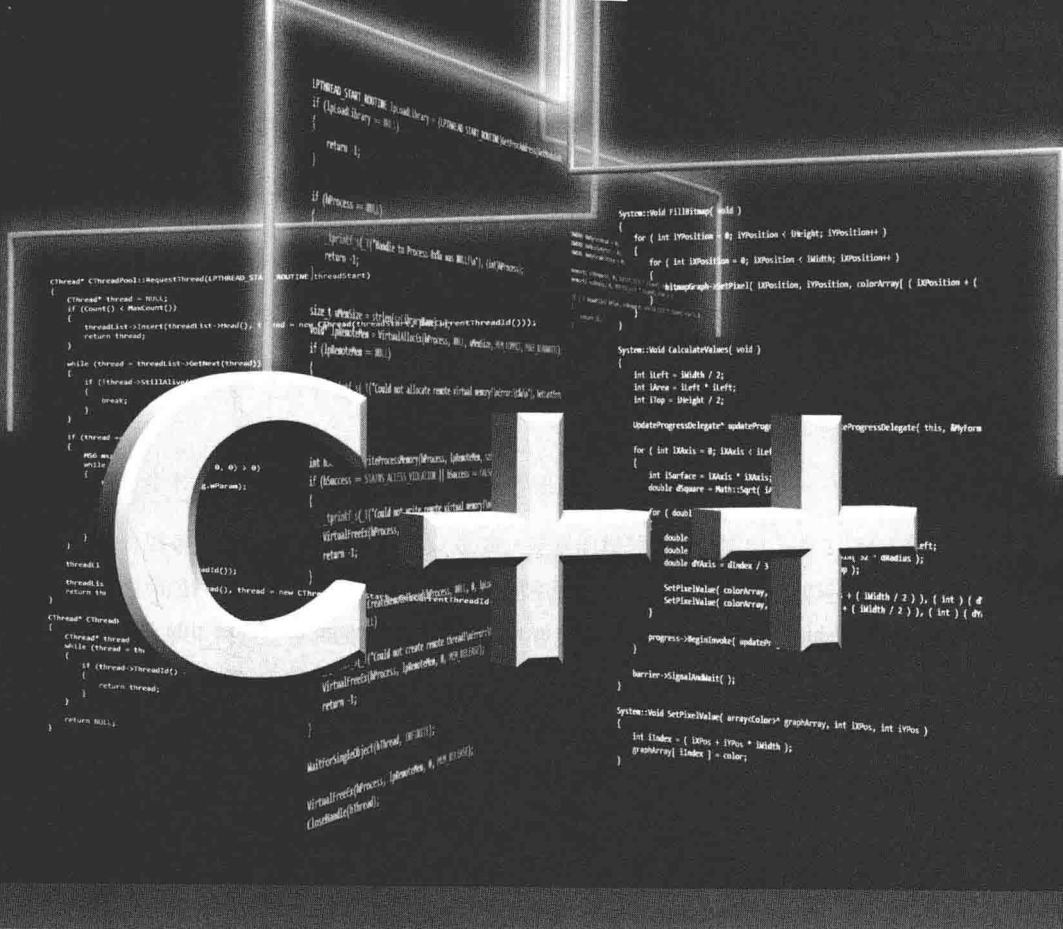


# C++

# 多线程编程实战

## C++ Multithreading Cookbook

[ 黑山共和国 ] Miloš Ljumović 著  
姜佑 译



# C++ 多线程编程实战

[ 黑山共和国 ] Miloš Ljumović 著  
姜佑 译

人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

C++多线程编程实战 / (黑) 留莫维奇著 ; 姜佑译

— 北京 : 人民邮电出版社, 2016.5

ISBN 978-7-115-41366-6

I. ①C… II. ①留… ②姜… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第085851号

## 版 权 声 明

Copyright © Packt Publishing 2014. First published in the English language under the title C++ Multithreading Cookbook.

All Rights Reserved.

