

行起来。1972 年至 1973 年,贝尔实验室的 D. M. Ritchie 在 B 语言基础上设计出了 C 语言,取 BCPL 中的第二个字母命名。

C 语言既保持了 BCPL 和 B 语言语法精练、接近于硬件的优点,又克服了它们过于简单、无数据类型等缺点。最初的 C 语言只是为描述和实现 UNIX 操作系统提供了一种工作语言而设计的。1973 年 K. Thompson 和 D. M. Ritchie 两人合作,把原来由两人用汇编语言编写的 UNIX 操作系统中 90% 以上的代码用 C 语言来重写,即 UNIX 5。后来 C 语言作了多次改进,不过它主要还是用在贝尔实验室内部。直到 1975 年用 C 语言写的 UNIX 6 公布后才引起业内人士的广泛关注。1978 年以后,C 语言已先后移植到大、中、小型计算机和微型计算机上,已独立于 UNIX 操作系统。

以 1978 年发表的 UNIX 7 中的 C 编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie(合称 K&R)合著了影响深远的著作 *The C Programming Language*。该书中介绍的 C 语言成为后来被广泛使用的 C 语言的基础,称它为标准 C。以后相继出现了很多 C 语言版本,如 Microsoft C、Turbo C、Quick C、Borland C 等等。它们在语法上基本上是相同的,但在函数数量和功能上有较大的区别,用时要注意区分。1983 年美国国际标准化协会(ANSI)根据 C 语言问世以来的各种版本对 C 的发展和扩充,制定了新的标准,称为 ANSI C。1988 年,K&R 按照 ANSI C 重写了 *The C programming Language*。1990 年,国际标准化组织 ISO(International Standard Organization)接受以 87 ANSI C 为 ISO C 的标准(ISO 9899—1990)。目前流行的 C 编译系统都是以它为基础的。

高级语言发展到现在,面向对象的程序设计方法越来越受到人们的青睐。比如:Visual FoxPro(VFP)、Visual Basic(VB)、Visual C++(VC++)、C++、Turbo C++、Borland C++、Java、J++、Power Builder(PB)等等。其中,功能比较强大的还是 C++ 语言,而这个面向对象的语言是以 C 语言为基础的。只有学好 C 语言,才能最终学习和掌握 C++。VC++、Borland C++、Turbo C++ 等都是 C++ 语言的编程环境。其中,VC++ 和 Borland C++ 的功能与编程环境深受广大程序设计人员的好评。

1.1.2 C 语言的特点

C 语言之所以能够在众多的高级语言竞争中脱颖而出,成为高级语言中的佼佼者,这主要是因为与普通高级语言相比,C 语言具有以下特点。

① 它的语言简洁、紧凑,使用方便、灵活。C 语言一共有 32 个关键字,9 种控制语句,程序书写形式自由,主要用小写字母表示。

② 数据类型丰富。C 语言数据类型包括整型、实型、字符型、枚举类型;结构体、共用体、数组和文件类型;指针类型;空类型。其中,整型、实型、字符型还有多种小的类型。用这些数据类型可以表示各种各样的数据结构(如链表、树、栈等)。指针类型是 C 语言中最具特点的一种数据类型,它使用起来非常灵活自如,把 C 语言的功能特点发挥得淋漓尽致,但是,由于指针在运用上非常灵活,所以,它也正是初学者最难驾驭之处。

③ 运算符多样。C 语言中的运算符包含的范围非常广泛,一共有 44 个运算符。除包括算术运算符、关系运算符、逻辑运算符等常规运算符之外,还含有指针运算符、地址运算符、位运算符、自增自减运算符、条件运算符、复合的赋值运算符,甚至连圆括号、方括

号、逗号、小数点等都是运算符。由于C语言的运算类型极为丰富,表达式类型多样化,所以用C语言能够实现各种各样的高级和低级运算。

④ 函数是程序的主体。C语言中每一项功能的完成都是由函数来实现的,可以由C系统中已提供的功能函数来实现,也可以是用户自编函数来实现。函数是程序的基本单位,是C语言的灵魂。

⑤ 语法检查不太严格,程序书写自由度大。比如,数组下标不作超界检查;整型、字符型、逻辑型量可以通用;一个物理行可以写多个语句,一个语句也可以分在连续的多个的物理行上。对这些规则,初学者要认真掌握。

