第21卷 第1期 北京工业大学学报 Vol.21 No.1
1995年3月 JOURNAL OF BEIJING POLYTECHNIC UNIVERSITY Mar. 1994

DOS系统中C函数层次结构图的自动生成

易小琳 吴宇靖
（北京工业大学计算机学院，100044）

摘 要 介绍了微机DOS系统中自动生成用C语言开发的系统中各个函数之间的调用层次结构图及其函数模块功能的方法．较好地解决了在较大系统的更新、调试及维护阶段不易快速、准确地查找到系统中各个函数之间的调用关系及各个函数模块功能的问题．在系统开发过程中为软件系统开发者提供了便利的工具．

关键词 DOS系统，层次结构图，模块功能
分类号 TP3.162

0 引言

在开发软件系统过程中，特别是为了更新、维护或修改某系统，首先需要了解系统中每个函数模块的功能及系统层次结构图，以此来掌握整个系统的功能及函数间调用关系，因此非常需要有一个方便、实用的工具，根据用户的要求，把一个系统中全部或部分函数间调用关系及其功能自动输出，使系统开发者能够对系统一目了然，从而便于理解、查找或更新修改系统．

本文提出了解决这一问题的方法．根据此方法，在微机DOS系统上设计并实现了C函数调用层次结构图及其各函数模块功能的自动生成，并解决了函数间递归调用的问题[1]．

1 系统功能

C软件有其独特的一面．它可以由多个源文件连接而成．在每个源文件中又可以定义一些函数．每个函数又可以调用其它函数或者函数本身[2]．另外，加上C语言中函数调用的灵活性，使得对C软件的理解变得十分困难．针对C函数调用关系的复杂性，本系统采用对C语言程序进行静态分析，处理其中的函数定义、函数功能及其调用关系．产生函数之间的调用层次结构图并同时输出各函数的功能．

2 系统结构的分析

本系统采用自顶向下，功能模块化的结构程序设计思想，其系统结构如图1所示．全系统总体可以分为3个大的功能模块(如图1中3个虚线框所示)．其3个部分的功能为：

收稿日期：1994－09－29．

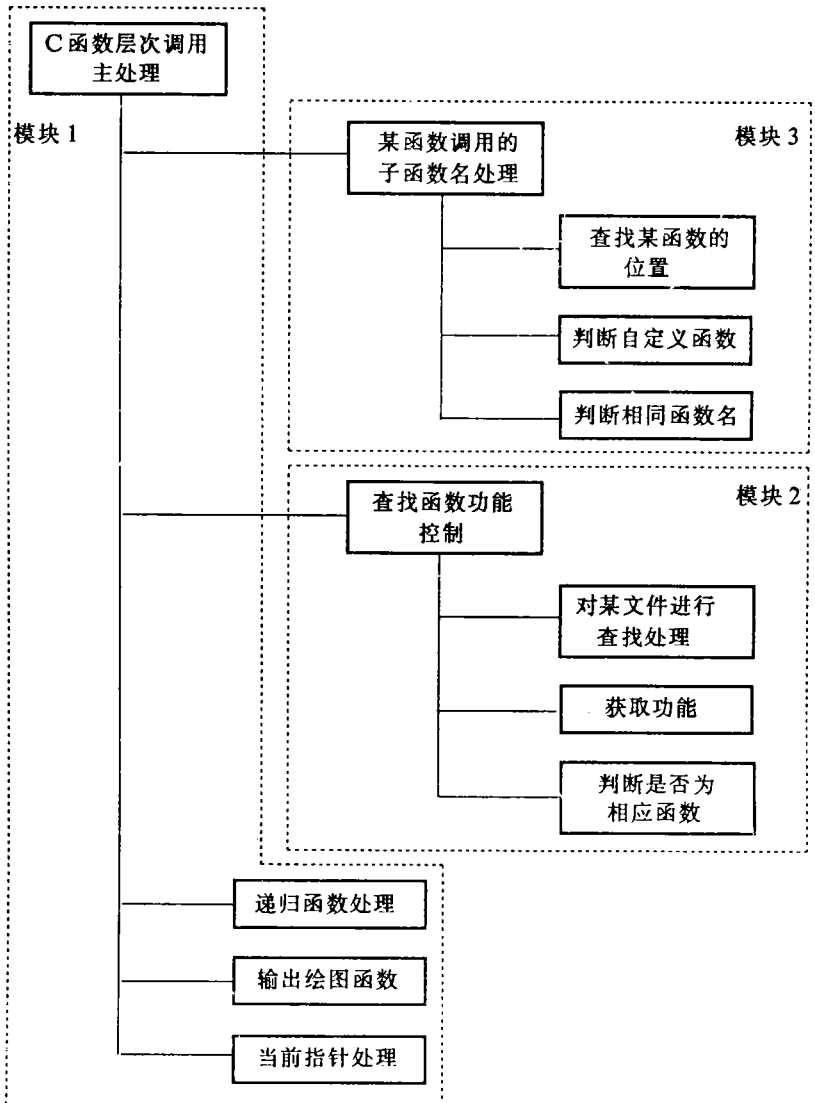


图 1　系统结构

模块 1: 全系统处理过程的控制及绘图输出；模块 2: 查找函数位置及函数功能的处理；模块 3: 函数实体处理.

