

C

语言编程初学者指南

C Programming

[美] Keith Davenport Michael Vine 著
李强 译



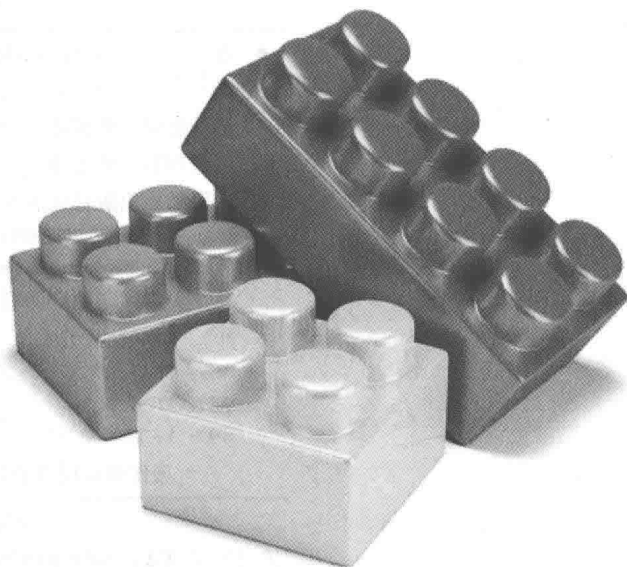
— C —

语言编程初学者指南

C Programming

[美] Keith Davenport Michael Vine 著

李强 译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

C语言编程初学者指南 / (美) 达文波特
(Keith Davenport), (美) 维恩 (Michael Vine) 著 ;
李强译. — 北京 : 人民邮电出版社, 2017. 5
ISBN 978-7-115-45129-3

I. ①C… II. ①达… ②维… ③李… III. ①C语言—
程序设计—指南 IV. ①TP312. 8-62

中国版本图书馆CIP数据核字(2017)第058586号

版权声明

C Programming for the Absolute Beginner, Third Edition

Keith Davenport, Michael Vine

Copyright © 2015 Cengage Learning PTR.

Original edition published by Cengage Learning. All Rights reserved.

本书原版由圣智学习出版公司出版。版权所有, 盗印必究。

Posts & Telecom Press is authorized by Cengage Learning to publish and distribute exclusively this simplified Chinese edition.

This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan).

Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书中文简体字翻译版由圣智学习出版公司授权人民邮电出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可, 不得以任何方式复制或发行本书的任何部分。

978-1-305-27376-4

Cengage Learning Asia Pte. Ltd.

151 Lorong Chuan, #02-08 New Tech Park, Singapore 556741

本书封面贴有 Cengage Learning 防伪标签, 无标签者不得销售。

-
- ◆ 著 [美] Keith Davenport Michael Vine
译 李 强
责任编辑 陈冀康
责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京天宇星印刷厂印刷
 - ◆ 开本: 800×1000 1/16
印张: 17
字数: 321 千字 2017 年 5 月第 1 版
印数: 1-3 000 册 2017 年 5 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2017-1457 号
-

定价: 49.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广字第 8052 号

内容提要

本书是专门为 C 语言编程的初学者编写的入门读物。全书共 12 章，通过清晰的讲解和详细的示例，介绍了 C 编程基础知识，包括基本数据类型、条件、循环结构、结构化程序设计、数组、指针、字符串、数据结构、动态内存分配、文件输入和输出与 C 预处理器指令等话题。本书中的每一章都包含了本章小结和编程挑战，而且给出了一个完整的程序，从而使得读者应用一章中所学的基本概念构建一个容易编写的应用。

本书适合于想要通过 C 开始学习编程的读者、已经具备其他高级语言的编程经验，想要继续学习 C 的爱好者和对 C 编程感兴趣的初学爱好者阅读参考。

前言

C 语言是一种强大的基于过程的编程语言，它于 1972 年由 Dennis Ritchie 在贝尔实验室发明。C 语言最初是开发来用于 UNIX 平台的，但却已经扩展到很多其他的系统和应用中。C 语言还影响了很多其他的编程语言，例如 C++ 和 Java。