⑥ C语言允许直接访问物理地址。C语言中含有位运算和指针运算,能够直接对内存地址进行访问操作,可以实现汇编语言的大部分功能,即直接对硬件进行操作。所以,它既具有高级语言的功能,又兼有汇编语言(低级语言)的大部分功能。有时,也称它为“中间语言”或“中级语言”。可以说,它是高级语言中的低级语言。

⑦ 生成的目标代码质量高。它比一般的高级语言生成的目标代码质量高约20%,但还是要比汇编语言低10%~20%。这在高级语言中已是出类拔萃的了。

⑧ 可移植性好。同其他高级语言一样,C语言程序不作大的修改就可以很容易地移植到其他类型的机器上去。

目前较为流行的高级语言主要有: BASIC、Pascal、Fortran、C及面向对象的和可视化的编程语言。BASIC是初学者指令代码,Pascal是一种教学语言,Fortran是用于数值计算的高级语言。C语言能完成各种功能,是其他高级语言无法比拟的,所以,越来越多的人开始学习和使用C语言进行程序设计。

1.1.3 C语言中32个关键字和语句形式

1. 32个关键字

① 程序控制语句关键字(12个):

if else for do while continue
switch break case default goto return

② 类型定义说明关键字(12个):

int char float double long short
signed unsigned enum struct union void

③ 存储类别定义说明关键字(4个):

auto register static extern

④ 常量、变量定义和自定义类型关键字(3个):

const volatile typedef

⑤ 字节测试关键字(1个):

sizeof

2. 语句形式

① 控制语句:

```

if( ~ ) ~ else ~          /* 选择分支语句 */
switch( ~ ) case ...;    /* 多分支语句 */
for( ~ ; ~ ; ~ ) ~       /* for 循环语句 */
while( ~ ) ~             /* while 循环语句 */
do ~ ; while( ~ );       /* do while 循环语句 */
continue;              /* 无条件进行下一次循环语句 */
break;                 /* 无条件结束当前层循环或跳出 switch 语句 */
goto ~                  /* 无条件跳转到程序指定处语句 */
return ~                /* 函数调用结束返回语句 */

```

② 函数调用语句:

函数名(实参表达式表);

③ 表达式语句:

表达式;

④ 空语句:

```

;或{}                      /* 不作任何操作 */

```

⑤ 复合语句(分程序):

```

{ ~ }

```

⑥ 注释语句:

```

/* ~ */                    /* 注释的部分不参加程序编译和运行 */

```

其中“~”代表一个或多个表达式或语句,括号内为表达式,括号外为语句。

1.1.4 程序的三种基本结构与流程图简介

1966 年,Bohra 和 Jacopini 提出了程序的三种结构,即顺序结构、分支结构和循环结构。流程图是用一些图框来表示程序或算法的运行走向的一种图示。用图形表示算法或程序的走向,直观形象、容易理解。美国国家标准化协会(ANSI)规定了一些常用的流程图符号(如图 1.1 所示),已被程序工作者所普遍采用。

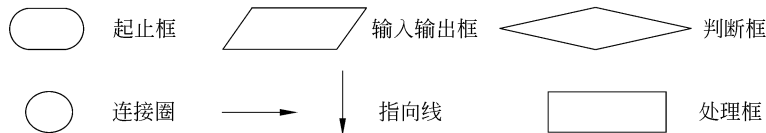


图 1.1 常用的流程图符号

1. 三种基本结构与其对应的流程图

(1) 顺序结构

程序由上至下每个语句顺序执行的结构称为顺序程序结构。与其对应的流程图见图 1.2。

(2) 分支结构

程序在从上至下顺序执行过程中遇到了一个岔路口,就要选择程序执行从哪条路走,关键是只能选择一条路,这种程序结构就称为分支程序结构。与其对应的流程图见图 1.3 和图 1.4。