下面对各模块分别加以介绍:

模块 1 是全系统的控制模块. 作用是控制全系统的工作及绘图输出. 模块 1 通过不断地调用模块 2 和模块 3 进行工作. 为便于函数层次间的移动, 采用了下述的数据结构:

```
CHAIN   {
            Char   function [FUNCMAX];
            Char   name [NAMEMAX];
            CHAIN  *pri, *back, *next;
```

};

其中，pri是父指针，back是子指针，next是兄弟指针.

例如，某系统的层次结构图如图2所示，此时，如果当前指针为F1，则链表的连接状况如图3所示.

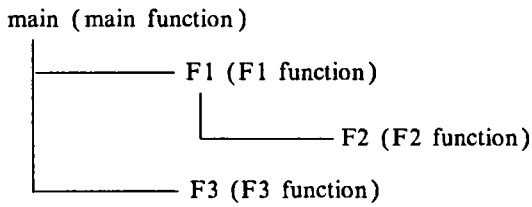


图2 系统层次结构图示例

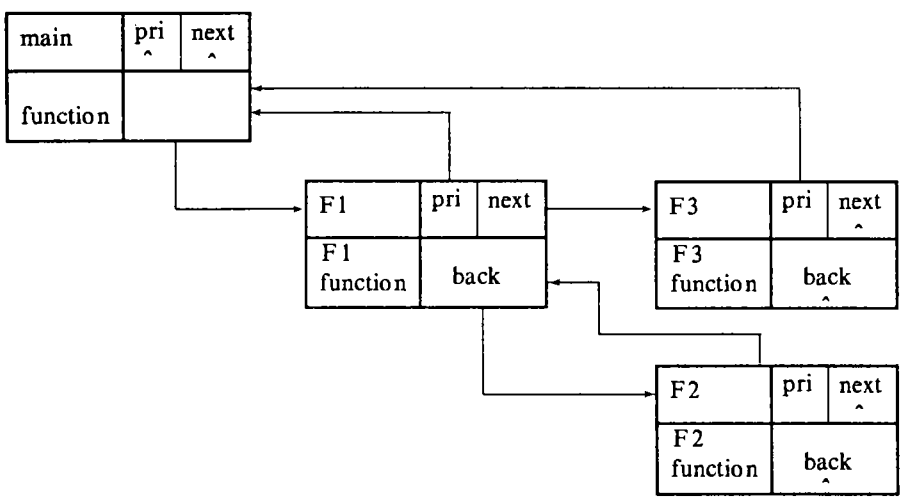


图3 链表连接示意图

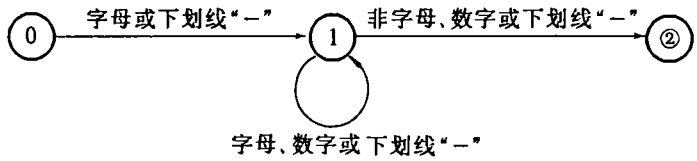
这种结构为回朔查找父函数、平行查找兄弟函数以及判断函数的递归调用(判断子函数名与其所有的父函数名是否相同)提供了方便. 因此，在处理过程中采用了这种动态链表结构[3]. 由于系统在运行过程中链表是动态生成的，因此对于不同大小的系统则会占用相应不同大小的内存. 这种方法不仅可以节省内存，而且可以适用于处理任意规模的系统.

模块2的功能是查找某一函数的位置及函数的功能. 由于开发好的系统一般是由多个源文件组成的，所以为了找到系统中某一函数的位置及功能，需要查询几个或所有的源文件. 为了使系统的应用具有灵活性、便利性，并且使函数的查找具有准确性及快速性，本系统允许用户把构成系统的源文件全路径名称存入到特定的文件中. 这样，不论系统有多大，分布于多少个目录下，都可以正确地、快速地、自动地获得系统的函数调用层次结构图.

