深究函数的递归调用

王善发,吴道荣

(保山师范高等专科学校 计算机科学系,云南 保山 678000)

【摘 要】程序设计从低级语言到高级语言、由过程式设计向面向对象发展,目前很多程序还是面向过程的设计方式,而且 C 语言一直是面向过程程序设计的最主要的实现语言之一,也是学习面向对象程序设计的基础语言。C 语言已经成为高等学校计算机程序设计的基础学科,甚至成为各个高校理、工科学生的专业基础课。在讲授 C 语言程序设计中,函数的递归是相当难教、难学和难理解的重点内容。笔者观摩了许多老师讲授 C 语言的教学方法,在教学过程中不断的探索、改进,总结出了一些直观、形象、容易理解的教学方法。

【关键词】C 语言;程序设计;函数递归

【中图分类号】TP312 【文献标识码】A 【文章编号】1673-1891(2007)04-0067-04

1 背景资料

C 语言是一种高效而通用的编译型程序设计语言,功能丰富,表达能力强,描述问题能力强,运算速度快,使用灵活,应用面宽,编译容易实现,目标程序质量高,并且有良好的可移植性,能实现对系统硬件及外围接口设备的控制操作。因此,C 语言不仅适用于应用软件的开发,也适用于系统软件的开发。所有这些特点,使得运用 C 语言进行程序设计已经成为软件开发的一个主流[1]。近年来,C 语言无论在国内还是国外都得到了迅速推广。目前,中国所有高等学校基本上都开设了《C 语言程序设计》这门计算机高级语言学科。

C 语言功能强、使用灵活,但是普遍认为 C 语言比较难学,学习和使用 C 语言需要有一定的硬件知识和软件基础知识,不太熟悉 C 语言的程序员常常出错而不知其所以然[3]。在讲授 C 语言程序设计中,函数的递归是相当难教、难学、难理解的重点内容。如果老师在讲授中不用心思考,不讲究方法,平铺直叙,照本宣科。这样的教学,学生很难理解,不易接受。最后,一知半解,形成了学习《C 语言程序设计》中的障碍,甚至放弃学习这门学科。这样,不但没有达到发挥函数递归的简洁高效的作用,反而成了《C 语言程序设计》教学中的拦路虎。针对这种情况,首先要消除学习者的畏难心理,激发他们的学习兴趣。然后,总结别人的经验,发挥出让学生好理解,易接受的讲授方法,轻松顺利的完成《C 语言程序设计》教学中函数递归的阶段性教学。

2 函数的递归调用教学中出现的问题

在函数递归调用这一部分教学中,很多老师就书本上的概念来讲解,不把教学内容与生活实际相联系,不注意细节,或者是没有方法讲清楚函数递归调用的全过程。现在以讲授用递归方法求 n! 的例题为例分析教学中出现的问题。

例:用递归方法求 n!

$$n! = \begin{cases} 1 & (n = 0, 1) \\ n \cdot (n-1)! & (n > 1) \end{cases}$$

程序如下:

```
float fac(int n)
{ float f;
   if(n <0)
     printf("n <0, data error\n");
   else if(n== 0||n== 1)
     f = 1;
   else
     f = n * fac(n - 1);
   return(f);
```

收稿日期:2007 - 07 - 10

作者简介:王善发(1967 -),男,云南镇雄人,讲师,硕士,主要从事基于 J2EE 的信息管理系统研究。

```
}
main()
{    int n;
     float y;
     printf("input a integer number: ");
     scanf("%d",&n);
     y=fac(n);
     printf("%d! =%15.0f",n,y);
}
```

在讲授到被调用函数 fac(int n)

……

```
float fac(int n)
{    float f;
     if(n<0)
        printf("n<0,data error\n");
     else if(n==0||n==1)
        f=1;
     else
        f=n*fac(n-1);
     return(f);
}
```

……

这个递归函数时，学生认为 n **最后总是要到降到** 1 **的，这时，通过** if(n==0||n==1) f=1; **语句，而不执行** else f=n*fac(n-1); **这个语句，最后执行** return(f); **语句，最终返回的值是** 1**，没办法求出** n!**。**

这种理解是错误的，这是学生没有真正理解函数调用时，编译程序在调用函数开始调用的位置做了现场记录，保留了调用前的信息，当调用结束后再将被调用函数的结果返回到调用函数的调用语句的位置，再继续执行后面的程序。有的同学在实验室调试出结果后，也不清楚整个函数的递归调用过程。出现这种情况，老师怎样才能给学生讲解清楚呢？这需要老师认真思考、仔细研究，改变传统教学方法，采取具有技巧、策略的讲授方法，才能让学生比较清楚的理解。

