

C 程序设计

考研指导

计算机专业 **考研** 系列教材

李春葆 编著



清华大学出版社

计算机专业考研系列教材

C 程序设计考研指导

李春葆 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

C 语言是一种基础性程序设计语言, 目前已成为各大专院校计算机专业的核心课程, 也是很多高校招收计算机专业硕士研究生的必考科目之一。

本书以通俗易懂的语言全面详实地介绍 C 语言的核心考点及相关知识, 还从 C 程序实现的角度精心设计出一系列富有代表性的例题和习题, 其中包含了大量的研究生入学考试试题, 突出解题思路, 传授解题方法, 相信能帮助读者深刻地领会和掌握 C 语言的编程思路。全书共分为 9 章, 第 1~2 章介绍了 C 语言的基础概念; 第 3~9 章以专题的方式介绍了 C 语言的各组成部分。每章的开头给出了该章的核心考点和考试频度, 其中 5 个“★”号表示考试频度最大。

本书主要针对计算机及相关专业研究生入学考试, 也适合于作为大专院校各专业 C 语言程序设计课程的教材, 还可供计算机软件水平考试者研习。

版权所有, 盗版必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目 (CIP) 数据

C 程序设计考研指导 / 李春葆编著. —北京:

清华大学出版社, 2002

(计算机专业考研系列教材)

ISBN 7-302-06025-8

I.C… II. 李… III. C 语言-程序设计-研究生-入学考试

-自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 083377 号

出版者: 清华大学出版社 (北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市耀华印刷有限公司

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 16.875 字数: 410 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

书 号: ISBN 7-302-06025-8/TP·3592

印 数: 0001~6000

定 价: 24.00 元

《计算机专业考研系列教材》丛书序

计算机专业是当今最热门也是发展最迅速的学科之一，很多学生为了进一步提高专业水平和应用能力，纷纷报考计算机专业研究生。据统计，近几年报考计算机软件与理论、计算机应用和计算机与通信专业硕士研究生的考生远远超过报考其他专业的考生，其中有相当一部分考生原来所学并非计算机专业，还有很多考生是工作多年的在职人员。为了方便报考者复习计算机专业课程，我们特地组织一批计算机专业教学第一线的教授和副教授（其中大多数编写者多年参与硕士研究生入学专业试卷命题工作）编写了本套丛书。丛书包括：

1. 《C程序设计考研指导》
2. 《数据结构考研指导》
3. 《操作系统考研指导》
4. 《编译原理考研指导》
5. 《计算机组成原理考研指导》
6. 《离散数学考研指导》

本丛书具有以下特点：

► 讲述全面而详实

本丛书全面涵盖各门专业课程的内容。不针对个别学校的命题特点，而是充分地讲授课程中的重点、难点和考点，并通过例题进行扩充与深化，使读者得以全面温习与提高，不留“死角”。

► 阐述简洁明了

不同于本、专科教材，本丛书旨在使考生花较少的时间温习各门课程的内容，因此，不过多地解释简单的术语，尽可能地高度概括和总结基本概念，使读者将主要精力集中在解题过程中。

► 重点突出解题思路

本丛书重点介绍解题的方式方法，不仅授人以“鱼”，更授人以“渔”。书中所选的例题和习题大多是计算机专业研究生入学考试试题（题目前标有“▲”号），并配上详解，具有很强的实战性。

► 强调内容的综合与提高

一般的教科书，按照内容的先后顺序按部就班地介绍。这种方式有助于初学者学习，但不便于综合复习，而考研题一般都具有很强的综合性，往往一道题涉及一本书中的好几个概念。本丛书打破了普通教材这种局限性，将相关的概念有机地融于一体，从而提高考生的解题能力。

► 答疑解惑

本丛书所选择的例题和习题大部分都具有较高的难度，书中不仅给出了答案，而且详细介绍解题思路和解题过程，有助于澄清概念和纠正误区。

尽管目前已有一些考研类参考书，但专门针对考研的系列教材还很少，本丛书希望在这方面作一些探索和尝试，起到抛砖引玉的作用。敬请广大读者和同行不吝赐教。

丛书编委会
2002年10月

前 言

C语言是一种基础性程序设计语言，它不仅具有表达能力强、代码质量高和可移植性好等特点，还极好地支持结构化、模块化等软件工程开发方法，更兼备高级语言和低级语言的许多优点。目前，C语言已成为各大专院校计算机专业的核心课程，也是很多高校招收计算机专业硕士研究生考试的科目之一。

