($pt1 \rightarrow next = pt2$)重新复制,然后将 1 与 3 连接起来,2 与 4 连接起来。如图 1.6 所示。

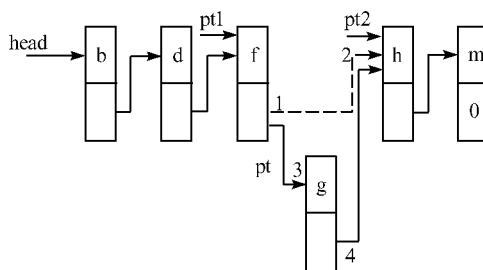


图 1.6 连接插入点

将 1 与 3 连接起来的语句是: $pt1 \rightarrow next = pt$; 将 4 与 2 连接起来的语句为: $pt \rightarrow next = pt2$; 所以应有以下语句: