

全国计算机等级考试指定教材配套辅导



National Computer
Rank Examination

全国计算机等级考试 考点解析与模拟训练

(三级 C 语言)

高峥虹 主编

李浩岩 赵峰 余亚杰 副主编

- 
- 全真等级考试模拟环境
 - 历年真题和典型习题题库
 - 评分系统突出考试重点难点
 - 答题解析总结高分策略

清华大学出版社



TP3
430D
:3
2007

National Computer
Rank Examination

全国计算机等级考试 考点解析与模拟训练

(三级 C 语言)

高峥虹 主编

李浩岩 赵峰 余亚杰 副主编

清华大学出版社
北京

内 容 简 介

本书是全国计算机等级考试三级 C 语言的考试辅导教材,结合作者对计算机等级考试的多年研究和丰富的教学经验,紧扣考试大纲,内容取舍得当。

书中针对三级 C 语言上机考试的题库,全面讲解考试中用到的 C 语言知识。例题和课后习题大多来源于三级考试上机题库,把考试的考点和 C 语言知识点紧密的结合在一起,形成一本具有完整知识体系又针对于计算机三级上机考试的 C 语言教材。

本书适合各种计算机等级考试机构做为教材使用,也可做为考生自学使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

全国计算机等级考试考点解析与模拟训练(三级 C 语言)/高峥虹主编. —北京:清华大学出版社,2007.2

ISBN 978-7-302-14582-0

I. 全… II. 高… III. ①电子计算机-水平考试-自学参考资料 ②C语言-程序设计-水平考试-自学参考资料
IV. TP3

中国版本图书馆 CIP 数据核字(2007)第 013322 号

责任编辑:冯志强 刘霞

责任校对:张剑

责任印制:王秀菊

出版发行:清华大学出版社

<http://www.tup.com.cn>

c-service@tup.tsinghua.edu.cn

社总机:010-62770175

投稿咨询:010-62772015

地址:北京清华大学学研大厦 A 座

邮编:100084

邮购热线:010-62786544

客户服务:010-62776969

印刷者:北京牛山世兴印刷厂

装订者:北京市密云县京文制本装订厂

经销:全国新华书店

开本:210×285 印张:11 字数:351 千字

(附光盘 1 张)

版次:2007 年 2 月第 1 版

印次:2007 年 2 月第 1 次印刷

印数:1~5000

定价:23.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。

联系电话:010-62770177 转 3103

产品编号:024261-01



前言 Preface

随着我国计算机应用的进一步普及和深入,人们已经达成了一个共识:计算机知识是当代人类文化的重要组成部分;计算机应用能力是跨世纪人才不可缺少的素质。全国计算机等级考试(National Computer Rank Examination, NCRE)是由教育部考试中心主办,用于考查应试人员的计算机应用知识与能力的考试。本考试的证书已经成为许多单位考核、录用工作人员的重要条件。

全国计算机等级考试从1994年开考以来,三级上机题库一共分为两类,一种是C语言题库,另一种是汇编语言题库。三级考试一共分4个科目,即三级网络技术、三级信息管理、三级数据库和三级PC技术。其中只有三级PC技术上机考试考汇编语言,其他科目均考C语言,而且这三科使用的是同一个题库。

在等级考试火热的今天,市面上针对计算机三级考试的C语言教材很少见。本书较好地体现了计算机等级考试三级的特点,侧重点清晰,有利于考生的复习与备考。本书紧扣考试大纲,内容取舍得当,是一本较好的备考教材。

本书编者依据多年来对计算机等级考试的研究,以丰富的教学经验编写了这本三级C语言教材。本书的特色在于,针对三级C语言上机考试的题库,全面讲解考试中用到的C语言知识。例题和课后习题大多来源于三级考试上机题库。这样就可以把考试的考点和C语言知识点紧密地结合在一起,形成一本具有完整知识体系又针对计算机三级上机考试的C语言教材。希望本书能助读者收到事半功倍的效果,顺利通过考试。

本书适合各类计算机等级考试机构作为教材使用,可作为考生自学使用。本书各章课后的习题答案可以在本书的配套教材《全国计算机等级考试考点解析与上机指导(三级C语言)》中找到。

本书由高峥虹、赵峰、余亚杰、李浩岩、李鑫、阿庆兴等编写,由于编写时间仓促,难免有疏漏之处,在学习过程中,如有意见或建议,请与我们联系:zhichengcomputer@163.com,或登录网站: <http://www.zcre.cn>。

