

第1章 初识 C 程序

主要内容

- 开发环境
- 简单的 C 程序

贝尔实验室是人们熟悉的著名实验室之一,C 语言就诞生于该实验室,主要发明人是该实验室的 Dennis Ritchie。1972 年,Dennis Ritchie 和同事 Ken Thompson 为了编写 UNIX 操作系统设计了一种新的实用语言,并使用该语言完成了 UNIX 操作系统的设计。Dennis Ritchie 在设计该语言时借鉴了同事 Ken Thompson 开发命名的 B 语言,由于 B 语言的名字来自当时的 BCPL(Basic Combined Programming Language)语言的第一个字母,为此,Dennis Ritchie 将其发明的新语言命名为 C 语言(取 BCPL 语言的第二个字母)。1978 年,Dennis Ritchie 出版了第一部 C 语言著作 *The C Programming Language*。1983 年,美国国家标准化组织(ANSI)根据 Dennis Ritchie 的著作确定出了第一个 C 语言标准——C89,习惯上将 C89 称为 ANSI C(标准 C)。

C 语言高效、灵活,被广泛应用于科学计算,可以在许多软件开发中看到 C 语言的影子,在 TIOBE 定期公布的语言排行榜上,C 语言经常和 Java 语言一起处于前二位(也曾多次处于第一位)。C 语言是非常适合用于编写操作硬件设备的编程语言,特别是随着信息时代的不断进步,嵌入式系统的开发越来越流行,比如人们使用 C 语言为汽车、飞行器、机器人、照相机、冰箱、DVD 播放机和其他许多现代化设备中的微处理器编写程序,以便控制这些设备,简言之,嵌入式领域更加注重代码的效率,因此 C 语言在嵌入式领域独领风骚。

C 语言也是进一步学习其他高级语言,如 C++、Java 和 C#(读作 see-sharp),以及相关的课程,如数据结构、操作系统、微机原理等课程的基础语言。

1.1 开发环境

1.1.1 基础知识

学习任何一门编程语言都需要选择一种针对该语言的开发工具。针对 C 语言的开发工具的核心任务之一就是人们按照该语言编写的代码(称为源文件)转变成操作系统能够识别执行的指令(也称为机器指令)。

本书将采用 Visual C++(本书简称 VC++)6.0 来开发 C 程序,VC++ 6.0 是比较流行和成熟的集成开发环境,集成开发环境习惯上称为 IDE(Integrated Development Environment)。VC++ 6.0 不仅是适合开发 C++ 程序的开发工具,同时也适合用来开发 C 程序,而且国内有关 C 语言的二级认证考试已经开始使用 VC++ 6.0,因此,使用 VC++ 6.0 有利于学生更好地熟悉认证考试环境。

1.1.2 基础训练——安装 VC++ 6.0

训练的能力目标是安装、配置 VC++ 6.0,能启动 VC++ 6.0。

1. 训练的主要内容

- 安装 VC++ 6.0
- 配置 Include files
- 配置 Library files

2. 训练步骤

(1) 安装 VC++ 6.0

双击 MSDEV.EXE, 将 VC++ 6.0 安装到某个目录, 这里将默认的安装目录修改为 D:\VC6.0 (也可以使用默认的安装目录)。安装完毕后, 单击“开始”→“程序”→“VC++ 6.0”命令启动 VC++ 6.0, 出现如图 1.1 所示的开发界面。