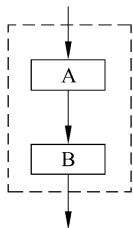


图 1.2 顺序结构

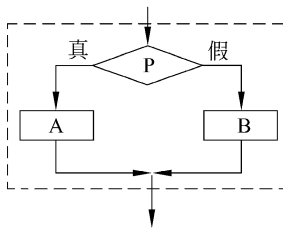


图 1.3 分支结构 1

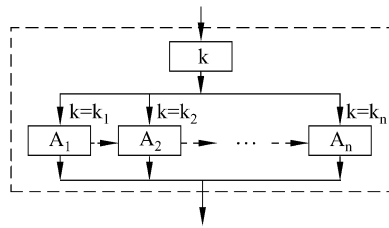


图 1.4 分支结构 2

(3) 循环结构

程序在从上至下顺序执行的过程中遇到了一个圈子,程序必须转完足够的圈子之后才能走出这个圈子,这种程序结构就称为循环程序结构。与其对应的流程图见图 1.5 和图 1.6。

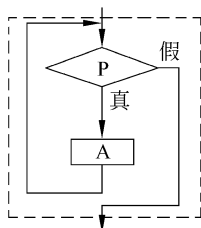


图 1.5 循环结构 1

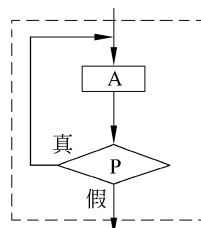


图 1.6 循环结构 2

【例 1.1】 根据年份 year 判断当年是否为闰年的框图(见图 1.7)。

【例 1.2】 判断一个整数 m 是否为素数的流程图(见图 1.8)。

2. N-S 图简介

从上面介绍可以看出,程序总是由上至下顺序执行的,那么基本结构之间的流向线就是多余的了。1973 年,美国学者 I. Nassi 和 B. Shneiderman 提出了一种新的流程图形式,称为 N-S 结构化流程图,简称 N-S 图。其对应的三种结构的流程图见图 1.9。

【例 1.3】 将例 1.1 改造成 N-S 图(见图 1.10)。

【例 1.4】 将例 1.2 改造成 N-S 图(见图 1.11)。

用框图来描述算法和程序的走向还有许多其他方式,在此,我们就不一一介绍了。根据框图,编程人员就很容易编写程序了,所以,框图设计好坏会直接影响编程的质量。但是,初学者是在掌握 C 语言语法和练习阶段,所涉及的程序一般都不会很大,也不很难,所以,也没有必要非得先画出程序流程图再进行编程不可。

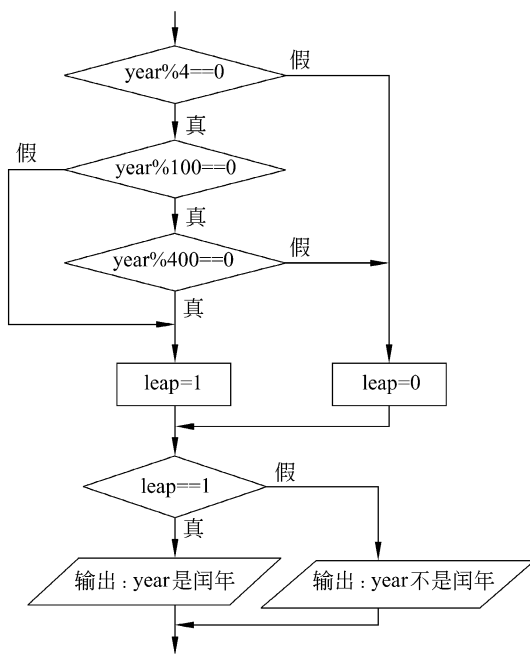


图 1.7 例 1.1 图

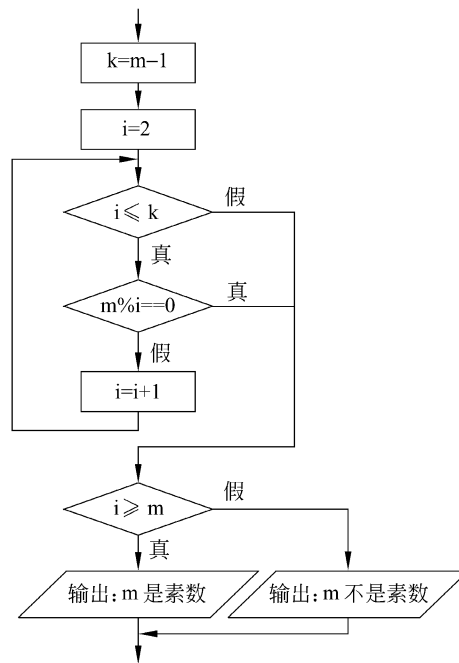
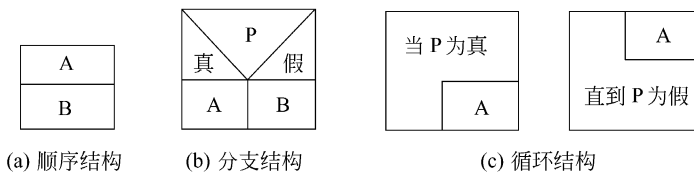