编 著



目录 Contents

第1章 C语言概述	1	2.6.1 变量赋初值	19
1.1 C语言概述	2	2.6.2 变量类型的转换	19
1.1.1 C语言的发展过程	2		
1.1.2 C语言的特点	2	第3章 运算符和表达式	21
1.2 简单的C程序介绍	3	3.1 运算符简介	22
1.3 C语言的基本语法	4	3.1.1 运算符的分类	22
1.3.1 C语言源程序的结构特点	4	3.1.2 运算符的优先级和结合性	22
1.3.2 书写程序时应遵循的规则	4	3.2 算术运算符	23
1.3.3 C语言的字符集	4	3.2.1 算术运算符的分类	23
1.3.4 C语言主要术语	5	3.2.2 算术运算符的优先级和结合性	23
1.4 Turbo C 2.0 集成开发环境的使用	6	3.2.3 算术运算符的应用	23
1.4.1 Turbo C 2.0 简介和启动	6	3.3 赋值运算符和赋值表达式	24
1.4.2 Turbo C 2.0 集成开发环境	6	3.4 类型转换运算符	24
1.4.3 常用菜单介绍	6	3.5 复合的赋值运算符	25
1.4.4 Turbo C 2.0 的配置文件	10	3.6 自增、自减运算符	25
第2章 C语言的数据类型	11	3.7 强制类型转换运算符	26
2.1 数据类型	12	3.8 逗号运算符和逗号表达式	27
2.1.1 基本数据类型	12	3.9 位运算符	27
2.1.2 构造数据类型	12	3.9.1 位运算符的分类	28
2.1.3 指针类型	12	3.9.2 按位与运算	28
2.1.4 空类型	12	3.9.3 按位或运算	28
2.2 常量与变量	12	3.9.4 按位异或运算	29
2.2.1 常量和符号常量	13	3.9.5 求反运算	29
2.2.2 变量	13	3.9.6 左移运算	29
2.3 整型数据	13	3.9.7 右移运算	30
2.3.1 整型常量	14	习题	30
2.3.2 整型变量	14		
2.4 实型数据	16	第4章 C语言程序设计初步	32
2.4.1 实型常量	16	4.1 C语句概述	33
2.4.2 实型变量的分类	16	4.2 赋值语句	34
2.5 字符型数据	17	4.3 C语言的输入输出	35
2.5.1 字符常量	17	4.3.1 putchar 函数(字符输出函数)和 getchar 函数(键盘输入函数)	35
2.5.2 转义字符	17	4.3.2 格式输出函数 printf	36
2.5.3 字符变量	17	4.3.3 格式输入函数 scanf	40
2.5.4 字符串常量	18	4.4 顺序结构程序设计举例	43
2.6 变量赋初值和类型转换	19	习题	44

第 5 章 分支程序设计	45	7.3.4 字符串和字符串结束标志	78
5.1 关系运算符和表达式	46	7.3.5 字符数组的输入输出	78
5.1.1 关系运算符及其优先次序	46	7.3.6 字符串处理函数	80
5.1.2 关系表达式	46	习题	84
5.2 逻辑运算符和表达式	46	第 8 章 函数	86
5.2.1 逻辑运算符及其优先次序	46	8.1 概述	87
5.2.2 逻辑运算的值	47	8.2 函数定义的一般形式	87
5.2.3 逻辑表达式	47	8.3 函数的参数和函数的值	89
5.3 if 语句	48	8.3.1 形式参数和实际参数	89
5.3.1 if 语句的 3 种形式	48	8.3.2 函数的返回值	90
5.4 if 语句的嵌套	53	8.4 函数的调用	91
5.5 条件运算符和条件表达式	54	8.4.1 函数调用的一般形式	91
5.6 switch 语句	55	8.4.2 函数调用的方式	91
习题	58	8.4.3 被调用函数的声明	92
第 6 章 循环控制	58	8.5 函数的嵌套调用	93
6.1 概述	59	8.6 数组作为函数参数	94
6.2 while 语句	59	8.7 局部变量和全局变量	99
6.3 do-while 语句	60	8.7.1 局部变量	99
6.4 for 语句	61	8.7.2 全局变量	100
6.5 循环的嵌套	65	8.8 变量的存储类别	102
6.6 break 和 continue 语句	66	8.8.1 动态存储方式与静态 动态存储方式	102
6.6.1 break 语句	66	8.8.2 auto 变量	102
6.6.2 continue 语句	66	8.8.3 用 static 声明局部变量	103
习题	67	8.8.4 register 变量	103
第 7 章 数组	69	8.8.5 用 extern 声明外部变量	104
7.1 一维数组的定义和引用	70	习题	104
7.1.1 一维数组的定义方式	70	第 9 章 预处理命令	106
7.1.2 一维数组元素的引用	71	9.1 概述	107
7.1.3 一维数组的初始化	72	9.2 宏定义	107
7.1.4 一维数组程序举例	73	9.2.1 无参宏定义	107
7.2 二维数组的定义和引用	74	9.2.2 带参宏定义	107
7.2.1 二维数组的定义	74	9.3 文件包含	108
7.2.2 二维数组元素的引用	74	第 10 章 指针	109
7.2.3 二维数组的初始化	75	10.1 地址指针的基本概念	110
7.2.4 二维数组程序举例	76	10.2 变量的指针和指向变量的指针变量	110
7.3 字符数组	76	10.2.1 定义一个指针变量	111
7.3.1 字符数组的定义	77	10.2.2 指针变量的引用	111
7.3.2 字符数组的初始化	77	10.2.3 指针变量作为函数参数	114
7.3.3 字符数组的引用	77		