编程初学者，特别是那些进入计算机科学和工程专业学习的人，需要构建有关操作系统、硬件和应用程序开发概念的牢固基础知识。很多的学院教授学生学习如何用 C 编程，从而使他们能够学习高级概念以及在 C 的基础上建立起来的其他语言。

学习 C 语言的很多学生也承认，它不是一种很容易学习的编程语言，但是，从专业人士的角度来看，本书清晰的讲解、完备的示例和图片，使得学习 C 语言变得容易而有趣。本书中的每一章都包含了本章小结和编程挑战，而且还给出了一个完整的程序，从而使得读者可以应用一章中所学的基本概念来构建一个容易编写的应用。

要完整地学习本书，你需要在计算机中安装 gcc（具体介绍参见本书第 1 章）或者 TCC（参见附录 D）这样的一个 C 编译器，并且至少要有一个文本编辑器。在 UNIX 或者类似的环境下，你可能会使用 vi、Vim 或 Emacs。在 Microsoft Windows 下，你可以使用 Notepad 或者任何其他的纯文本编辑器。

本书主要内容

如果要学习如何用计算机编程，那么，你必须在技能上有所提升。如果你根本没有编写过程序，那么，你会发现按顺序阅读完本书的各章是很容易的。但是，编程不是通过阅读就能够学习的技能。你必须通过编写程序来学习。学习本书的内容可以使得学习编程的过程相当轻松而有趣。

每章一开始，都会简单介绍一章所要讲述的基本概念。在一章之中，你会接触到一系列的编程概念，还有一些小的程序来说明本章的每一个主要知识点。在每一章的末尾，我们综合应用这些概念来编写一个完整的程序。所有的程序都比较短，以便你能够自行录入它们（这是近距离接触代码的好办法），当然，也可以通过出版社的网站来获取这些程序。在每一章的

最后都有一个小结，列出了所学的关键概念。通过这个小结，你可以再次巩固并记住这些重要的概念。在小结之后，每章给出了编程挑战，可以帮助你学习和应用一章中所学到的概念。

在本书中，有如下一些显著的板块：

技巧

这部分包含了值得特别留意的信息。

陷阱

这些警告是你很容易犯的错误，或者会遇到的问题。

提示

这部分针对相关的主题给出额外的见解或信息。

主题边栏

当你学习本书中的概念的时候，这部分介绍在更高的编程层级或者在现实世界中如何使用这些概念。

本书目标读者

本书是专门为初学者编写的。本书不适合那些想要学习 C++ 面向对象编程或链表这样的高级 C 语言数据结构的、有经验的 C 语言程序员。

本书适合以下人群阅读：

- 通过 C 语言开始学习编程的大学或高中学生；
- 有其他高级语言（Visual Basic、VBA、HTML 或 JavaScript）的编程经验的程序员，但想要继续学习 C 语言；
- 想要自学 C 语言的编程爱好者；
- 对学习 C++、C# 和 Java 感兴趣，但有人建议你先学习 C 语言；
- 一直想要学习如何编程，并且选择 C 语言作为自己的第一门语言。

如果你属于以上的某一类人，那么，你可以通过阅读本书来学习 C 编程。特别是，我将

使用非图形化的文本编辑器和 ANSI C 编译器 gcc 来教你学习基本的编程概念，例如变量、条件、循环、数组、结构、文件 I/O 等，这些概念在学习任何编程语言的时候都很有用。当然，你还将学习一些适用于 C 的特定主题，例如指针和动态内存分配，这些功能使得 C 语言很独特也很强大。

配套网站下载

在本书的配套网站上，可以下载本书的示例代码 www.cengageptr.com/downloads 或 www.epubit.com.cn。

致谢

为了让一本书从字处理文件中到你的书架上，各种有天赋的人必须在精确性、时间和技能方面扮演好自己的角色。当这些参与者也很专业、好心且投入的时候，对于作者和读者来说，都会有一种很好的体验。我要感谢 **Mitzi Koontz**、**Karen Gill**、**Michael Vine** 与我们分享他们的聪明才智，并且允许我参与这本书的新版的编写工作。很骄傲能够与你们每一个人一起工作。还要感谢本版的校对者 **Sam Garvey**，感谢你不仅尽职尽责，而且提供了一些不错的勘误。