3 技巧性的教学方法

笔者通过了几次采用不同的方法讲解，多次研究。总结出了比较通俗易懂，比较切合实际的讲授方法。

假如 n=3**，求** 3!**。**

```
main()
{    int n;
     float y;
     printf("input a integer number: ");
     scanf("%d",&n);
     y=fac(n);
     printf("%d! =%15.0f",n,y);
}
```

main()**中**

……

```
y=fac(3);
```

……

第一次 fac()**函数中** n=3

```
float fac(int n)
{   float f;
    if(n<0)
printf("n<0,data error\n");
   else if(n==0||n==1)
      f=1;
   else
      f=3*fac(3-1);
   return(f);
}
```

第二次 fac()**函数中** n=2

```
float fac(int n)
{   float f;
    if(n<0)
        printf("n<0,data error\n");
   else if(n==0||n==1)
        f=1;
   else
        f=3*2*fac(2-1);
   return(f);
}
```

第三次 fac()**函数中** n=1

```
float fac(int n)
{   float f;
    if(n<0)
        printf("n<0,data error\n");
    else if(n==0||n==1)
        f=1;
    else
```

```
        f = 3 * 2 * 1;
    return(f);
}
```

即：最后 return(f); **语句返回的值是** 6.0**，得到正确的结果，但是理解起来还是不怎么容易。这时我们把** fac(3); **语句、** fac(2); **语句和** fac(1); **语句都带进整个被调用函数的代码进去，情况就大不一样了，这样就直观、明了、显而易见。过程如下：**

y = fac(n); n = 3 **时**

```
main()
{     int n;
float y;
printf("input a integer number: ");
scanf("%d",&n);
y =
float fac(int n)
{   float f;
    if(n <0)
          printf("n <0, data error\n");
    else if(n== 0 || n== 1)
          f = 1;
    else
          f = 3 * fac(3 - 1);
    return(f);
}
;
printf("%d! = %15.0f", n, y);
}
```

y = fac(n); n = 2 **时**

```
main()
{   int n;
    float y;
    printf("input a integer number: ");
    scanf("%d",&n);
y =
float fac(int n)
{   float f;
    if(n <0)
          printf("n <0, data error\n");
    else if(n== 0 || n== 1)
          f = 1;
    else
          f = 3 * fac(3 - 1);
    return(f);
}
;
printf("%d! = %15.0f", n, y);
}
```

y = fac(n); n = 1 **时**

```
main()
{   int n;
float y;
printf("input a integer number: ");
scanf("%d",&n);
y =
float fac(int n)
{   float f;
    if(n <0)
          printf("n <0, data error\n");
    else if(n== 0 || n== 1)
          f = 1;
    else
          f = 3 * fac(3 - 1);
    return(f);
}
;
printf("%d! = %15.0f", n, y);
}
```

上面是回推的过程。

现在再回过头来看递推的过程：

由上面回推的最后一个过程，即： y = fac(n); n = 1 **时的过程递推得：**

y = fac(n); n = 2 **时**

```
main()
{   int n;
    float y;
    printf("input a integer number: ");
    scanf("%d",&n);
y =
float fac(int n)
{   float f;
    if(n <0)
          printf("n <0, data error\n");
    else if(n== 0 || n== 1)
          f = 1;
    else
          f = 3 * fac(3 - 1);
    return(f);
}
;
printf("%d! = %15.0f", n, y);
}
```

y = fac(n); n = 3 **时**

```
main()
{     int n;
float y;
printf("input a integer number: ");
scanf("%d",&n);
y =
float fac(int n)
{   float f;
    if(n <0)
          printf("n <0, data error\n");
    else if(n== 0 || n== 1)
          f = 1;
    else
          f = 3 * fac(3 - 1);
    return(f);
}
;
printf("%d! = %15.0f", n, y);
}
```

```
float fac(int n)
{   float f;
    if(n <0)
        printf("n <0, data error\n");
    else if(n==0 || n==1)
        f=1;
    else
        y = f=3*
            float fac(int n)
            {   float f;
                if(n <0)
                    printf("n <0, data error\n");
                else if(n==0 || n==1)
                    f=1;
                else
                    f=2*1;
                return(f);
            }
        ;
    return(f);
}
    printf("%d! =%15.0f", n, y);
}
```