10.2.4 指针变量几个问题的进一步说明.....	116	11.7.3 结构指针变量作函数参数.....	142
10.3 数组指针和指向数组的指针变量.....	119	习题.....	145
10.3.1 指向数组元素的指针.....	120	第 12 章 文件	146
10.3.2 通过指针引用数组元素.....	120	12.1 C 文件概述.....	147
10.3.3 数组名作函数参数.....	122	12.2 文件指针.....	147
10.4 字符串的指针和指向字符串的指针变量.....	126	12.3 文件的打开与关闭.....	148
10.4.1 字符串的表示形式.....	126	12.3.1 文件打开函数 fopen.....	148
10.4.2 使用字符串指针变量与字符数组的区别.....	129	12.3.2 文件关闭函数 fclose.....	150
10.5 字符串应用举例.....	130	12.4 文件的读写.....	150
习题.....	131	12.4.1 字符读写函数 fgetc 和 fputc.....	150
第 11 章 结构体与共用体	133	12.4.2 字符串读写函数 fgets 和 fputs.....	152
11.1 定义一个结构的一般形式.....	134	12.4.3 格式化读写函数 fscanf 和 fprintf.....	154
11.2 结构类型变量的说明.....	134	12.5 C 库文件.....	155
11.3 结构变量成员的表示方法.....	136	12.6 文件操作程序举例.....	156
11.4 结构变量的赋值.....	137	附录	158
11.5 结构变量的初始化.....	137	附录 A 常用字符 ASCII 代码对照表.....	159
11.6 结构数组的定义.....	138	附录 B C 语言的运算符和结合性.....	159
11.7 结构指针变量的说明和使用.....	140	附录 C C 语言的关键字.....	160
11.7.1 指向结构变量的指针.....	140	附录 D 常用 C 库函数.....	161
11.7.2 指向结构数组的指针.....	141	附录 E Turbo C 2.0 编译错误信息.....	164

第1章

C 语言概述



1.1 C 语言概述

1.1.1 C 语言的发展过程

C 语言是国际上广泛流行的很有发展前途的计算机高级语言。它适合描述系统语言，主要用来编写系统软件，也可以编写应用软件。

C 语言是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时由 K.Thompson 和 D.M.Ritchit 合著了著名的“THE C PROGRAMMING LANGUAGE”一书，通常简称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有定义一个完整的标准 C 语言，后来由美国国家标准协会（American National Standards Institute）在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C。

C 语言是在 B 语言基础上发展起来的。1971 年剑桥大学的 K.Thompson 在 PDP-11/20 上实现了 B 语言，并编写了 UNIX 操作系统。但 B 语言过于简单，功能有限。1973 年 K.Thompson 和 D.M.Ritchit 两人合作把 UNIX 的 90% 以上用 C 改写，即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国的贝尔实验室开发成功的，使用的是汇编语言。

随着 UNIX 操作系统的日益广泛使用，C 语言也迅速得到推广。C 语言和 UNIX 可以说是一对孪生兄弟，在发展过程中相辅相承。现在 C 语言早已风靡全世界，成为世界上应用最广泛的几种计算机语言之一。

早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们所认识，到了 20 世纪 80 年代，C 开始进入其他操作系统，并很快在各类大、中、小和微型计算机上得到了广泛的使用，成为当代最优秀的程序设计语言之一。

目前最流行的 C 语言有以下几种：

- Microsoft C 或称 MS C；
- Borland Turbo C 或称 Turbo C；
- AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

教育部考试中心规定，全国计算机等级考试三级使用编程软件 Turbo C 2.0。

1.1.2 C 语言的特点

C 语言有以下基本特点。

(1) C 语言简洁、紧凑，使用方便、灵活。

ANSI C 一共只有 32 个关键字（见附录 C）和 9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。

(2) 运算符丰富。C 语言中关键字虽然少了，运算符却很丰富，共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理，从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的运算。关于运算符请查阅本书附录。

(3) 数据结构类型丰富。除了 C 系统自有的数据结构，程序员还可以自创类型，所以 C 语言使用起来灵活、多样。

(4) 语法限制不太严格，程序设计自由度大。

(5) 具有结构化的控制语句。

(6) C 语言允许直接访问物理地址,能进行位 (bit) 操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此不能简单地把 C 语言说明为高级语言,它是一种介于汇编语言和高级语言之间的中级语言。