为了正确得到各个函数的功能，在编程时，应对函数的功能有一个较规范的说明. 例如，在函数定义前，应对函数功能做如下形式的说明:

/* function: 函数功能　*/其中，function 可以大写也可以小写．

模块 3 的功能是查找某一函数又调用了哪些函数(主要对自定义函数进行处理)．这样就涉及到一些词法分析问题，在处理中通过以下形式的状态转换图得到文件中的标识符．



如果此标识符在"("之前出现，还要检查是否为用户自定义的标识符．如果是自定义的，则得到相应的函数名．此时还需要判断此函数名与父函数调用的已找到的其它子函数名是否重复，以避免函数的重复处理．按上述的处理方法对函数实体进行处理后，则可以得到相应函数中调用的全部子函数．

3 系统实现过程

本系统是在微机 DOS 系统上实现的．用户可以要求生成某系统主函数(main( ))下面(即整个系统)的函数调用层次结构图，也可以要求只生成某个函数(例如：函数 F1( ))所调用的部分函数调用层次结构图．

下面以图 2 的层次结构为例，简述本系统的工作过程．

首先，模块 1 把 main 作为根结点，调用模块 2 查找到 main 的功能．随后调用模块 3 处理 main 函数内部，由返回结构得到 main 有两个子函数 F1 和 F3．系统将 F1 和 F3 链入链表．首先处理 F1，调用模块 2 得到 F1 的功能，再调用模块 3 得知 F1 调用了子函数 F2．此时指针下移，处理 F2，得到 F2 的功能后，再调用模块 3 发现 F2 无子函数，兄弟函数也为空．此时指针回朔到 F1，F1 的兄弟为 F3，对 F3 进行处理后，发现 F3 既无子函数也无兄弟函数，则此时指针回朔到 main．标志看整个系统的生成工作结束．

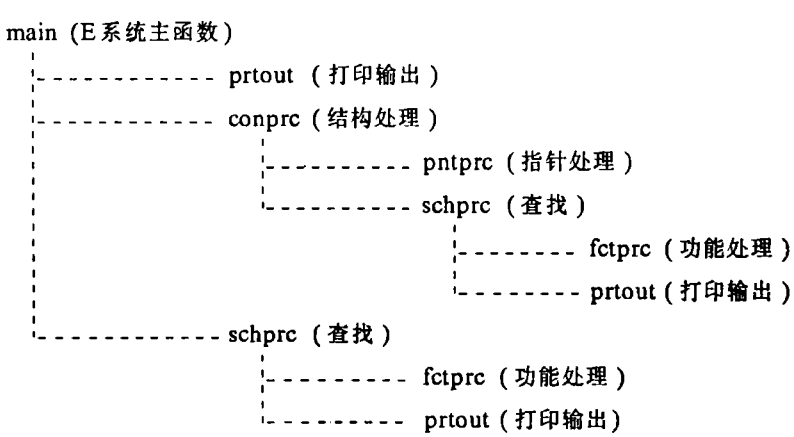


图 4　系统函数层次结构图例

模块1通过计算函数调用的深度来控制打印格式，可得到如图2所示的输出结果(各括号中的内容为相应各函数的功能).

例如，如果要生成某个已开发好的系统的函数层次结构图，该系统的各个程序分别存储在子目录/usr/dos/dvp1、/usr/dos/dvp2及/usr/dos/dvp3的下面，当运行本系统后，则用户所需要的系统函数层次结构图便会自动生成(系统函数层次结构图如图4所示).

4 结束语

本系统在微机DOS系统中得以实现．由于本系统可以象DOS命令一样执行，操作简便，且无特殊要求，所以应用范围较广，可以为广大程序设计者和软件维护提供一个方便的工具．此系统不仅可以自动产生各系统的函数调用层次结构图，而且同时也能够自动生成各函数的功能，使得整个系统的结构及函数功能一目了然，增强了系统开发者对整个系统功能的理解掌握．较好地解决了在较大系统的更新、调试及维护过程中，不易快速准确地查找到系统中各个函数之间的调用关系及各函数的功能的问题．

本系统在设计时采用了功能模块化的设计方法，为系统的扩展做了有利的准备．相信本系统将会用户提供更多的方便．

参 考 文 献

1 Yi Xiaolin. The automatic producing method of AML Modules Flow-Diagram in GIS Developments. Proceedings of the Thirteenth Annual ESRI User Conference. 1993, 2: 571～576

2 王泰峰．微机C语言基础与应用．北京：中国铁道出版社，1991. 1～151, 296～316

3 严蔚敏，吴伟民．数据结构．北京：清华大学出版社，1991. 50～130

The Automatic Generation of C Function Hierarchical Structure Diagrams in the DOS Environment

Yi Xiaolin Wu Yujing

(Computer Institute, Beijing Polytechnic University, 100044)

Abstract An automatic generation method for hierarchical structure diagrams and their module functions for C language function calls on DOS microcomputers is discussed. The problem of retrieving C function calls and their module functions speedily and accurately has been resolved in renewing, testing or maintaining the larger systems. This method also provides a functional tool for systems development software programmers.

Keywords DOS environment, hierarchical structure diagram, module function