全书共分为9章，第1章是绪论；第2章介绍了数据类型及其运算，是C语言程序设计的基础；第3章是选择语句和循环语句，介绍C语言中的两种基本控制结构，其中，重点讨论了穷举法编程的基本思想和方法；第4章是数组，重点讨论了基本的数组排序和查找算法；第5章是指针，也是C语言中最难理解的内容之一，讨论了指针和数组、字符指针和字符串、指针数组、多级指针和数组指针等重要概念，特别是C语言的地址计算方法、第6章是函数，函数最能体现出结构化设计思想，其中重点介绍了函数间传递数据的方法、递归函数的设计方法等；第7章是结构体和共用体，介绍了C语言提供的几种构造数据类型，特别是讨论了使用结构体类型实现链表和树型结构的过程；第8章是编译预处理；第9章是文件，介绍了文件的类型、输入输出函数和随机读写的方法。每章的开头给出了该章的核心考点和考试频度，其中5个“★”号表示考试频度最大。

本书除了介绍C语言的基本内容外，各章还给出“本章基础要点”，列出了本章中包含的一些最重要的概念，这些概念往往在考题中以选择题或填空题的形式出现。另外，书中还精选了大量的研究生入学考试试题（前面加有“▲”号）。

本书的特点是概念清晰，文字描述简洁明了。内容与习题部分各占一半，所有习题都给出详细的解答。

尽管本书主要是针对考研者应试复习和提高，但同样也适合于作为大专院校各专业C语言程序设计课程的教学参考书，还可供计算机软件水平考试者研习。

由于作者水平有限，书中难免存在缺点和不足之处，敬请广大读者不吝指正。

作 者
2002年9月

目 录

第1章 绪论	1
1.1 计算机语言	1
1.2 C语言	3
1.3 C程序示例	3
1.4 C程序的编译过程	5
1.5 本章基础要点	8
1.6 习题及参考答案	8
第2章 数据类型及其运算	10
2.1 标识符	10
2.2 C语言的数据类型	11
2.3 常量	12
2.3.1 整型常量	12
2.3.2 实型常量	13
2.3.3 字符常量	14
2.3.4 字符串常量	15
2.3.5 符号常量	15
2.4 变量	16
2.4.1 变量的说明	16
2.4.2 变量的类型	17
2.5 运算符与表达式	19
2.5.1 算术运算符和算术表达式	20
2.5.2 递增、递减运算符	22
2.5.3 赋值运算符和赋值表达式	23
2.5.4 位运算符	24
2.5.5 其他运算符	25
2.6 简单输入与输出	25
2.6.1 数据输出	25
2.6.2 数据输入	28
2.7 本章基础要点	30
2.8 习题及参考答案	31
第3章 选择语句和循环语句	37
3.1 语句的概念	37

3.2 选择语句	38
3.2.1 if 语句	38
3.2.2 switch(开关)语句	41
3.3 循环语句	45
3.3.1 while循环语句	45
3.3.2 do-while循环语句	47
3.3.3 for循环语句	48
3.3.4 break语句	50
3.3.5 continue语句	51
3.4 穷举法编程	52
3.5 本章基础要点	55
3.6 习题及参考答案	56
第4章 数组	66
4.1 数组说明和引用	66
4.1.1 一维数组	66
4.1.2 二维数组	68
4.2 字符数组和字符串数组	72
4.2.1 字符数组	72
4.2.2 字符串数组	75
4.3 数组的排序	77
4.3.1 冒泡排序	77
4.3.2 直接插入排序	78
4.3.3 直接选择排序	79
4.3.4 快速排序	80
4.4 数组的查找	81
4.4.1 顺序查找	81
4.4.2 二分查找	81
4.5 本章基础要点	82
4.6 习题及参考答案	83
第5章 指针	94
5.1 指针与指针变量	94
5.2 指针变量的定义及运算	96
5.2.1 指针的说明	96
5.2.2 指针运算符	96
5.2.3 指针的初始化	96
5.2.4 指针运算	98
5.3 指针和数组	100
5.3.1 指针与一维数组	100