(7) 与汇编语言相比,用 C 语言编写的程序可移植性好。

(8) 生成目标代码质量高,程序执行效率高。

C 语言的特长不在科学计算和管理领域。对操作系统和系统实用程序以及需要对硬件进行操作的场合,使用 C 语言会明显地优越于其他高级语言,有些大型应用软件也是用 C 语言编写的。C 语言除了用于教学外,还有广泛的应用领域,因此更有生命力。

由于 C 语言对程序员较其他高级语言要高,使得程序员编写程序会感到限制很少、灵活性大、功能强,因而从理论上来说可以编写任何类型的程序。

但是, C 语言也有缺点。C 语言对程序员要求高,较其他高级语言在学习上要困难一些,尤其对于初学者。

1.2 简单的 C 程序介绍

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中可以了解到组成一个 C 源程序的基本部分和书写格式。

例 1.1

```
main()
{
    printf("I love china.\n");
}
```

说明:

- (1) main 是主函数的函数名,表示这是一个主函数。
- (2) 每一个 C 源程序都必须有且只能有一个主函数 (main 函数)。
- (3) printf 函数的功能是把要输出的内容送到显示器显示。
- (4) printf 函数是一个由系统定义的标准函数,可在程序中直接调用。

例 1.2

```
#include<stdio.h>                                /* include 被称为文件包含命令*/
main()
{
    double x,y,sum;                                /*定义 3 个实数变量,供后面使用*/
    printf("input number:\n");                    /*显示提示信息*/
    scanf("%lf%lf",&x,&y);                          /*从键盘获得一个实数 x*/
    sum=x+y;                                        /*求 x 和 y 的和,并把它赋给变量 sum*/
    printf("x= %lf y=%lf sum=%lf\n",x,y,sum);      /*显示程序运行结果*/
}                                                    /* main 函数结束*/
```

说明:

(1) C 程序是由函数构成的。一个 C 源程序至少包含一个 main 函数,也可以调用其他函数,本例中的 scanf() 和 printf() 就是 C 语言的系统函数,具有输入输出功能。

(2) 需要说明的是, C 语言规定对 scanf 和 printf 这两个函数可以省去对其头文件的包含命令。所以在本例

中也可以删去第 2 行的包含命令 `#include<stdio.h>`。

同样,在例 1.1 中使用了 `printf` 函数,也省略了包含命令。

(3) 在例 1.2 中的主函数体中又分为两部分,一部分为说明部分,另一部为分执行部分。说明是指对变量的类型说明。例 1.1 中未使用任何变量,因此无说明部分。C 语言规定,源程序中所有用到的变量都必须先说明,后使用,否则将会出错。这一点是编译型高级程序设计语言的一个特点。本例中使用了 3 个变量: `x`, `y`, `sum`, 用来表示输入的自变量用类型说明符 `double` 来说明。说明部分后的 4 行为执行部分(或称为执行语句部分),用以完成程序的功能。执行部分的第 1 行是输出语句,调用 `printf` 函数在显示器上输出提示字符串,请用户输入自变量 `x` 的值。第 2 行为输入语句,调用 `scanf` 函数,接受键盘上输入的数并存入变量 `x`, `y` 中。第 3 行是使用赋值表达式对变量 `sum` 进行赋值。第 4 行是用 `printf` 函数输出变量 `sum` 的值,程序结束。

(4) C 语言的格式非常自由,一行内可以书写几条语句,一条语句可以分写在多行上。C 程序是没有行号的,没有严格规定书写格式。

(5) 每个语句的最后必须有一个分号。分号是 C 语言的必要组成部分,是这种语言的特点。

运行本程序时,首先在显示器屏幕上给出提示串 `input number`, 这是由执行部分的第 1 行完成的。用户在提示下从键盘上键入某些数,如 56,按下回车键,接着在屏幕上给出计算结果。

1.3 C 语言的基本语法

1.3.1 C 语言源程序的结构特点

C 语言源程序主要有以下特点。

- (1) 一个 C 语言源程序可以由一个或多个源文件组成。
- (2) 每个源文件可由一个或多个函数组成。
- (3) 一个源程序不论由多少个文件组成,都有一个且只能有一个 `main` 函数,即主函数。
- (4) 源程序中可以有预处理命令(`include` 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。
- (5) 每条语句都必须以分号结尾。但预处理命令、函数头和花括号“`}`”之后不能加分号。
- (6) 标识符和关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

1.3.2 书写程序时应遵循的规则

从书写清晰及便于阅读、理解、维护的角度出发,在书写程序时应遵循以下规则。

- (1) 每条语句独占一行。
- (2) 用“`{}`”括起来的部分通常表示程序的某一层结构。“`{}`”一般与该结构语句的第一个字母对齐,并单独占一行。
- (3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便阅读更加清晰,增加程序的可读性。