作者简介

Keith Davenport 是这个行业里的多面手，从事出版、媒体和技术工作长达 20 年时间。在这个产业之外，他还是一位电影编剧和制片人，并且充当人们所需要的各种社会工作者。

Michael Vine 是一位资深的 IT 专业人士，专门从事数据仓储和商务智能工作。除了在公司工作，他还在大学和学院教授计算机科学课程，并且他还是很多软件编程图书的作者。

目 录

| | |
|--------------------------------------|-------------------------------------|
| 第 1 章 C 编程入门.....1 | 1.9 本章小结.....22 |
| 1.1 安装和配置 Cygwin 环境.....2 | 1.10 编程挑战.....23 |
| 1.2 认识 main()函数.....4 | 第 2 章 基本数据类型.....24 |
| 1.3 使用注释.....7 | 2.1 内存概念简介.....24 |
| 1.4 理解关键字.....8 | 2.2 理解数据类型.....25 |
| 1.5 使用程序语句.....9 | 2.2.1 使用整数.....26 |
| 1.5.1 转义序列\n.....11 | 2.2.2 使用浮点数.....26 |
| 1.5.2 转义序列\t.....12 | 2.2.3 使用字符.....27 |
| 1.5.3 转义序列\r.....12 | 2.3 初始化变量和赋值运算符.....27 |
| 1.5.4 转义序列\\.....13 | 2.4 打印变量的内容.....28 |
| 1.5.5 转义序列".....13 | 2.5 使用转换修饰符.....29 |
| 1.5.6 转义序列'.....13 | 2.5.1 用 printf()显示整数数据 类型.....30 |
| 1.6 使用指令.....14 | 2.5.2 用 printf()显示浮点数据 类型.....30 |
| 1.7 创建并运行第一个 C 程序.....15 | 2.5.3 用 printf()显示字符数据 类型.....31 |
| 1.8 调试 C 程序.....16 | 2.6 理解常量.....32 |
| 1.8.1 常见错误之 1: 漏掉程序 语句块标识符.....19 | 2.7 使用编程惯例和风格.....33 |
| 1.8.2 常见错误之 2: 漏掉语句 终结符.....20 | 2.7.1 使用空白.....33 |
| 1.8.3 常见错误之 3: 无效的预 处理器指令.....20 | 2.7.2 变量命名惯例.....34 |
| 1.8.4 常见错误之 4: 无效的转义 序列.....21 | 2.7.3 用前缀表示数据类型.....35 |
| 1.8.5 常见错误之 5: 无效的注释 语句块.....21 | 2.7.4 正确地使用大写字母和 小写字母.....35 |
| | 2.7.5 给变量一个有意义的名称.....36 |