5.3.2 指针与二维数组	102
5.4 字符指针和字符串	103
5.5 指针数组	105
5.6 多级指针	107
5.7 数组指针	109
5.8 本章基础要点	110
5.9 习题及参考答案	110
第6章 函数	125
6.1 函数的定义和使用	125
6.1.1 函数定义	125
6.1.2 函数调用	126
6.1.3 被调函数说明	127
6.2 函数的存储类型	128
6.2.1 extern型函数	128
6.2.2 static型函数	128
6.3 变量的作用域和存储类型	129
6.3.1 局部变量和全局变量	129
6.3.2 变量的存储类型	130
6.4 函数的数据传递	132
6.4.1 数据复制方式	133
6.4.2 地址传递方式	133
6.4.3 return传递数据	135
6.4.4 全局变量传递数据	135
6.5 在函数间传递数组	136
6.5.1 一维数组作为函数参数	136
6.5.2 二维数组作为函数参数	137
6.6 指针型函数	139
6.7 指向函数的指针	140
6.8 递归函数及其设计方法	142
6.8.1 递归模型	142
6.8.2 递归的执行过程	142
6.8.3 递归设计	143
6.9 命令行参数	144
6.10 本章基础要点	145
6.11 习题及参考答案	146
第7章 结构体与共用体	160
7.1 结构体说明和变量定义	160
7.1.1 结构体说明	160

7.1.2 结构体变量的定义	161
7.1.3 结构体变量的引用和初始化	162
7.2 结构体数组	164
7.2.1 结构体数组的定义	164
7.2.2 结构体数组的引用	165
7.2.3 结构体数组的初始化	165
7.3 结构体指针	167
7.3.1 结构体变量指针定义	167
7.3.2 结构体数组指针	168
7.4 函数间结构体变量的数据传递	169
7.4.1 结构体数据复制方式	169
7.4.2 结构体地址传递方式	170
7.4.3 在函数间传递结构体数组	171
7.5 结构体的应用：链表	173
7.5.1 C语言动态分配函数	173
7.5.2 单链表及其基本运算的实现	173
7.5.3 单链表的应用例子	177
7.6 结构体的应用：二叉树	180
7.6.1 二叉树的定义与存储结构	180
7.6.2 二叉树的遍历	180
7.6.3 二叉树的应用例子	183
7.7 共用体	184
7.7.1 共用体类型的定义	184
7.7.2 共用体变量的定义	185
7.7.3 共用体变量的引用	185
7.8 枚举类型	186
7.9 用户自定义类型	187
7.10 本章基础要点	188
7.11 习题及参考答案	189
第8章 编译预处理	202
8.1 宏	202
8.1.1 无参宏	202
8.1.2 带参宏	202
8.2 条件编译	204
8.3 文件包含	207
8.4 本章基础要点	208
8.5 习题及参考答案	208

第9章 文件	211
9.1 文件概述	211
9.1.1 文件的分类	211
9.1.2 设备文件	212
9.1.3 流和文件指针	212
9.1.4 文件缓冲区	213
9.2 文件的输入输出	213
9.2.1 文件的打开和关闭	213
9.2.2 文件的字符输入输出函数	215
9.2.3 文件的字符串输入输出函数	217
9.2.4 文件的格式化输入输出函数	219
9.2.5 文件的数据块输入输出函数	220
9.3 文件的定位操作	222
9.4 文件综合操作	225
9.5 本章基础要点	229
9.6 习题及参考答案	230
附录A 一份《C程序设计》全真考研试题及参考答案	250
附录B C语言运算符及优先级	254
附录C 部分字符与ASCII代码对照表	256
参考文献	257

第1章 绪论

核心考点：C语言的特点，C程序的组成和执行过程。

考试频度：★

1.1 计算机语言

计算机语言是人们描述计算过程(即程序)的规范书写语言。程序是对计算机处理对象和计算规则的描述。语言的基础是一组记号和规则，根据规则由记号构成记号串的总体就是语言。

我们知道，人类自然语言如汉语是人们交流和表达思想的工具。那么，人与计算机如何“交流”呢？为此，就产生了计算机语言，其功能是人利用计算机语言编写一系列的指令，计算机能够“理解”这些指令，按照该指令去执行。正是存在这种相同点，才把计算机语言和自然语言都称为“语言”。

当然，自然语言具有历史性、文化性，除了语法外，还包含复杂的语义和语境，所以，很多不完全符合语法的语句，人们也能理解。但计算机语言是人发明的，它主要是用语法来表达人(即程序员)的思想，所以编写程序时必须严格遵守语法规则。