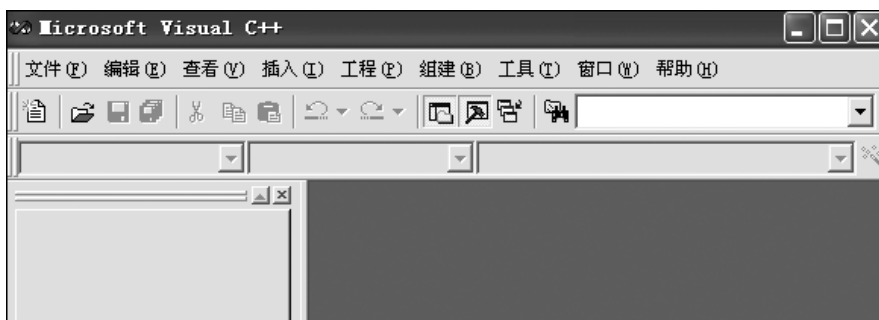


图 1.1 VC++ 6.0 的开发界面

(2) 配置 Include files

接下来配置 Include files (如果使用默认安装目录, 安装时已经由系统自动配置过, 可不必重新配置)。在图 1.1 所示的开发界面上选择“工具”→“选项”命令, 在弹出的图 1.2(a)所示的对话框中选择“目录”选项卡来配置 Include files。如果安装 VC++ 6.0 时修改了安装目录, 只需将 Include files 重新配置为新的目录即可。比如, 修改后的目录是 D:\VC6.0, 那么将目录列表中的 Include files 选项对应的“路径[D]:”做如下更改:

```
D:\VC6.0\VC98\INCLUDE
D:\VC6.0\VC98\MFC\INCLUDE
D:\VC6.0\VC98\ATL\INCLUDE
```



(a) 配置 Include files



(b) 配置 Library files

图 1.2 配置设置

(3) 配置 Library files

然后配置 Library files (如果使用默认安装目录, 安装时已经由系统自动配置过, 可不必重新配置)。在图 1.1 所示的开发界面上选择“工具”→“选项”命令, 在弹出的图 1.2(b)所示的对话框中选择“目录”选项卡, 并在“目录”下拉列表中选择 Library files 来配置 Include files。图 1.2(b)所示的对话框显示的是默认配置。如果安装 VC++ 6.0 时修改了安装目录, 只需将 Library files 重新配置为新的目录即可。比如, 修改后的目录是 D:\VC6.0, 那么将目录列表中的 Library files 选项对应的“路径[D]:”做如下更改:

```
D:\VC6.0\VC98\LIB  
D:\VC6.0\VC98\MFC\LIB
```

3. 训练小结与拓展

学习配置 Include files 和 Library files 的目的是能使用 VC++ 6.0 中的库函数。另外,尽管学习 C 语言可以选择任意一个成熟的开发工具,但 VC++ 6.0 是目前比较流行的开发工具之一。

1.1.3 上机实践——启动 VC++ 6.0

除可以单击“开始”→“程序”→VC++ 6.0 命令启动 VC++ 6.0 外,也可以执行 MSDEV.EXE 来启动 VC++ 6.0。假如安装目录是 D:\VC6.0,那么可以双击 D:\VC6.0\COMMON\MSDEV98\BIN 目录中的 MSDEV.EXE 文件启动 VC++ 6.0。

1.2 简单的 C 程序

1.2.1 基础知识

无论 C 程序的规模大小,开发一个 C 程序都需经过如下 4 个基本步骤。

1. 源文件

编写源文件,也称为编写源代码文件。所谓源文件就是按 C 语言的语法规则,使用文本编辑器编写的扩展名为.c 的文本文件,如 first.c、second.c 等,也就是说,C 程序的源代码存放在扩展名为.c 的文本文件中。

2. 编译

操作系统不能直接运行源代码文件,因此必须把源代码文件转化为操作系统能够识别的机器指令。编译器将检查源代码文件中是否有语法错误,如果有语法错误,将提示有关错误;如果没有语法错误,编译器就会将源代码文件转化为二进制文件,该二进制文件被称为源代码文件的目标文件。目标文件的名字与源代码文件的名字相同,但扩展名为.obj。

3. 链接

目标文件是供链接器使用的文件(操作系统不能直接运行目标文件),也就是说,目标文件中含有待确定的链接信息,链接器必须把这些信息替换成真正的链接代码,形成一个完整的可执行的代码。即链接器将程序使用的库函数以及其他函数链接在一起(比如 printf() 函数就是库函数),并向可执行文件中加入操作系统的启动代码,形成一个完整的可执行的文件(扩展名是.exe)。没有启动代码操作系统就无法运行可执行文件,启动代码相当于程序和操作系统之间的接口。所以 C 程序得到的可执行文件依赖于操作系统,即在 Window 操作系统下编译得到的可执行文件,可能无法在 Linux 操作系统上正确运行或不能在 Linux 操作系统上运行(需要在 Linux 操作系统上重新编译、链接得到可执行文件)。