| | | | |
|-------------------------------|-----------|-------------------------------|-----------|
| 2.7.6 使用 scanf()函数 | 36 | 3.9 本章小结 | 67 |
| 2.8 用 C 做算术运算 | 38 | 3.10 编程挑战 | 68 |
| 2.9 理解运算符优先级 | 40 | 第 4 章 循环结构 | 69 |
| 2.10 本程序:Shop Profit | 40 | 4.1 循环的伪代码 | 69 |
| 2.11 本章小结 | 41 | 4.2 循环的流程图 | 71 |
| 2.12 编程挑战 | 42 | 4.3 其他运算符 | 75 |
| 第 3 章 条件 | 43 | 4.3.1 ++运算符 | 75 |
| 3.1 用于条件的算法 | 43 | 4.3.2 --运算符 | 78 |
| 3.1.1 表达式和条件运算符 | 44 | 4.3.3 +=运算符 | 79 |
| 3.1.2 伪代码 | 45 | 4.3.4 -=运算符 | 81 |
| 3.1.3 流程图 | 46 | 4.4 while 循环 | 81 |
| 3.2 简单的 if 结构 | 49 | 4.5 do while 循环 | 84 |
| 3.3 嵌套的 if 结构 | 51 | 4.6 for 循环 | 85 |
| 3.4 布尔代数简介 | 53 | 4.7 break 和 continue 语句 | 87 |
| 3.4.1 and 运算符 | 54 | 4.8 系统调用 | 89 |
| 3.4.2 or 运算符 | 54 | 4.9 本程序: Concentration | 90 |
| 3.4.3 not 运算符 | 55 | 4.10 本章小结 | 92 |
| 3.4.4 运算优先级 | 55 | 4.11 编程挑战 | 92 |
| 3.4.5 用布尔运算符构建复合 条件 | 56 | 第 5 章 结构化程序设计 | 94 |
| 3.5 复合 if 结构和输入验证 | 57 | 5.1 结构化程序设计简介 | 94 |
| 3.5.1 &&运算符 | 57 | 5.1.1 自顶向下的设计 | 95 |
| 3.5.2 运算符 | 58 | 5.1.2 代码可复用性 | 97 |
| 3.5.3 检查大写字母和小写 字母 | 58 | 5.1.3 信息隐藏 | 98 |
| 3.5.4 检查值的范围 | 59 | 5.2 函数原型 | 99 |
| 3.5.5 isdigit()函数 | 60 | 5.3 函数定义 | 101 |
| 3.6 switch 结构 | 61 | 5.4 函数调用 | 103 |
| 3.7 随机数 | 64 | 5.5 变量作用域 | 105 |
| 3.8 本程序: Fortune Cookie | 66 | 5.5.1 局部作用域 | 106 |
| | | 5.5.2 全局作用域 | 107 |

| | | | | | |
|--------------|------------------|------------|--------------|----------------------------|------------|
| 5.6 | 本程序: Trivia | 108 | 8.2 | 读取和打印字符串 | 154 |
| 5.7 | 本章小结 | 111 | 8.3 | 字符串数组 | 155 |
| 5.8 | 编程挑战 | 112 | 8.4 | 将字符串转换为数字 | 157 |
| 第 6 章 | 数组 | 113 | 8.5 | 操作字符串 | 159 |
| 6.1 | 数组简介 | 113 | 8.5.1 | strlen()函数 | 159 |
| 6.2 | 一维数组 | 114 | 8.5.2 | tolower()和 toupper() 函数 | 160 |
| 6.2.1 | 创建一维数组 | 115 | 8.5.3 | strcpy()函数 | 161 |
| 6.2.2 | 初始化一维数组 | 115 | 8.5.4 | strcat()函数 | 162 |
| 6.2.3 | 搜索一维数组 | 119 | 8.6 | 分析字符串 | 163 |
| 6.3 | 二维数组 | 121 | 8.6.1 | strcmp()函数 | 163 |
| 6.3.1 | 初始化二维数组 | 122 | 8.6.2 | strstr()函数 | 165 |
| 6.3.2 | 搜索二维数组 | 123 | 8.7 | 本程序: Word Find | 166 |
| 6.4 | 本程序: Tic-Tac-Toe | 125 | 8.8 | 本章小结 | 168 |
| 6.5 | 本章小结 | 129 | 8.9 | 编程挑战 | 168 |
| 6.6 | 编程挑战 | 129 | 第 9 章 | 数据结构简介 | 170 |
| 第 7 章 | 指针 | 131 | 9.1 | 结构 | 170 |
| 7.1 | 指针基础 | 131 | 9.1.1 | struct 关键字 | 171 |
| 7.1.1 | 声明和初始化指针变量 | 132 | 9.1.2 | typedef 关键字 | 172 |
| 7.1.2 | 打印指针变量的内容 | 134 | 9.1.3 | 结构的数组 | 174 |
| 7.2 | 函数和指针 | 136 | 9.2 | 结构和函数 | 175 |
| 7.3 | 给函数传递数组 | 139 | 9.2.1 | 按值传递结构 | 176 |
| 7.4 | const 限定符 | 143 | 9.2.2 | 按引用传递结构 | 177 |
| 7.5 | 本程序: Cryptogram | 145 | 9.2.3 | 传递结构的数组 | 179 |
| 7.5.1 | 加密简介 | 145 | 9.3 | 联合体 | 180 |
| 7.5.2 | 编写 Cryptogram 程序 | 147 | 9.4 | 强制类型转换 | 182 |
| 7.6 | 本章小结 | 149 | 9.5 | 本程序: Card Shuffle | 183 |
| 7.7 | 编程挑战 | 149 | 9.6 | 本章小结 | 186 |
| 第 8 章 | 字符串 | 151 | 9.7 | 挑战 | 187 |
| 8.1 | 字符串简介 | 151 | | | |