图 1.8 例 1.2 图



(a) 顺序结构

(b) 分支结构

(c) 循环结构

图 1.9 N-S 结构化流程图

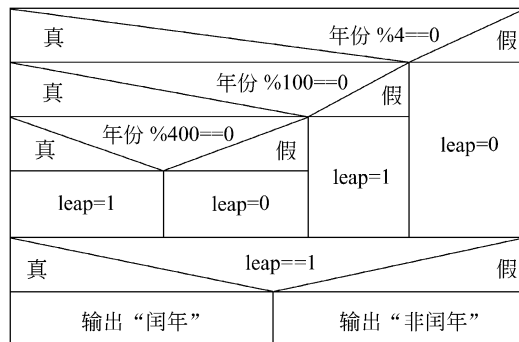


图 1.10 例 1.3 图

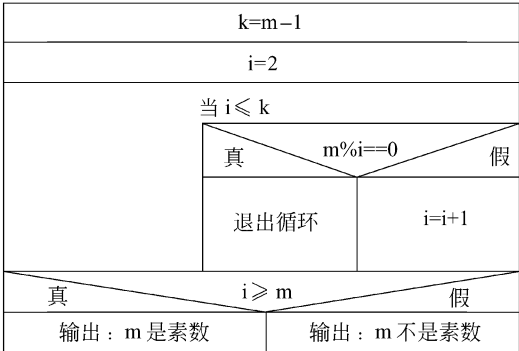


图 1.11 例 1.4 图

1.2 Turbo C 上机操作指导

Turbo C 是 C 语言的编程环境之一,它是在 DOS 界面下的一个较好的编程环境。只有对这个编程环境有个大体上的了解和掌握,才能进行上机调试程序。即使程序编得再好,如果不去上机调试,也很难保证程序是正确的,所以要想学好 C 语言,上机操作是一个非常重要的环节。完成一个设计题目基本上应按如下这几个步骤进行:

- ① 分析问题、描述算法;
- ② 画出算法流程图;
- ③ 编程;
- ④ 上机输入程序;
- ⑤ 调试程序;
- ⑥ 分析运行结果的正确性与合理性;
- ⑦ 写出完成报告。

其中,对于初学编程的人员来说,编程和调试阶段是难点,但是只要努力去做,就没有做不好的事。

1.2.1 Turbo C 用户窗口界面

通常,Turbo C 被安装在 C 盘或 D 盘的 TC 目录(在 Windows 中称为文件夹)下。这样只要找到 TC 文件夹,双击其中的 tc. exe 可执行文件就可启动 Turbo C 了。也可以利用 Turbo C 的快捷方式启动它。Turbo C 被启动后将出现一个蓝底的窗口,可以使用 Alt+Enter 键在全屏和 Windows 窗口之间进行切换。启动后的 Turbo C 界面如图 1.12 所示。

这时的界面看上去很乱,可以单击  按钮或按下 Alt+Enter 组合键使之变成全屏的显示方式,见图 1.13。其中顶行为菜单行,底行为功能键行,中间为 Turbo C 的显示区。Edit 区为编辑区,Edit 区的顶行为状态行,描述了当前编辑光标所在的行列位置、是否设置了插入状态、被编辑修改的文件名等等。Message 区为信息区,可以通过用户设置来查