在编程时应力求遵循这些规则,以养成良好的编程风格。

1.3.3 C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

注意: C 语言严格区分大小写字母, 即大写 ABC 和小写 abc 具有不同的含义。

空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用, 在其他地方出现时, 只起间隔作用, 编译程序对它们忽略不计。因此在程序中使用空白符与否, 对程序的编译不发生影响, 但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

1.3.4 C 语言主要术语

在 C 语言中使用的词汇分为 6 类: 标识符、关键字、运算符、分隔符、常量, 注释符。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除了库函数的函数名由系统定义外, 其余都由用户自定义。C 语言规定, 标识符只能是字母 (A~Z, a~z)、数字 (0~9)、下划线(_)组成的字符串, 并且其第一个字符必须是字母或下划线。

例如, 以下标识符是合法的:

a, x, x3, BOOK_1, sum5

而以下标识符是非法的:

6us 以数字开头

x*Pi 出现非法字符*

-ttx 以减号开头

hyuijn-1 出现非法字符- (减号)

在使用标识符时还必须注意以下几点。

(1) 标准 C 语言不限制标识符的长度, 但它受各种版本的 C 语言编译系统限制, 同时也受到具体机器的限制。例如, 在某版本 C 语言中规定标识符前 8 位有效, 当两个标识符前 8 位相同时, 则被认为是同一个标识符。

(2) 在标识符中, 大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义, 但标识符是用于标识某个量的符号。因此, 命名应尽量有相应的意义, 以便于阅读理解, 做到“顾名思义”。

2. 关键字

关键字是由 C 语言规定的具有特定意义的字符串, 通常也称为保留字。C 语言中的关键字都是小写的。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类。

(1) 类型说明符

用于定义、说明变量、函数或其他数据结构的类型。如前面例题中用到的 double 等。

(2) 语句定义符

用于表示一个语句的功能。如 if、else 就是条件语句的语句定义符。

(3) 预处理命令

用于表示一个预处理命令。如前面各例中用到的 include。

3. 运算符

C 语言中含有相当丰富的运算符。运算符与变量和函数一起组成表达式, 表示各种运算功能。运算符由一个或多个字符组成。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中, 分隔各个变量。空格多用于语句各单词之间, 作为间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔, 否则将会出现语法错误, 例如, 若把 “int a;” 写成 “inta;C”, 则编译器会把 “inta” 当成一个标识符处理, 其结果必然出错。

5. 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在后面章节中将专门给予介绍。

6. 注释符

C 语言的注释符是以 “/*” 开头并以 “*/” 结尾的串。在 “/*” 和 “*/” 之间的即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译程序跳过它们，不作处理，待调试结束后再去掉注释符。

1.4 Turbo C 2.0 集成开发环境的使用

1.4.1 Turbo C 2.0 简介和启动

我们上机实习和将来考试都是使用 Turbo C 2.0 版本。该系统是 DOS 操作系统支持下的软件，在 Windows 环境下，可以在 DOS 窗口下运行该系统。由于全国计算机等级考试上机环境要求在 Windows 下使用 Turbo C，因此本书不再介绍 DOS 下 Turbo C 的使用方法。

在 Windows 的 DOS 窗口下运行 Turbo C 2.0 时，只要在 Turboc2 子目录下键入 TC 并回车，即可进入 Turbo C 2.0 集成开发环境。

在 Windows 环境下，也可以选择“运行”菜单，然后键入相应的路径，如 c:\turboc2\tc 即可，也可以在 TC 文件夹找到 tc.exe 文件，然后双击该文件名，也可进入 Turbo C 2.0 集成开发环境。

Turbo C 是美国 Borland 公司的产品，Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件，如 Turbo BASIC，Turbo Pascal，Turbo Prolog，这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品，其中使用了全新的集成开发环境，即使用了一系列下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化，大大方便了程序的开发。1988 年，Borland 公司又推出 Turbo C 1.5 版本，增加了图形库和文本窗口函数库等，而 Turbo C 2.0 则是该公司 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能，并可以在 Tiny 模式下直接生成 COM（数据、代码、堆栈处在同一 64KB 内存中）文件，还可对数学协处理器（支持 8087/80287/80387 等）进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++，它继承发展了 Turbo C 2.0 的集成开发环境，并包含了面向对象的基本思想和设计方法。1991 年为了适用 Microsoft 公司的 Windows 3.0 版本，Borland 公司又将 Turbo C++ 做了更新，即 Turbo C 的新一代产品 Borland C++ 也已经问世了。

1.4.2 Turbo C 2.0 集成开发环境

进入 Turbo C 2.0 集成开发环境中后，屏幕上显示如图 1-1 所示的界面。

其中最上一行为 Turbo C 2.0 主菜单，中间区域为编辑区，接下来是信息区，最底下一行为参考行。这几部分构成了 Turbo C 2.0 的主屏幕，以后的编码、编译、调试以及运行都将在这个主屏幕中进行。