下面，介绍一些程序语言的发展简史。这不单是讲历史，也是介绍软件技术的一些内在规律。

计算机硬件向使用者直接提供的使用手段主要是指令系统，也就是说，指令是计算机硬件能直接理解的语言，因此把指令系统称为机器语言。

在20世纪50年代，人们用这种机器语言编写程序指令序列，完成所需的计算工作。那时，人们首先将计算工作细致地分解成许多步骤，对其中的每一步，在指令系统中选用相应的指令来实现。由于指令都是用代码“0”和“1”组成的，所以，用机器语言编写的程序可读性很差，更难以修改和扩充。在编写程序时，程序员还常常需要人为地指定该程序的存储地址，并分配数据的存储单元。总之，最早期的程序员兼操作员直接面向的是0、1这种比特信息流。



由于各种计算机的指令系统差别很大,因此所编写的程序也只能专用于特定类型的计算机,不能移植。另外,这种语言的编程效率极低,还要求较强的专业性,因此,当时的计算机使用者只能是极少数计算机专业技术人员。

虽然现在已没有人用机器语言编写程序了,但是直到现在,所有直接能在计算机硬件上运行的程序(目标程序)仍然是机器指令代码序列,简称机器代码。这是计算机最基本的原理之一。因此,虽然把机器语言作为最早期的第一代程序语言,但今天的用户还是需要了解机器代码(如*.exe文件)的有关概念和特征(可运行,难以阅读,难以直接修改,一般来说,每个模块必须连续存储)。

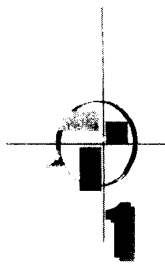
正是由于人机语言之间的巨大差异,协调两者之间的关系就成为程序语言发展的动力。后来,为了提高程序的易读性,人们用助记符来表示指令中的操作码和操作数的地址码。例如,用ADD表示操作“加”,用变量名A表示该变量所存放的存储单元地址。这样,对于专业程序员读懂并且编写程序确实提供了很大帮助,同时,也使程序中的很多语句能独立于具体的机型,将程序移植到别的机器上所需的工作量也有所减少。这种符号形式的指令系统被称为汇编语言,其中的汇编语句基本上与指令一一对应。用汇编语言编写的程序需要通过专门的翻译程序“汇编程序”将其翻译成机器语言,才能在机器上执行。因此,在人机之间,汇编语言比起机器语言来,算是向人的方面靠拢了一步。为了提高使用汇编语言的效率,人们又设计了用一条汇编指令来描述若干条指令的宏指令。具有宏指令的汇编语言称为宏汇编。

汇编语言属于第二代程序语言,它在总体上仍然依赖于具体的机型,仍然难以阅读,难以移植。正是由于汇编语言仍然没有完全摆脱对具体机型的依赖,在人机之间仍然比较靠近机器方,所以在人机之间划分的层次结构中,它和机器语言一样都是位于低层(人处于最高层)的。因此,汇编语言和机器语言都称为低级程序语言,都是面向机器的语言。

为了进一步改进程序语言,使程序员能够像书写算术表达式那样编写程序,从20世纪60年代初以后,陆续推出了多种程序语言,如BASIC、Fortran、C和Pascal语言等,称为第三代程序语言。对一般的科技工作者来说,这些语言容易读懂,也不难编写。更重要的是这些语言基本上不依赖于具体的机型,易于在各种计算机上移植使用。在人机之间,这些语言更靠拢人,因此,第三代程序语言也称为高级程序语言。

计算机程序语言的发展总趋势是越来越向人所用的自然语言靠拢,越来越多地采用自动编程技术,越来越盛行软件部件重用。特别是近些年来,随着图形用户界面(GUI)的普及,面向对象程序设计方法的推广以及可视化软件开发工具的兴起,软件开发者的编程工作量大为降低,软件开发的效率以及软件的质量都大为提高。

但是,我们也要清醒地认识到,各种软件开发工具以及新的软件开发技术都是建立在高级程序语言的基础上的。所有的自动化编程技术都不可能完全取消编程。作为软件技术人员,单单了解软件的自动生成操作方法是远远不够的,更需要了解其中的机理,以利于适应复杂多变的应用需求,更好地维护和完善应用系统。为此,许多高校都把常用的C语言作为硕士研究生入学考试科目。



1.2 C语言

C语言是一种高级程序设计语言，它是由美国AT&T公司贝尔实验室的D. 里奇在1972年开发出来的。由于用C语言编写的程序简洁、容易阅读和修改，在计算机上执行效率高，并且在不同类型的计算机上都可运行，所以C语言成为高级语言中的佼佼者。目前，它应用范围最广，发展最快，是国际上公认的优秀的设计语言。