| | |
|-------------------------------------|--|
| 第 10 章 动态内存分配188 | |
| 10.1 动态内存概念.....188 | |
| 栈和堆.....189 | |
| 10.2 sizeof 运算符.....189 | |
| 10.3 malloc()函数.....191 | |
| 10.3.1 使用 malloc()管理 字符串.....193 | |
| 10.3.2 释放内存.....194 | |
| 10.3.3 操作内存段.....194 | |
| 10.4 calloc()和 realloc()函数.....195 | |
| 10.5 本程序: Math Quiz.....198 | |
| 10.6 本章小结.....200 | |
| 10.7 编程挑战.....201 | |
| 第 11 章 文件输入和输出203 | |
| 11.1 数据文件简介.....203 | |
| 11.2 位和字节.....204 | |
| 字段、记录和文件.....204 | |
| 11.3 文件流.....205 | |
| 11.3.1 打开和关闭文件.....206 | |
| 11.3.2 读取数据.....208 | |
| 11.3.3 写数据.....211 | |
| 11.3.4 添加数据.....213 | |
| 11.4 goto 语句和错误处理.....215 | |
| 11.5 本程序: Character Roster.....217 | |
| 11.6 本章小结.....219 | |
| 11.7 编程挑战.....220 | |
| 第 12 章 C 预处理器221 | |
| 12.1 理解 C 预处理器.....221 | |
| 12.1.1 符号常量.....222 | |
| 12.1.2 创建和使用宏.....224 | |
| 12.2 编译较大的程序.....226 | |
| 12.2.1 头文件.....227 | |
| 12.2.2 函数定义文件.....227 | |
| 12.2.3 main()函数.....228 | |
| 12.3 本程序: Function Wizard.....229 | |
| 12.3.1 ch12_calculate.h.....230 | |
| 12.3.2 ch12_calculate.c.....230 | |
| 12.3.3 ch12_main.c.....231 | |
| 12.4 本章小结.....232 | |
| 12.5 编程挑战.....232 | |
| 12.6 如何继续学习.....233 | |
| 附录 A 常用 UNIX 命令234 | |
| 附录 B Vim 快速参考236 | |
| 附录 C nano 快速指南238 | |
| 附录 D TCC 快速参考241 | |
| D.1 在 Windows 下安装和配置 TCC.....241 | |
| D.1.1 下载 TCC.....241 | |
| D.1.2 安装 TCC.....242 | |
| D.1.3 配置 TCC.....243 | |
| D.2 编写、编译并执行代码.....244 | |
| D.2.1 编写和编辑源代码.....244 | |
| D.2.2 编译和执行源代码.....245 | |
| 附录 E ASCII 字符代码247 | |
| 附录 F 常用 C 库函数251 | |

第 1 章

C 编程入门

欢迎阅读本书。C 编程语言是培养你的编程职业技能和爱好的一门优秀的基础语言。不管你是计算机专业的学生、自学成才的程序员，或者是一名资深的软件工程师，学习 C 语言都能够给你丰富的概念知识并培养实践技能，从而很好地帮助你理解其他的计算机相关主题（包括操作系统概念、内存管理和其他高级的编程语言）。

在整个本书中，我将引导你学习一系列的示例，这些示例设计来讲解 C 编程的基础知识。我假设读者没有 C 编程的经验，也不了解计算机科学的基本概念。阅读本书不需要任何的经验（包括高级的数学知识），但是我假设你至少了解 Microsoft 或基于 UNIX（或基于 Linux）的操作系统的基础知识，并且知道如何使用一款文本编辑器。