主菜单中，除 Edit 外，其他各项均有子菜单。只要用 Alt 键加上某菜单第一个字母，就可进入其的子菜单中。

1.4.3 常用菜单介绍

1. File 菜单

按 Alt+F 键可进入 File 菜单，如图 1-2 所示。

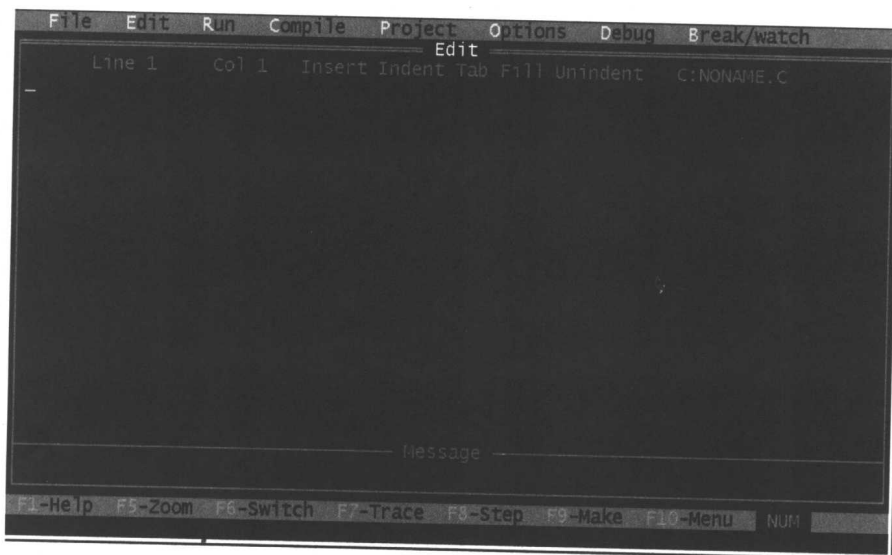


图 1-1

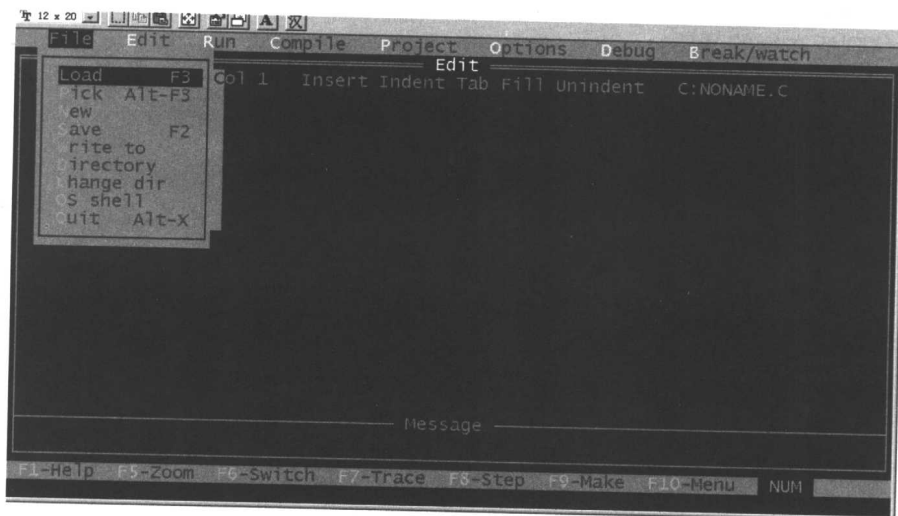


图 1-2

File 菜单的子菜单共有 9 项，分别叙述如下：

(1) Load: 装入一个文件，可用类似 DOS 的通配符（如*.C）来进行列表选择，也可装入其他扩展名的文件，只要给出文件名（或只给出路径）即可。该项的热键为 F3，即只要按 F3 键即可进入该项，而不需要先进入 File 菜单，再选择此项。

(2) Pick: 将最近装入编辑窗口的 8 个文件列成一个表，让用户选择，选择后将该程序装入编辑区，并将光标放在上次修改过的地方。其热键为 Alt+F3。

(3) New: 新建文件，默认文件名为 NONAME.C，存盘时可改名。

(4) Save: 将编辑区中的文件存盘，若文件名是 NONAME.C 时，将询问是否更改文件名，其热键为 F2。

(5) Write to: 可由用户给出文件名来将编辑区中的文件存盘，若该文件已存在，则询问是否覆盖。

(6) Directory: 显示目录及目录中的文件，并可由用户选择。

(7) Change dir: 显示当前默认目录，用户可以改变默认目录。

(8) Os shell: 暂时退出 Turbo C 2.0，回到 DOS 提示符下，此时可以运行 DOS 命令，若想回到 Turbo C 2.0 中，只要在 DOS 状态下键入 EXIT 即可。