看程序运行状态下的各个量的当前值。下面我们分别介绍 Turbo C 中各个菜单项的功能。

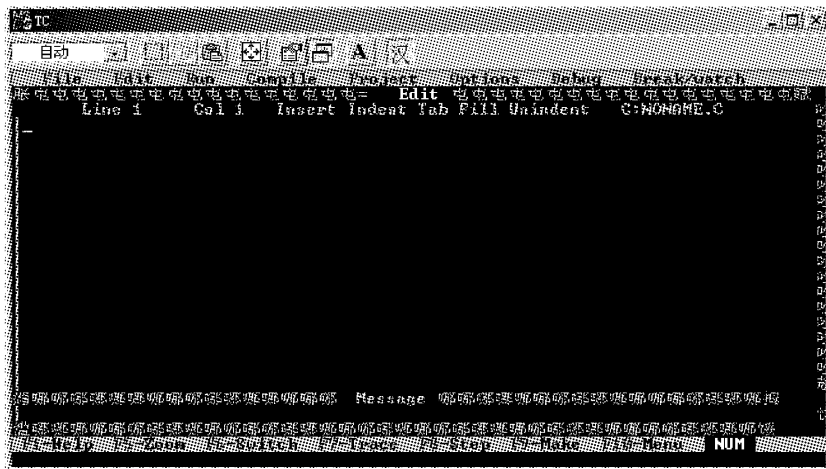


图 1.12 Windows 窗口下的 Turbo C 界面



图 1.13 全屏显示的 Turbo C 界面

1.2.2 Turbo C 的主要菜单功能

在 Turbo C 中比较常用的菜单项有：File、Edit、Run、Compile 和 Options。掌握了这几项，对本课程的学习已经足够。当然，自己能多掌握点更好。打开各个菜单项的基本方法：一是用 Alt+X(X 表示菜单项名的第一个字母)组合键，例如：Alt+F 键打开文件菜单、Alt+E 键直接进入编辑状态等等；二是按下 F10 功能键，再用光标移动键找到菜单项，然后再用回车键确定即可。下列菜单小项中，菜单小项右侧对应的是快捷键，即直接按快捷就可以完成该项功能。