需要注意的是,如果不小心将库函数 printf() 函数的名字误写为 qprintf,那么在编译阶段可能无任何错误,但在链接阶段就可能报错,其原因是找不到库函数 qprintf() 或用户自己定义的 qprintf() 函数。

4. 运行

产生可执行文件后,就可以把该文件交给操作系统去执行。

开发一个 C 程序的基本步骤如图 1.3 所示。

1.2.2 基础训练——程序开发步骤

训练的能力目标是能够编写一个简单的 C 程序的源文件,并编译该源文件、链接目标文件得到可执行文件,然后运行可执行文件,即运行程序。

1. 训练的主要内容

编写一个简单程序,该程序输出两行文字:“很高兴学习 C 语言”和“We are students”。

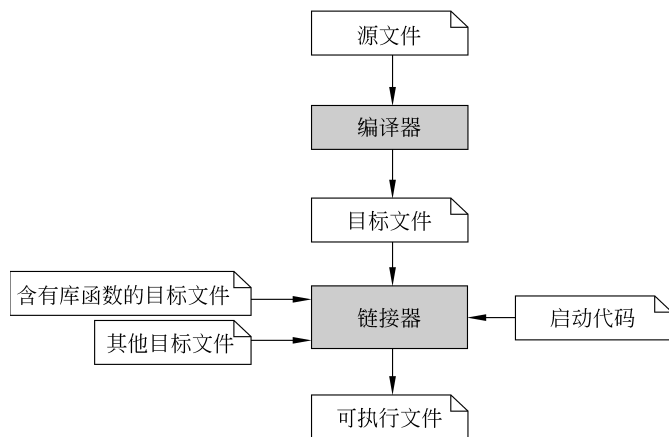


图 1.3 C 程序的开发步骤

2. 训练步骤

训练的 6 个基本步骤是：创建工程、向工程中添加源文件、编写源文件、编译、链接和运行。

(1) 创建名字是 project1_1 的工程

VC++ 使用一个工程对应一个 C 程序,也就是说,在 VC++ 环境中,通过创建一个工程来创建一个 C 程序。

创建工程的步骤如下。

① 在 VC++ 开发界面上单击“文件”菜单,选择其中的“新建”命令,在弹出的“新建”对话框中选择“工程”选项卡。

② 在当前对话框左侧的选项列表中选中 Win32 Console Application (注意,不可以选择 Win32 Application)。

③ 在当前对话框右侧的“位置”文本框中输入存放工程的位置,比如 C:\C_program。

④ 在当前对话框的“工程名称”文本框中输入工程的名称,比如 project1_1。

⑤ 在弹出的“选择工程类型”对话框中选择“空工程(An empty project)”。

⑥ 在工作空间界面(VC++ 开发界面的左侧)的下方选择 FileView 视图。

(2) 向工程中添加名字是 example1_1.c 的源文件

向工程中添加源文件的步骤如下。

① 在 VC++ 开发界面上单击“文件”菜单,选择“新建”命令。

② 在弹出的“新建”对话框中选择“文件”选项卡。

③ 在当前对话框左侧的选项列表中选中 C++ Source File。

④ 在右侧的“文件名称”文本框中输入源文件的名称,比如 example1_1.c (必须带扩展名.c)。

⑤ 选中“添加到工程”复选框,即把源文件添加到工程中。

(3) 编写源文件 example1_1.c

在程序代码编辑区(VC++ 6.0 开发环境提供的一个文本编辑器)输入如下源代码。

example1_1.c 源文件的内容如下:

```

#include <stdio.h>
int main() {
    printf("很高兴学习 C 语言\n");    //输出“很高兴学习 C 语言”,并换行
    printf("We are students\n");      //输出 We are students,并换行
    system("pause");                  //命令行提示按任意键继续
    return 0;
}
  
```

(4) 编译源文件 example1_1.c

在 VC++ 开发界面上单击“组建”(某些 VC++ 6.0 版本单击“编译”)菜单,选择“编译”命令对源文件进行编译,如果源文件没有错误,将产生目标文件,目标文件的名字是 example1_1.obj(因为源文件的名是 example1_1.c)。如果源文件有错误,编译器将提示有关错误,这时必须改正错误,然后重新编译源文件。也可以单击界面上的“编译”快捷按钮编译源文件。