(9) Quit: 退出 Turbo C 2.0, 返回到 DOS 操作系统中。其热键为 Alt+X。

说明: 以上各项可用光标键移动色棒进行选择, 回车则执行, 也可用每一项的第一个大写字母直接选择。若要退回主菜单或从它的下一级菜单列表框退回, 可用 Esc 键。Turbo C 2.0 所有菜单均采用这种方法进行操作, 以下不再说明。

2. Edit 菜单

按 Alt+E 键可进入 Edit 菜单, 若再回车, 则光标出现在编辑区, 此时用户可以进行文本编辑。可用 F1 键获得有关编辑方法的帮助信息。

与编辑有关的功能键如下:

F1: 获得 Turbo C 2.0 编辑命令的帮助信息;

F5: 扩大编辑区到整个屏幕;

F6: 在编辑区与信息区之间进行切换;

F10: 从编辑区转到主菜单。

常用编辑命令简介如下:

PageUp: 向前翻页;

PageDown: 向后翻页;

Home: 将光标移到所在行的开始;

End: 将光标移到所在行的结尾;

Ctrl+Y: 删除光标所在的一行;

Ctrl+T: 删除光标所在处的一个词。

说明:

(1) Turbo C 2.0 的双界符包括以下几种符号:

- 花括号 “{” 和 “}”
- 尖括号 “<” 和 “>”
- 圆括号 “(” 和 “)”
- 方括号 “[” 和 “]”
- 注释号 “/*” 和 “*/”
- 双引号 “”
- 单引号 ‘’

(2) Turbo C 2.0 在编辑文件时还有一种功能, 就是能够自动缩进, 即光标定位和上一个非空字符对齐。在编辑区中, Ctrl+OL 为自动缩进开关的控制键。

3. Run 菜单

按 Alt+R 键可进入 Run 菜单, 该菜单有以下各项, 如图 1-3 所示。

(1) Run: 运行由 Project/Project name 项指定的文件名或当前编辑区的文件。如果上次编译后的源代码未做过修改, 则直接运行到下一个断点 (没有断点则运行到结束); 否则先进行编译、连接后才运行。其热键为 Ctrl+F9。