由上面回推的倒数第二个过程，即：y = fac(n); n = 2 **时的过程递推得：**

```
main()
{   int n;
    float y;
    printf("input a integer number: ");
    scanf("%d", &n);
    y =
        float fac(int n)
        { float f;
        if(n <0)
            printf("n <0, data error\n");
        else if(n==0 || n==1)
            f=1;
        else
            f=3*2;
        return(f);
        }
    ;
    printf("%d! =%15.0f", n, y);
}
```

由上面回推的倒数第三个过程，即：y = fac(n); n = 3 **时的过程递推得：**

```
main()
{   int n;
    float y;
    printf("input a integer number: ");
    scanf("%d", &n);
    y=6;
    printf("%d! =%15.0f", n, y);
}
```

最后输出结果：

3! = 6.0

能理解 3! **的递归调用，也就不难理解** n! (n> 3) **的递归调用。通过这样讲授，学生就能完全理解函数递归调用的过程。**

4 结束语

在软件工程已经被公认为一门重要学科的今天，面向过程程序设计方式还没有退出程序和学习的重要课程。笔者希望得到各位同仁的理解和支持，一起讨论《C语言程序设计》的教学方法，通过教学实践，把教学经验、心得体会，设计的优秀程序写出来，让大家一起学习、探讨，提高教学质量。

参考文献：

[1]李编著. C语言程序设计与应用[M]. 昆明:云南大学出版社, 2002,6.

[2]谭浩强著. C语言程序设计[M]. 北京:清华大学出版社, 2000,1.

[3][美]KrisJamsa著. 张春晖, 刘大庆, 李越等译. C/C++/C#程序员实用大全[M]. 北京:中国水利水电出版社, 2002,10.

[4]徐士良主编. C常用算法程序集[M]. 北京:清华大学出版社, 1996,11.

[5]谭浩强著. C程序设计[M]. 北京:清华大学出版社, 1999,12.

[6]崔武子,赵重敏,李青著. C程序设计教程[M]. 北京:清华大学出版社, 2003,7.

[7]钱能主编. C++程序设计教程[M]. 北京:清华大学出版社, 1999,4.

[8]Bruce Eckel: Thinking in C++ (Second Edition), Volume One: Introduction to Standard C++. Chinese simplified language edition published by China Machine Press. Copyright2002 by China Machine Press.

[9]陈家骏,郑滔编著. 程序设计教程用C++语言编程[M]. 北京:机械工业出版社, 2004,8.

(下转74页)

参考文献：

[1][美]Steve Mack 编著．刑栩嘉等译．流媒体宝典[M]．北京：电子工业出版社，2003.

[2]吴国勇，邱学刚，万燕仔．网络视频流媒体技术与应用[M]．北京：北京邮电大学出版社，2001.

[3]钟小平，张金石编著．网络服务器配置与应用（第3版本）[M]．北京：人民邮电出版社，2007.

[4]马杰，田金义，柳键．流媒体技术及其文件格式[J]．计算机工程与应用，2003(23)：45－52.

Analysis and Application of Streaming Media System Based on WMS

HAN De[1]，WANG Yun－shan[2]

(Xichang College, Xichang, Sichuan 615013)

Abstract: On the base of a brief introduction to principle and standard protocol of Streaming media technology, and combining, with properties of Windows Media technology, from the perspective of practice, key technology and method are described in implementation of streaming media application system based on WMS.

Key words: Streaming media; Windows media services ; RTSP; Protocol rollover

（责任编辑：张荣萍）

（上接70页）

The Adaptation and Application of Recursion of Function on a Deep Level

WANG Shan－fa，WU Dao－rong

(The Department of Computer Science, Baoshan Normal College, Baoshan, Yunnan 678000)

Abstract: The program design has witnessed a change from low－grade language to advanced language and a development from process－based design to object－oriented design. Nevertheless, the former is still adopted by a considerable number of programs. The C programming language has always been major access to process－based design and the base of object－oriented design; therefore the C programming language is the basic course in most universities for science and technology majors. In expatiating the C programming language, the recursion of function has been a tough point. In view of this, the author of this paper concludes some intuitionistic and visual pedagogical ideas through emulating others' teaching and his exploring and improving.

Key words: The C programming language；Programming design；Recursion of function

（责任编辑：张荣萍）