如果你已经有了关于其他编程语言的一些编程知识，例如 Java、Visual Basic、Ruby、PowerBuilder 或 Python，你仍然能够从本书中获益。我期望你阅读了本书之后，会认为这是一本有用的 C 编程参考书。

本章包括以下内容：

- 安装和配置 Cygwin 环境；
- 理解 main() 函数；
- 使用注释；
- 理解关键字；
- 使用程序语句；
- 使用指令；
- 创建和运行第一个 C 程序；
- 调试 C 程序。

1.1 安装和配置 Cygwin 环境

要学习 C 编程，所需的所有东西只不过是一台计算机、一款文本编辑器、C 库和一个 C 编译器。在整个本书中，我们使用一款简单的文本编辑器来编写 C 程序。和很多高级的编程语言（如 Visual Basic 或 C#）不同，C 语言并不需要一个高级的图形化用户界面（graphical user interface, GUI）。实际上，一个复杂的、功能丰富的界面，可能会让想要学习编程的初学者分心。他们很容易去关注漂亮界面的那些细枝末节，而不是关注诸如变量和循环这样的基本的编程概念，而这些概念才是编程初学者应该重点关注的问题。

提示

你知道计算机和文本编辑器是什么，但是，不知道 C 库是什么。C 编译器又是什么呢？本章后面会介绍这两个概念，但是，简而言之，库是预先编写好的代码，可以用来执行某些标准函数，例如，从用户那里获取输入。而诸如 gcc 这样的编译器则是一个程序，它们接受你输入到一个文本文件中的代码，将其转换为计算机可以使用的格式并创建可执行的程序。

有几种免费的 C 编译器和文本编辑器可供使用。当然，还有很多商业化的版本。如果你已经有了必需的编程工具，可以跳过这个小节。但是如果你还没有，Cygwin 已经开发了一款用于 Windows 的、简单的、健壮的、类似于 Linux 的环境，其中包含了很多免费的软件包，例如，一款叫做 gcc 的 C 编译器、文本编辑器以及其他常用的工具。可以从 www.cygwin.com 下载 Cygwin 的免费软件包。

Cygwin 的安装过程很简单，但是，如果你遇到问题，可以访问位于 <http://cygwin.com/cygwin-ug-net/cygwin-ug-net.html> 的、在线的用户指南。一旦安装了 Cygwin，可以通过 UNIX shell 或 Windows 命令提示符来使用众多的基于 UNIX 的工具。

要安装该软件，最少需要 400MB 的可用硬盘空间（根据所选的组件，所需空间可能会略大或略小）。要安装 Cygwin 及其相关的组件，从 <http://cygwin.com/install.html> 下载安装文件。按照安装界面提示进行，直到打开 Cygwin Setup—Select Packages 窗口。一旦打开了该窗口，可以选择你想要安装的组件。默认选取的组件再加上“gcc-core: C Compiler”安装组件，就足够你运行本书中的所有代码了。然而，注意 gcc-core: C Compiler 组件默认是没有选中的。要选中这个组件，向下滚动，直到找到 gcc-core: C Compiler component。点击 Skip 按钮来选择安装组件。

技巧

如果你想要使用 Cygwin 中的文本编辑器，找到 Editors 部分并且选择 Nano（本书附录 C 会介绍 Nano）或者 Vim（本书附录 B 会介绍 Nano），或者两个编辑器都选中。也可以在 Cygwin 的外部，使用诸如 Notepad 这样的文本编辑器来编写代码。

注意，安装程序可能会提示你安装其他的软件包以解决依赖性。如果是这样的话，允许安装程序包含所需的那些软件包。

在成功安装了 Cygwin 环境之后，你将能够通过 UNIX shell 访问一个模拟的 UNIX (Linux) 操作系统环境。要启动 UNIX shell，直接在桌面或者通过开始菜单的程序组找到 Cygwin 的快捷方式。