(2) Program reset: 中止当前的调试, 释放分给程序的空间。其热键为 Ctrl+F2。

(3) Go to cursor: 调试程序时使用, 选择该项可使程序运行到光标所在行。光标所在行必须为一条可执行语句, 否则提示错误。其热键为 F4。

(4) Trace into: 在执行一条调用其他用户定义的子函数时, 若用 Trace into 项, 则执行长条将跟踪到该子函数内部去执行。其热键为 F7。

(5) Step over: 执行当前函数的下一条语句, 即使用户函数调用, 执行长条也不会跟踪进函数内部。其热键为 F8。

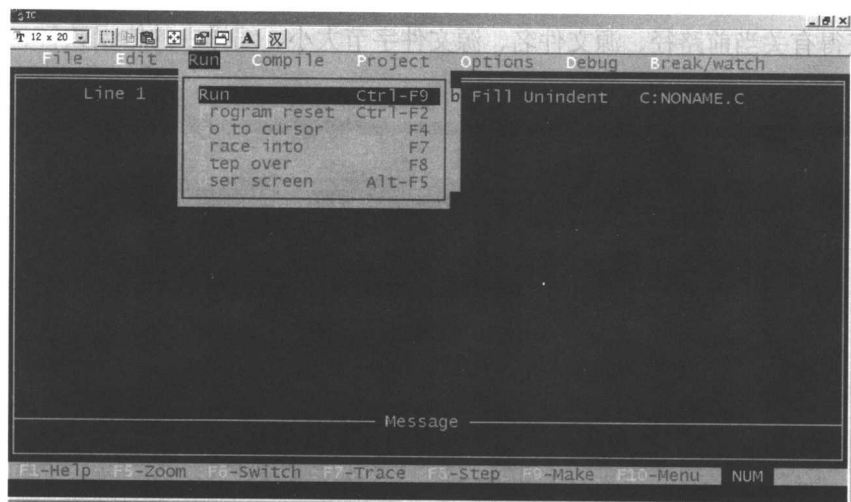


图 1-3

(6) User screen: 显示程序运行时在屏幕上显示的结果。其热键为 Alt+F5。

4. Compile 菜单

按 Alt+C 键可进入 Compile 菜单, 该菜单有以下几个菜单项, 如图 1-4 所示。

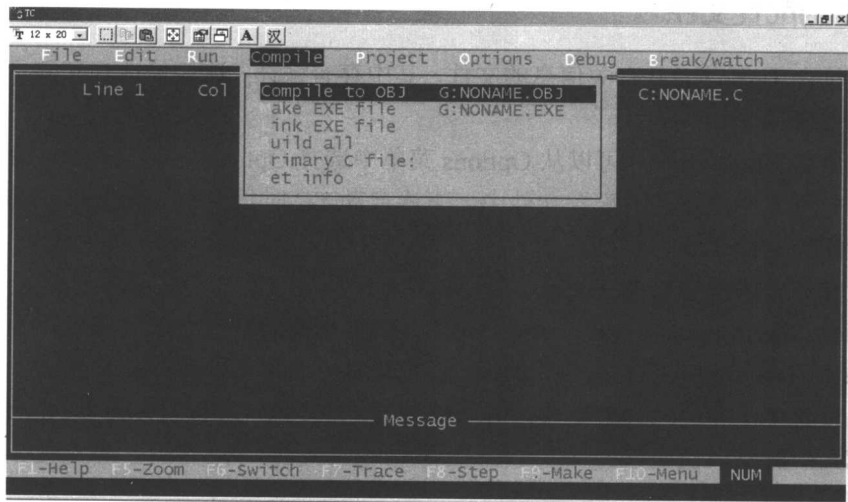


图 1-4

(1) Compile to OBJ: 将一个 C 源文件编译生成 OBJ 目标文件, 同时显示生成的文件名。其热键为 Alt+F9。

(2) Make EXE file: 此命令生成一个 .EXE 的文件, 并显示生成的 .EXE 文件名。其中 .EXE 文件名是下面几项之一:

- 由 Project/Project name 说明的项目文件名;
- 若没有项目文件名, 则由 Primary C file 说明的源文件;
- 若以上两项都没有文件名, 则为当前窗口的文件名;

(3) Link EXE file: 把当前 .OBJ 文件及库文件连接在一起生成 .EXE 文件。

(4) Build all: 重新编译项目里的所有文件并进行装配, 生成 .EXE 文件。该命令不作过时检查 (上面的几条命令要作过时检查, 即如果目前项目里源文件的日期和时间与目标文件相同或更早, 则拒绝对源文件进行编译)。

(5) Primary C file: 当在该项中指定了主文件后, 在以后的编译中, 如没有项目文件名, 则编译此项中规定的主 C 文件, 如果编译中有错误, 则将此文件调入编辑窗口, 不论目前窗口中是不是主 C 文件。

(6) Get info: 获得有关当前路径、源文件名、源文件字节大小、编译中的错误数目、可用空间等信息, 如图 1-5 所示。

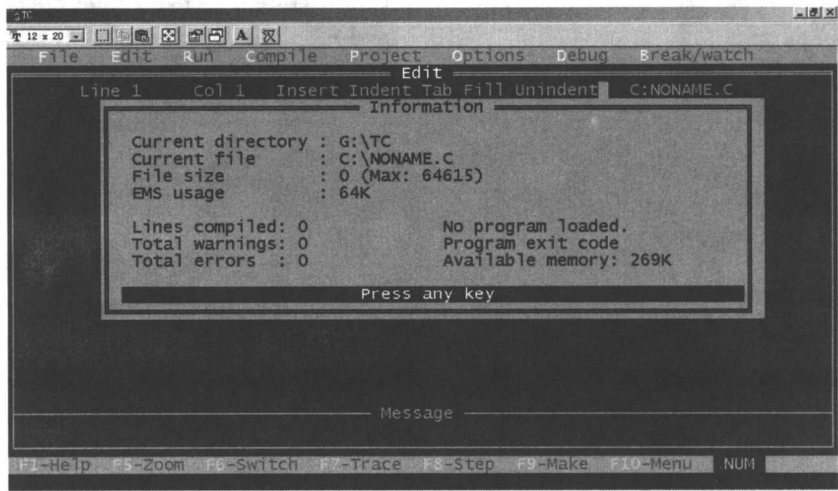


图 1-5

1.4.4 Turbo C 2.0 的配置文件

所谓配置文件是包含 Turbo C 2.0 有关信息的文件, 其中存有编译、连接的选择和路径等信息。可以用下述方法建立 Turbo C 2.0 的配置。

(1) 建立用户自命名的配置文件: 可以从 Options 菜单中选择 Options/Save options 命令, 将当前集成开发环境的所有配置存入一个由用户命名的配置文件中。下次启动 TC 时只要在 DOS 窗口下键入:

tc/c<用户命名的配置文件名>

就会按这个配置文件中的内容作为 Turbo C 2.0 的选择。

(2) 若设置 Options/Environment/Config auto save 为 ON, 则退出集成开发环境时, 当前的设置会自动存放到 Turbo C 2.0 配置文件 TCCONFIG.TC 中。Turbo C 在启动时会自动寻找这个配置文件。

(3) 用 TCINST 设置 Turbo C 的有关配置, 并将结果存入 TC.EXE 中。Turbo C 在启动时, 若没有找到配置文件, 则取 TC.EXE 中的默认值。