

# 面向 C 语言的软件结构引导型 编码与调试系统(C-DSSCTS)\*

柴玉梅

王黎明

(郑州工学院计算机与自动化系) (郑州大学计算机科学系)

**摘 要:** 为了解决用户在编码和调试过程中的一些不便, 本文着重介绍 C-DSSCTS 的思想、功能和一些实现技术特点。

**关键词:** 软件结构、编码、调试

**中图分类号:** TP31

提高软件生产率, 不仅以前是计算机软件发展的重要目标之一, 而且至今依然如此。目前, 在这方面的工作主要分两大类: 一类是软件开发方法的研究。在软件开发过程中, 软件开发方法是主要因素, 但是仅仅靠方法还不能设计出好的软件。因为软件开发是一个复杂的活动, 必须有一定的方便合理的描述工具和辅助工具来支持, 这样才可以生产出高质量的软件。所以, 另一类是软件开发环境的研究与开发, 而语言处理系统是软件开发环境中不可缺少的工具系统, 方便与否直接影响到软件开发的效率, 事实上, 这类系统现已推出许多, 然而都往往没有很好地考虑到当用户在编码或调试一个大型软件的过程中首先要花费一定的精力来管理自己的源代码, 有的由于自己管理不妥会带来许多麻烦。而本文介绍的系统将用户所开发的系统的结构始终呈现在用户的面前, 用户可以对系统结构的各个节点进行操作, 给用户一个直观的感觉。

## 1 系统的功能和结构

本系统是一个集于多工具为一体的工具系统, 它主要由以下几部分组成: 编辑、C 编译、连接、调试、程序库管理, 软件结构生成器、求助、编辑功能函数库和 C 函数库。其中软件结构生成器是本系统的一个主要组成部分, 它接收用户的模块关系描述, 生成一个软件结构投影时, 这是本系统的基础性功能, 以后用户的各种工作都要在软件结构投影树上进行, 这样用户的软件结构从编码到调试都呈现在用户的面前, 它可以提高用户管理代码和操作的工作效率。

---

\* 收稿日期: 1992-12-03

系统的结构如下:

## 2 软件结构投影树的产生

软件结构投影树是由软件结构生成器产生的。但用户需要做初步设计阶段的工作,并能对各模块之间的关系进行分析,最后用模块关系描述语言将分析的结果描述出来,由软件结构生成器接收模块关系描述程序自动产生软件结构投影树。

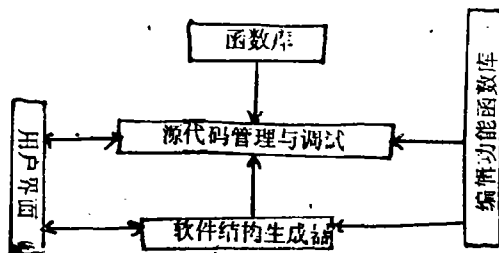


图 1 系统结构

模块关系描述程序的 BNF 公式如下:

<程序>:: = MODEL DZSCRIDTION:

<描述体>

END

<描述体>:: = <MODEL 关系描述语句>

{, <MODEL 关系描述语句>}

<MODEL 关系描述语句>:: =

'<'<MODEL 句名>'Super'<MODEL 名>'Sub'<MODEL 名表>'>

<MOSEL 名表>:: = <MODEL 名>{, <MODEL 名>}空

<MODEL 名>:: = <标识符>

<标识符>:: = 略

Super 后面的模块表示本模块的父模块, Sub 后面的模块是本模块的下属模块。

软件结构生成器的工作流程如下:

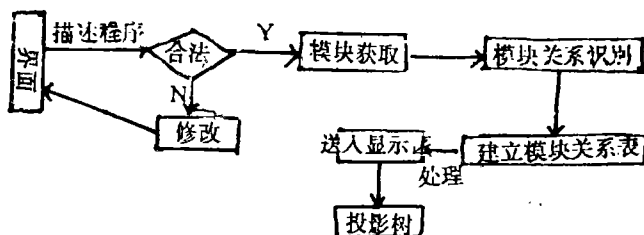


图 2

例如: 有如下图的软件结构实体及其投影树

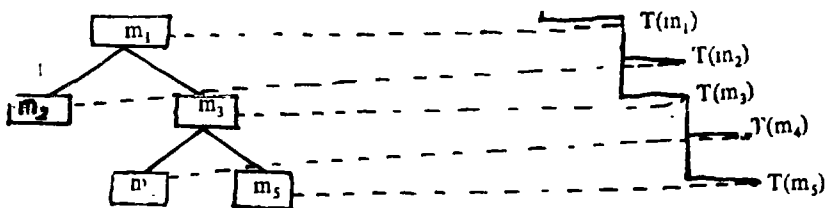


图 3

上图例的模块关系表如下:

$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$m_2$		$m_4$		
$m_3$		$m_5$		