(5) 链接目标文件和库函数

在 VC++ 开发界面上单击“组件”菜单,选择“组建(链接)”命令(有的版本是选择“构建”命令)对目标文件以及所用的库函数进行链接,生成可执行文件。VC++ 6.0 将链接操作称为链接工程,生成的可执行文件的名是 project1_1.exe(因为工程名是 project1_1)。也可以单击界面上的“组建(链接)”快捷按钮进行链接操作。

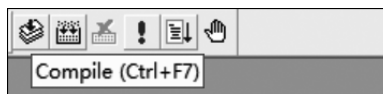


图 1.4 编译的快捷按钮

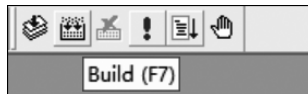


图 1.5 链接的快捷按钮

(6) 运行可执行文件

在 VC6.0++ 开发界面上单击“组建”菜单,选择“执行”命令,运行可执行文件 project1_1.exe。也可以单击界面上的“运行”快捷按钮运行可执行文件。

可执行文件的运行结果如图 1.7。

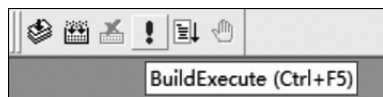


图 1.6 执行的快捷按钮

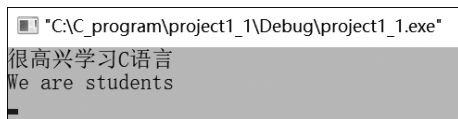


图 1.7 运行结果

注 程序运行后,要及时关闭弹出的控制台,按回车键即可关闭,通常为命令行窗口,以免影响 VC++ 6.0 继续编译其他源文件或运行其他 C 程序。

在 VC++ 6.0 开发中运行程序属于调试运行,只有脱离 VC++ 6.0 环境,运行程序才是软件意义的运行。

3. 训练小结与拓展

(1) main()函数

C 程序的基本结构就是函数,有时 C 程序可以只有一个函数,比如只有 main()函数(称为主函数)。除 main()函数外,C 程序还可以有多个其他函数,main()函数可以调用如 printf()库函数等。现在只需知道:一个 C 程序(工程)必须有且仅有一个 main()函数,操作系统从 main()函数开始执行 C 程序。按 ANSI C 标准,main()函数的格式如下:

```
int main() {
    ...
    return 0;
}
```

main()函数前面的 int 称为它的类型(要求 main()函数使用 return 返回一个整数值,通常返回 0 即可),main 后面的一对小括号表明这是一个函数,紧接着的一对大括号及其所包含的内容是它的函数体。

如果工程中没有 main()函数,将无法进行链接操作,也就无法生成可执行文件。

(2) printf()函数

printf()函数是库函数,其作用是可以输出数据到终端设备,比如输出一行文本,输出的文本中如果包含换行转义符\n,那么 printf()函数在输出文本时将\n输出为换行。为了使用 printf()函数,需要在源文件中

使用 `#include` 命令包含 `stdio.h` 函数库。

(3) 大括号的使用风格

源代码的许多地方都使用一对大括号,比如 `main()` 函数的函数体。“行尾”风格是左大括号在上一行的行尾,而右大括号独占一行。“独行”风格是左、右大括号各自独占一行。下列是“行尾”风格:

```
int main() {  
    其他代码  
}
```

下列是“独行”风格:

```
int main()  
{  
    其他代码  
}
```

(4) 注释

编译器会忽略注释内容,注释的目的是便于代码的维护和阅读,因此必须养成给代码注释编程的习惯。C 编译器支持两种格式的注释:单行注释和多行注释。

单行注释使用“`//`”表示单行注释的开始,即该行中从“`//`”开始的后续内容为注释。多行注释使用“`/*`”表示注释的开始,以“`*/`”表示注释的结束。示例如下:

```
/* 以下是 main() 方法,操作系统首先执行该方法 */  
int main() {  
    printf("你好");           //在控制台(通常是命令行)输出“你好”  
    return 0;  
}
```