C语言之所以如此普及并广受称赞，是与它的一系列显著特点分不开的。C语言的主要特点有：

- **简洁** 用C语言编写的程序往往短小精悍，代码的行数不多，功能却很强，使用起来也很灵活，易于学习和应用。另外，C编译程序的代码量较小，便于在微型机上应用。
- **功能强** C语言是一种通用的语言，在不同机型上都可运行，适用范围很广，可用在科学计算、商业处理、软件开发等方面。它具有某些高级语言所缺乏的功能(如二进制位的操作)，可以充分反映当前计算机的特性。
- **效率高** 许多试验表明，针对同一问题，用C语言编写的程序，执行速度比用其他高级语言(如Fortran、BASIC)编写的程序快许多倍。
- **可移植性好** 可移植是指编写的程序不需要做很多改动就可从一种机型上移到另一种机型上运行。C语言在这方面确有专长，从而扩大了应用范围，增强了生存力。

另外，C语言完全支持结构化程序设计方法。将一个大问题分解为若干小“模块”，像搭积木那样。掌握了这种方法，对以后进一步提高程序设计能力，以及实际开发软件，都是很有益处的。

C语言有很多优点，但也和其他语言一样，存在某些缺点。例如，运算符较多，某些运算符优先顺序与习惯不完全一致，类型转换比较随便等。然而与它的优点相比，这些问题还是很次要的。当然，在学习和使用C语言时要注意这些问题。

1.3 C程序示例

用C语言编写的程序称为C程序，其主要成分是函数。在C语言中，函数被描述成一个功能块，能独立地实现某个功能。C程序由一个或多个函数组成，但主函数main()是一个特殊的、必不可少的函数，程序就从这里开始运行。主函数相当于程序的总控，可以在其中再调用其他一系列函数。

例如，如下完整的C程序用于计算n! 的值。

```
/*文件名:mult.c*/  
#include <stdio.h>          /*插入标准I/O库的头文件*/  
int fun(int x)              /*定义函数fun(),形参x为整型,返回整型值*/  
{
```



```
int i, y;                /*说明局部变量i和y为整型*/
y=1;                    /*变量y赋初值1*/
for (i=1;i<=x;i++)      /*通过循环求y值*/
    y=y*i;
return y;               /*返回y的值*/
}
main()                  /*主函数*/
{
    int n, m;           /*说明局部变量n, m为整型*/
    printf("n:");       /*输出"n:"*/
    scanf("%d", &n);    /*输入n值*/
    m=fun(n);           /*调用函数fun()*/
    printf("%d!=%d\n", n, m); /*输出结果*/
}
```

这是一个简单的C程序，其解释如下：

1. C程序中小写字母与大写字母是不同的，语言本身规定的关键字(如main, int等)都是小写字母。
2. 每个语句用分号结尾。“/*”与“*/”之间的部分是注解，只是为了阅读方便，不参与编译执行。
3. 在C语言编译系统中有许多以.h为扩展名的文件(称为头文件)。在这些头文件中，对相应的函数原型及符号常量等进行了说明和定义。例如，一般的C源程序都需要用库函数进行输入输出操作，因此在源程序的最前面一般需要使用语句：

```
#include <stdio.h> /*本程序中的输入输出函数有scanf()、printf()*/
```

表示在此处插入标准输入输出库函数相应的头文件stdio.h。

4. 为了增强程序的可读性，程序中的语句常用缩格(行前插入空格)形式以体现层次结构，程序中的标识符应当含义明确，便于记忆。
5. 本程序由两个函数组成，fun()是一个自定义函数，其功能是返回x!的值；main()是主函数，由它调用fun()函数。

一般地，C程序存储在以.c为扩展名的文件中。例如，上述程序存储在mult.c文件中，在编译成可执行文件mult.exe后，执行该程序，要求输入一个整数，然后输出该整数的阶乘值，如：

```
n:5↵
5!=120
```

其中，“↵”表示回车键，下划线表示是由用户输入的。其他部分是由程序执行后输出的。本书中均采用这种表示方式。

1.4 C程序的编译过程

高级程序语言具有一定的语法规则，人们可以用其中的各种语句来描述算法过程，形成比较容易理解的源程序。将这种源程序输入计算机后，需经“编译程序”或“解释程序”的编译才能变成计算机硬件可直接理解执行的指令代码序列。一般地，C程序是经“编译程序”编译成指令代码序列，从编写一个C程序到把它转换成可执行程序的过程如图1.1所示。