### 3 投影树上的管理和操作

模块之间构成一个软件结构投影树, 对其的管理操作采用公开的方式, 就是让用户在树上直接对源代码进行编辑、查阅、打印、编译、连接、调试等操作, 这种公开方式是样实现的: 系统将模块实体及相互关系投影到屏幕上, 在屏幕上可以显示投影树的全貌, 而用户实际上是在投影树上进行操作, 具体说是支持用户控制光标在投影树的节点上任意移动, 用户可以对光标所在的节点或被标记的节点进行操作, 系统将用户的操作再映射到模块实体上, 从而达到用户操作的目的。

软件的编码与调试过程也是对投影树管理和操作的过程, 其工作屏幕形式如下:

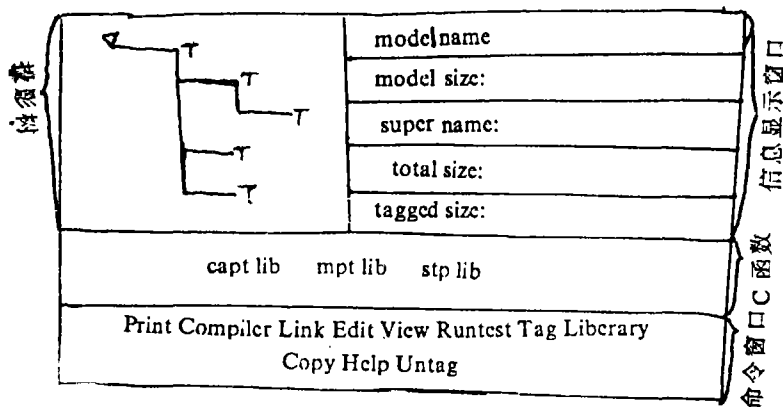


图 4

概括地说, 用户在编码和调试阶段的工作都是在软件结构引导下进行的, 其工作图示如下:

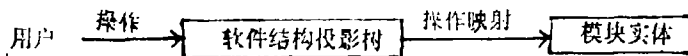


图 5

### 4 调试机制

系统的调试机制不仅可以执行程序而且可以显示程序代码, 它的工作采用对源代码步跟踪的方式, 用户可以控制源代码的执行, 并能显示和修改当前数据流。

由于调试源代码是在运行中进行的, 而且要追踪源代码中数据流的流动, 所以必须将源代码与可执行代码构成一种联系, 当执行被调试程序的某段可执行代码时, 系统才可以根据这种联系将与这段可执行代码对应的源代码显示出来, 为此, 系统在调试之前采用了特殊的编译技术, 需要对源代码做特殊的编译和连接。

特殊编译要求编译程序在做调试编译时满足以下两点:

①对源代码进行编译的同时将行号和一些符号信息加到目标文件中,符号信息包括适当的源代码和需要查找文件的路径,这样源代码和相应的可执行代码就可以通过符号信息联系起来。连接是将这些带有符号信息的目标代码连接起来生成可执行文件。

②对源代码进行编译时不作任何优化,因为优化会使代码结构发生变化,从而破坏源代码与可执行代码之间的对应关系。

这样,对编译程序作了上述改动和限制之后,调试工具就可以接收带有符号信息的可执行文件,进入对源代码的调试阶段。

调试工具的实现是分如下三部分进行的:

①调试界面:交互方式采用菜单驱动和命令驱动。

②代码的执行:代码执行工作实际由两部分组成,其一是对可执行代码的执行,这个工作如何进行对用户来说是不可见的,而用户看到的是被调试的源代码,所以,要把对可执行代码的执行体现在源代码上,数据流的流动也要能通过调试控制操作在源代码中体现出来。因此,其二是可执行代码对应源代码的显示。

③操作

操作按功能主要分两类:一类是调试控制操作,这类操作控制源代码的执行和显示;另一类是对源代码数据流的修改和查看操作,这类操作是在第一类操作的控制下,对程序中的数据流进行追踪,可以对执行当前行以前的某些变量和表达式进行求值显示和修改,这样用户就可以查出程序中数据流的流动是否正常,从而查出错误原因。

系统设置调试工具的目的在于提高用户调试程序的效率,加快软件开发进度,保证软件的质量。

## 5 结论

作者长期从事软件开发工作,发现许多语言系统界面虽然很美观,但并不直观,不能说很方便,为此,才开发了C-DSSCTS,目的在于提高软件的编码和调试效率,尤其是提高源代码管理的效率。本系统已实现完毕,可以达到预期目的。

## 参 考 文 献

- (1) 王黎明. 基于面向对象的软件结构设计及其辅助工具. 计算机科学. 1990. 4
- (2) MICROSOFT软件资料,中国科学院软件所开发公司翻译.

## C-Language-Oriented code and debug system directed by software structure

Chai Yumei

Wang Liming

(Zhengzhou Institute of Technology)

(Zhengzhou University)

**Abstract:** This paper introduce mainly thinker function and technique characteristic of C-DSSCTS for Convenience in the course of code and debug.

**Keywords:** software structure, code, debug