(5) 关于编译

C 语言是面向过程的编程语言,简单地说,编程人员只需按照 C 语言的语法编写解决问题的过程,C 语言使用函数封装这样的过程(在第 6 章将还会详细讲述)。过程语言的源文件的特点之一是更接近人类的“自然语言”,比如,C 语言源程序中的如下函数:

```
int max(int a, int b) {  
    if(a > b)  
        return a;  
    else  
        return b;  
}
```

该函数负责计算两个整数间的最大值。使用面向过程的语言,人们只需按自己的意图编写各个函数,语言的语法更接近人们的自然语言,所以,习惯上也称过程语言是高级语言。但是,无论哪种高级语言编写的源文件,计算机中的操作系统都不能直接执行,因为计算机的操作系统只能直接识别、执行机器指令。因此,必须把源文件转换成机器指令,然后再由计算机去执行相应的机器指令。所谓机器指令是指可以被计算机操作系统直接识别、执行的一种由 0、1 组成的序列代码。每种操作系统都会形成自己独特的机器指令,比如,某种操作系统可能用 8 位序列代码 1000 1111 表示一次加法操作,以 1010 0000 表示一次减法操作,而另一种计算机可能用 8 位序列代码 1010 1010 表示一次加法操作,以 1001 0011 表示一次减法操作。因此,必须把源代码转化为操作系统能够识别的机器指令来执行。

编译器将检查源文件中的代码是否有语法错误,如果有语法错误,将提示有关错误;如果没有语法错误,编译器就会将源文件转化为二进制文件,该二进制文件被称作源文件的目标文件。目标文件的名字与

源文件的名称相同,但扩展名为.obj(某些编译器得到的目标文件的扩展名为.O)。目标文件是供链接器使用的文件(不是可执行文件),也就是说,目标文件中含有待确定的链接信息,链接器必须把这些信息替换成真正的链接代码,形成一个完整的可执行的代码。

(6) console 控制台

因为用户建立的是 Win32 Console Application,所以 C 程序在 console 控制台(通常是命令行窗口)运行。用户双击 C 程序的可执行文件,比如 project1_1.exe(不是在编程环境 VC++ 6.0 中执行),C 程序在控制台(命令行窗口)显示结果后,控制台就消失了,导致用户无法看到程序的运行结果。因此,可以在 main() 函数结束前,即 return 语句之前添加如下语句:

```
system("pause");
```

该语句的作用是,等待用户按回车键后才结束程序,即在用户没有按回车键之前,控制台始终对用户可见。

(7) 关闭与打开工程

单击 VC++ 6.0 开发界面上的“文件”菜单,选择“关闭工作空间”命令关闭当前工程。选择“打开工作空间”命令,在弹出的对话框中选择扩展名是.dsw 的工程文件,即通过打开工程文件打开之前建立的工程。

(8) 注意逗号、分号等符号

编写源文件时,初学者需要特别注意,要在英文输入法状态下输入源文件中用到的逗号、分号和小括号等符号。

1.2.3 上机实践——工程与源文件

1. 控制回车(\n)

将 example1_1.c 中的代码: printf("很高兴学习 C 语言\n"); 更改为: printf("很高兴\n学习 C 语言\n"); 观察程序的输出结果会有怎样的变化。

2. 可执行文件

如果工程 project1_1 建立在 C:\C_program 目录中,那么在 C:\C_program\project1_1 目录下的 Debug 子目录中可以看到名字为 project1_1.exe 的可执行文件,运行该文件(双击该文件)也称运行工程。可以将 project1_1.exe 可执行文件复制给其他 Window 系统平台的用户,用户双击 project1_1.exe 就可以运行这个 C 程序。

3. 多个源文件的工程

一个 C 程序(工程)中可以有多个源文件,这些源文件中包含 C 程序所需要的函数。

(1) 建立名字为 project1_2 的空工程

在 VC++ 6.0 开发界面上单击“文件”菜单,选择“新建”命令,在弹出的“新建”对话框在该对话框中选择“工程”选项卡,输入工程名 project1_2,工程的位置是 C:\C_program,如图 1.8 所示。要特别注意,在当前对话框左侧的选项列表中应选择 Win32 Console Application 选项,不可以选择“Win32 Application”。