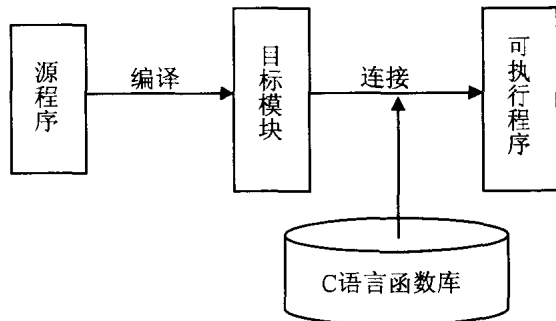


图 1.1 C 源程序的编译过程

编译程序将源程序编译成目标程序后保存在另一个文件中，该目标程序可以脱离编译程序直接在计算机上反复多次运行。大多数软件产品都是以目标程序形式发行给用户的，不仅便于直接运行，同时又使别人难以盗用其中的技术。

编译过程是一个复杂的整体过程。图1.2给出了一个编译过程的各个阶段，这是一种典型的划分方法。事实上，某些阶段可以组合在一起。这些阶段间的源程序的中间表示形式就没必要构造出来了。

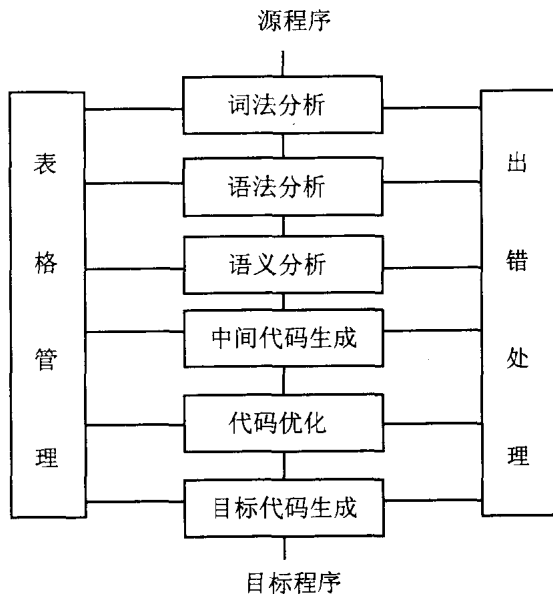


图 1.2 编译过程的各个阶段



1. 词法分析阶段 是编译过程的第一个阶段。这个阶段的任务是从左到右一个字符一个字符地读入源程序，识别出一个个单词符号。单词符号是程序设计语言的基本语法符号，如：基本字(也称关键字或保留字)、标识符、常数、运算符和定界符。词法分析程序所输出的单词符号常采用以下二元式表示：

(单词种别, 单词自身的值)

其中，单词种别是语法分析需要的信息，而单词自身的值则是编译其他阶段需要的信息。

词法分析所依据的是语言的词法规则，即描述单词结构的规则。例如，某源程序片断如下：

```
{ float s, f, g; s=f+g*10; }
```

词法分析阶段将构成这段程序的字符组成如下单词序列：

- | | |
|--------------|-------|
| (1) 复合语句开始符 | { |
| (2) 保留字 | float |
| (3) 标识符 | s |
| (4) 逗号 | , |
| (5) 标识符 | f |
| (6) 逗号 | , |
| (7) 标识符 | g |
| (8) 语句结束符 | ; |
| (9) 标识符 | s |
| (10) 赋值号 | = |
| (11) 标识符 | f |
| (12) 加号 | + |
| (13) 标识符 | g |
| (14) 乘号 | * |
| (15) 整数 | 10 |
| (16) 语句结束符 | ; |
| (17) 复合语句结束符 | } |

经过词法分析后，赋值语句 $s=f+g*10$ 表示为 $id1=id2+id3*10$ ，其中 $id1$ 、 $id2$ 和 $id3$ 分别表示 s 、 f 和 g 三个标识符的内部形式。

2. 语法分析阶段 语法分析的任务是在词法分析的基础上将单词符号序列分解成各类语法单位，如“程序”、“语句”、“表达式”等。语法分析所依据的是语言的语法规则，即描述程序结构的规则。通过语法分析确定整个输入串是否构成一个语法上正确的程序。

上述的 $id1=id2+id3*10$ 经语法分析得知其是C语言的“赋值语句”，可表示成图1.3所示的语法树。