在启动了程序之后，Cygwin UNIX 如图 1.1 所示。

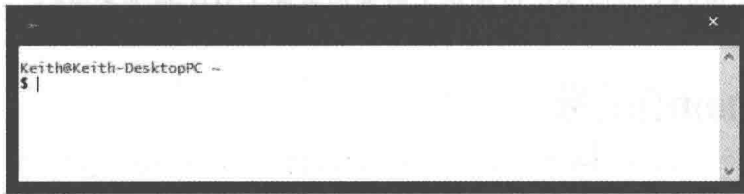


图 1.1 启动 Cygwin UNIX shell

注意图 1.1 中用于 UNIX 命令提示符的语法，你的语法将会不同。

```
Keith@Keith-DesktopPC ~  
$
```

第 1 行显示了是哪一个用户登录到了 UNIX shell（就是登录到你所使用的计算机的用户，在我的计算机上，用户是 Keith）以及计算机的名称（Keith-DesktopPC 正是我的计算机的独特的名字）。下一行以一个美元符号 (\$) 开头。这是 UNIX 命令提示符，我们在这里输入并执行 UNIX 命令。

根据你的 Cygwin 的具体安装（Cygwin 版本）和配置（所选的组件）的不同，你可能需要将 Cygwin 的 bin 目录添加到系统的 PATH 环境变量中。如果你安装了 32 位的 Cygwin，使用如下命令：

```
c:\cygwin\bin
```

如果你安装了 64 位的 Cygwin，使用如下命令：

```
c:\cygwin64\bin
```

技巧

为了简单起见，我假设你安装了 32 位的 Cygwin。如果你安装了 64 位的 Cygwin，也没

问题，在使用本书所引用的文件路径的时候，用 `c:\cygwin64` 代替 `c:\cygwin` 就好了。

Cygwin 和其他的程序都使用 `PATH` 环境变量来找到要运行的可执行文件。如果你使用基于 Microsoft 的操作系统，可以有几种方法来编辑 `PATH` 变量。一种方法是，通过开始菜单，在运行对话框中，输入关键字 `cmd`，从而打开一个基于 Microsoft 的命令 shell（DOS 窗口）。在 `c:\`提示符之后，输入如下命令：

```
set PATH=%PATH%;c:\cygwin\bin
```

这条命令把 `c:\cygwin\bin` 添加到了 `PATH` 变量的末尾而不会覆盖它。要验证这条命令是否执行成功，直接在同样的基于 Microsoft 的命令 shell 窗口输入关键字 `PATH`。注意，在 `PATH` 的值中，每一个不同的目录结构都是用一个分号隔开的。如果需要，请查阅你的系统的文档来了解关于环境变量的更多信息，特别是了解如何更新 `PATH` 系统变量。

1.2 认识 `main()` 函数

本小节首先介绍每个 C 程序开始的内容，也就是 `main()` 函数。首先，我想要用打比方的方式说明什么是函数。从编程的角度来讲，函数使你能够将逻辑上的一系列的动作（或程序语句）组织到一个名字之下。例如，假设你想要创建一个名为 `bakeCake` 的函数。

烘焙蛋糕的算法（过程）如下所示：

在用于搅拌的碗中，搅拌湿的成分（水、奶油、蛋汁等）；

加入干的成分（面粉等）；

将面糊铺满烤盘；

在烤箱中，用 350 度的温度烘焙 30 分钟。

注意，给你的函数起一个富有描述性的名称，将有助于任何人读懂你的代码，即便你自己在以后某个日子里忘记了，也可以很容易读懂代码，从而明白这个函数要完成什么任务。

函数通常不是静态的，这意味着，它们是有生命的、会呼吸的实体，还是打个比方吧——它们会接受一些信息并返回一些信息。因此，`bakeCake` 函数会接受要烘焙的成分的一个列表（称为参数），并且返回一个完成的蛋糕（称为值）。

`main()` 函数和任何其他的函数一样，因为它也将动作组织到一起，并且能够接受参数（信息）并返回值（也是信息）。与其他函数的不同之处在于，`main()` 函数是向操作系统返回值，