(2) 向工程添加 add.c 源文件

在 VC++ 6.0 开发界面上单击“文件”菜单,选择“新建”命令。在弹出的“新建”对话框中选择“文件”选项卡,并在当前对话框左侧的选项列表中选择 C++ Source File,在右侧的“文件名称”文本框中输入源文件的名称 add.c(注意必须带扩展名),选中“添加到工程”单选按钮,即将 add.c 添加到 project1_2 工程中,然后在 add.c 的编辑区输入如图 1.9 所示的内容。

(3) 向工程添加 example1_2.c 源文件

在 VC++ 6.0 开发界面上单击“文件”菜单,选择“新建”命令,在弹出的“新建”对话框中选择“文件”选项卡,并在当前对话框左侧的选项列表中选择 C++ Source File,在右侧的“文件名称”文本框中输入源文件的名称 example1_2.c(注意必须带扩展名),选中“添加到工程”单选按钮,即将 example1_2.c 添加到 project1_2



图 1.8 建立名字是 project1_2 的工程



图 1.9 编辑源文件 add.c

工程中,然后在 example1_2.c 的编辑区输入如图 1.10 所示的内容。

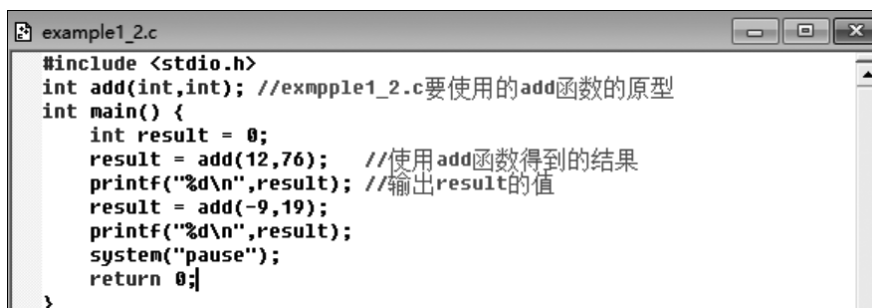


图 1.10 编辑源文件 example1_2.c

(4) 分别编译 add.c 和 example1_2.c

让 add.c 和 example1_2.c 的编辑区处于活动状态,然后分别编译它们。如果没有语法错误,将产生名字为 add.obj 和 example1_2.obj 的目标文件;如果有语法错误,编译器将提示有关错误,这时必须修改错误再重新编译。

(5) 链接

单击“组建”菜单,选择“组建(链接)”命令,对目标文件和所用的库函数进行链接,生成可执行文件。

(6) 运行

链接成功后,在 VC++ 6.0 开发界面上单击“组建”菜单,选择“执行”命令,运行可执行文件 project1_2.

exe(也可以到 C:\C_program\project1_2\Debug 目录下双击 project1_2.exe 运行 C 程序),运行结果如图 1.11 所示。