1. File 菜单项

它的各项菜单见图 1.14。各个小项的功能为：

Load(F3)：打开指定文件。

Pick(Alt+F3)：打开(拾取)当前编辑过的文件。

New：建立新文件(临时文件名为 NONAME.C)。

Save(F2)：将当前正在编辑的文件存盘。如果是第一次按此方式存盘，则还要给出文件名。

Write to：将正在编辑的文件以另一个文件名称写入到磁盘文件中。

Directory：获取当前文件列表，并可查看目录，选择一个文件装入编辑。

Change dir：显示或修改当前目录。

Os shell：转到 DOS 界面工作。这时若想返回到 Turbo C 环境，则必须使用 EXIT 命令，切记！

Quit(Alt+X)：退出 Turbo C 环境，返回到操作系统界面。这时，如果用户未存盘，则系统会提示请用户存盘。

2. Edit 菜单项

它下面没有小项。使用此项功能可直接进入到编辑状态。

3. Run 菜单项

它的各项菜单见图 1.15。各个小项的功能如下：

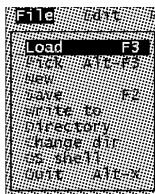


图 1.14 File 菜单项

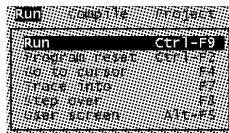


图 1.15 Run 菜单项

Run(Ctrl+F9)：运行程序。可直接进行编译连接，生成目标程序。

Program reset(Ctrl+F2)：程序复位。

Go to cursor(F4)：程序运行到当前光标处。

Trace into(F7)：跟踪执行方式。可使程序按步执行到函数中。

Step over(F8)：单步执行程序方式。可使程序单个语句执行，但不进入到函数中。

User screen(Alt+F5)：查看程序运行结果界面。这时，按任意键将返回到 Turbo C 环境中。

4. Compile 菜单项

它的各项菜单见图 1.16。各个小项的功能为：

Compile to OBJ C:NONAME.OBJ: 编译指定的文件,生成. OBJ 文件。

Make EXE file C:NONAME.EXE: 检查日期,连接对应的. OBJ 文件,生成. EXE 可执行文件。

Link EXE file: 不检查日期,直接连接生成可执行文件。

Build all: 不检查日期,编译连接本程序中的所有文件,生成对应的目标程序文件。

Primary C file: 为编译和 make 规定编译连接的文件名。

Get info: 显示出现一个窗口,提供正在编辑的文件信息。

5. Project 菜单项

它的各项菜单项见图 1.17。各个小项的功能为：

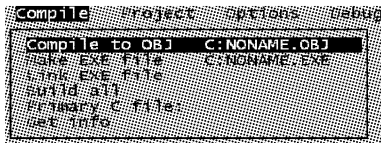


图 1.16 Compile 菜单项

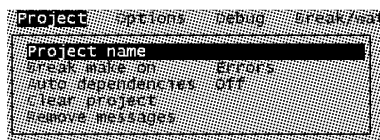


图 1.17 Project 菜单项

Project name: 工程文件名。当一个程序由多个 C 源程序文件构成时,这小项是有用的。

Break make on Errors: 设置出错中断执行程序方式。

Auto dependencies off: 设置自动更新文件与否的方式。

Clear project: 清除工程文件名和信息窗口。

Remove messages: 删除信息窗口内容。

这些小项有很多是系统设置参数,一般不需要用户对此进行修改。

6. Options 菜单项

它的各项菜单项见图 1.18。各个小项的功能为：

Compiler: 用于设置编译器。

Linker: 用于设置连接器。

Environment: 用于系统环境设置。

Directories: 用于设置系统文件路径。

Arguments: 允许使用命令行参数。

Save options: 保存系统参数设置。

Retrieve options: 恢复系统默认参数设置。

这一项主要用于系统参数设置。其中的“Directories”小项可能会经常用到。

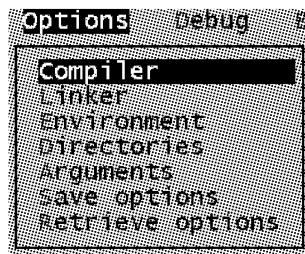


图 1.18 Options 菜单项

7. Debug 菜单项

它的各项菜单项见图 1.19。各个小项的功能为：

Evaluate(Ctrl+F4): 设置计算公式、显示结果、赋给新值。

Call stack(Ctrl+F3): 检查堆栈情况。

Find function: 显示规定函数。

Refresh display: 刷新显示。用于编辑窗口偶然被用户窗口重写时。

Display swapping Smart: 使显示窗口在用户窗口和编辑窗口之间改变。

Source debugging On: 设置源调整跟踪方式打开。

8. Break/watch 菜单项

它的各项菜单见图 1.20。各个小项的功能为:

Add watch(Ctrl+F7): 增加数据观察窗口。

Delete watch: 删除数据观察窗口中的最近的或指定的表达式。

Edit watch: 编辑数据观察窗口。

Toggle breakpoint(Ctrl+F8): 对当前行设置或清除程序断点。

Clear all breakpoints: 清除程序中所有的断点。

View next breakpoint: 光标定位到下一个断点。

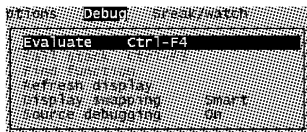


图 1.19 Debug 菜单项

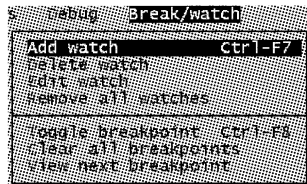


图 1.20 Break/watch 菜单项

这项菜单是用于调试程序的。可以在程序运行时观察各个变量的值,可以设置程序断点,使程序运行到断点处自动停下来,以便观察各个量的变化情况。这对调试程序是很有用的。

在上述各个菜单小项中,我们比较常用的项是: Open(F3)、Pick(Alt+F3)、New、Save(F2)、Quit(Alt+X)、Edit、Run(Ctrl+F9)、Trace into(F7)、Step over(F8)、User screen(Alt+F5)、Compile to OBJ、Make EXE file、Built all、Project name、Directories、Save options、Evaluate(Ctrl+F4)、Add watch(Ctrl+F7)、Clear all breakpoints 等等。在上机实验过程中,一定要弄清楚这些小项的功能及用法。

1.2.3 获取帮助与编辑操作功能

在不太熟悉 Turbo C 环境时,获取帮助是必须的。在 Turbo C 中获取帮助有两种方式:一是按 F1 键,进入编辑操作帮助状态;二是按 Ctrl+F1 组合键进入语句帮助状态。常用的编辑操作键及其功能介绍如下:

1. 光标移动

① ↑、↓、←、→键: 上移一行、下移一行、左移一个字符、右移一个字符。