创建具有多个源文件的工程的步骤如图 1.12 所示。

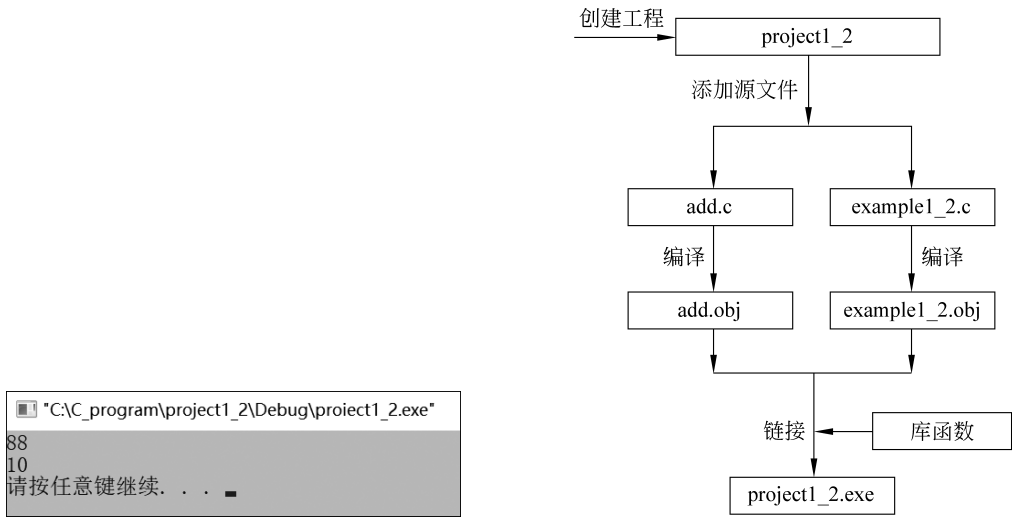


图 1.11 程序运行结果

图 1.12 有多个源文件的工程的创建步骤

注 一个工程里,需要有一个源文件含有 main()函数(相关细节将在第 6 章中讨论)。

小结

- C 语言不仅是学习其他计算机课程的基础,而且本身也是很强大的一门编程语言,有着自己独特的语言特性,尤其在嵌入式领域有着很重要的应用地位。
- 本章中读者只需大致了解 C 程序的基本结构,重点掌握开发 C 程序的基本步骤:编写源文件、编译源文件、链接目标文件、运行可执行文件。
- 学会配置 VC++ 6.0,熟练使用 VC++ 6.0 创建工程、向工程添加源文件,以及使用 VC++ 6.0 进行编译、链接等操作。

习题 1

1. 判断题(题目叙述正确的,在后面的括号里打√,否则打×)

- (1) C 语言的发明人是 Dennis Ritchie 和 Ken Thompson。 ()
- (2) C 语言是非常适合用于编写操作硬件设备的编程语言。 ()
- (3) C 语言的源文件是扩展名为.c 的文本文件。 ()
- (4) 操作系统可以直接运行扩展名是.c 的源文件。 ()
- (5) 编译器将 C 语言的源文件编译成扩展名是.exe 的可执行文件。 ()
- (6) 编译器将 C 语言的源文件编译成扩展名是.obj 的目标文件。 ()
- (7) 链接器将目标文件、库函数进行链接,并加入操作系统的启动代码产生一个可执行文件。 ()
- (8) 一个 C 程序(工程)必须有且仅有一个 main()函数,操作系统从 main()函数开始执行 C 程序。 ()
- (9) 一个 C 程序(工程)中可以有多个源文件。 ()
- (10) 编译器得到的目标文件是可执行文件。 ()

2. 单项选择题

- (1) 下列选项中叙述正确的是()。

- A. 一个 C 程序(工程)由若干个函数构成,这些函数必须在同一个源文件中
B. 一个 C 程序(工程)由若干个函数构成,这些函数可以在一个源文件中,也可以分布在若干个源文件中,其中必须有一个源文件含有 main()函数
C. 一个 C 程序(工程)的源文件必须含有 main()函数
D. 一个 C 程序(工程)中可以有多个 main()函数
- (2) 编译器编译源文件得到的目标文件的扩展名是()。
- A. .obj B. .cpp C. .exe D. .dsw
- (3) 连接器得到的可执行文件的扩展名是()。
- A. .obj B. .dsp C. .bat D. .exe
- (4) 下列选项中叙述错误的是()。
- A. 一个 C 程序(工程)中可以有多个源文件
B. C 语言的发明人是 Bill Gates
C. 编译器将 C 语言的源文件编译成扩展名是.obj 的目标文件
D. C 语言的源文件是扩展名为.c 的文本文件
- (5) 下列选项中叙述正确的是()。
- A. 操作系统可以直接运行扩展名是.c 的文件
B. 操作系统可以直接运行扩展名是.obj 的文件
C. 操作系统可以直接运行扩展名是.exe 的文件
D. 操作系统可以直接运行扩展名是.txt 的文件

3. 阅读程序

- (1) 写出下列 main()函数中【代码 1】、【代码 2】的输出结果。

```
#include <stdio.h>
int main() {
    printf("how are you");          //【代码 1】
    printf("你好");                 //【代码 2】
    system("pause");
    return 0;
}
```

- (2) 写出下列 main()函数中【代码 1】、【代码 2】的输出结果。

```
#include <stdio.h>
int sub(int x,int y){
    return x-y;
}
int main(){
    int result =0;
    result =sub(100,2);
    printf("%d",result);           //【代码 1】
    result =sub(7,2);
    printf("%d",result);           //【代码 2】
    system("pause");
    return 0;
}
```