本书由英国 Packt Publishing 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

- 
- ◆ 著 [黑山共和国] Miloš Ljumović
  - 译 姜 佑
  - 责任编辑 傅道坤
  - 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>
  - 三河市海波印务有限公司印刷
  
  - ◆ 开本: 800×1000 1/16
  - 印张: 20
  - 字数: 538 千字 2016 年 5 月第 1 版
  - 印数: 1-2 500 册 2016 年 5 月河北第 1 次印刷
- 著作权合同登记号 图字: 01-2015-0749 号
- 

定价: 59.00 元

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

反盗版热线: (010) 81055315

# 内容提要

本书是一本实践为主、通俗易懂的 Windows 多线程编程指导。本书使用 C++本地调用，让读者能快速高效地进行并发编程。

全书共 8 章。第 1 章介绍了 C++编程语言的概念和特性。

第 2~5 章介绍了进程、线程、同步、并发的相关知识。其中，第 2 章介绍进程和线程的基本概念，详细介绍了进程和线程对象。第 3 章讲解线程管理方面的知识，以及进程和线程背后的逻辑，简要介绍了线程同步、同步对象和同步技术。第 4 章重点介绍了消息传递技术、窗口处理器、消息队列和管道通信。第 5 章介绍了线程同步和并发操作，讲解了并行、优先级、分发器对象和调度技术，解释了同步对象（如互斥量、信号量、事件和临界区）。

第 6 章介绍 .NET 框架中的线程，概述了 C++/CLI .NET 线程对象。简要介绍了托管方法、.NET 同步要素、.NET 线程安全、基于事件的异步模式和 BackgroundWorker 对象，以及其他主题。

第 7~8 章为水平较高的读者准备了一些高级知识，概述了并发设计和高级线程管理。其中，第 7 章讲解理解并发代码设计，涵盖了诸如性能因素、正确性问题、活跃性问题的特性。第 8 章讲解高级线程管理，重点介绍更高级的线程管理知识。详细介绍了线程池的抽象、定制分发对象，以及死锁的解决方案。

附录涵盖了 MySQL Connector C 和 WinDDK 的具体安装步骤，介绍了如何为驱动程序编译和 OpenMP 编译设置 Visual Studio。另外，还介绍了 DebugView 应用程序的安装步骤，并演示了它的使用步骤。

本书主要面向中高级读者，可作为用 C++进行 Windows 多线程编程的参考读物。本书介绍的同步概念非常基础，因此也可作为对这方面技术感兴趣的读者和开发人员的参考书籍。

## 作者简介

Miloš Ljumović 于 7 月 26 日出生在欧洲黑山共和国的首都波德戈里察，在那里度过了小学和中学的时光，还到音乐学校学习了吉他。随后在黑山大学自然科学和数学学院进修了计算机科学。他对计算机有浓厚兴趣，主修操作系统并获得了硕士学位。2009 年 12 月，Miloš 和他的朋友 Danijel 一起成立了自己的公司，并作为一名程序员和高水平的团队一起致力于提供高技术含量的 IT 解决方案。不久，许多资深的开发者加入了他们，合作开发了许多应用程序和系统软件、Web 应用程序和数据库系统。他的客户不仅包括黑山政府，还涉及一些大型的国有企业。他开发了一个新的金融系统 MeNet 以及一些与图片和其他数字媒体类型相关的视频识别软件。除此之外，他还开发了许多网站和其他网络应用程序。他的客户数量众多，不胜枚举。

Miloš 作为国际顾问在美国一家大型的互联网电子商务贸易和数据采集公司工作了几个月。随后于 2014 年 7 月创立了一家新公司：EXPERT.ITS.ME。除了开发软件，他还为 IT 行业的小型企业提供咨询服务，鼓励并帮助他们在处理好企业管理问题的同时，把企业做大做强。另外，Miloš 还是黑山国家委员会（ICT）成员和门萨俱乐部成员。他热爱编程，擅长 C/C++/C# 语言，精通 HTML、PHP、TSQL 等，梦想能开发出自己的操作系统。

在业余时间里，Miloš 喜欢打网球、潜水、狩猎和下象棋。他喜欢和自己的团队进行头脑风暴，想出一些在 IT 领域和计算机科学领域新鲜、时尚的好点子。他紧跟 IT 的发展步伐，不断学习新知识、解决新问题。尤其喜欢通过专门课堂和课程教授计算机科学和数学专业的学生，将他们塑造为合格的程序员，帮助他们发现科学之美。想了解更多他的兴趣爱好和近况，请浏览他的公司网站（<http://expert.its.me>）或个人网站（<http://milos.expert.its.me>），也可以通过 [milos@expert.its.me](mailto:milos@expert.its.me) 与他联系。

# 致 谢

撰写本书，感慨良多。

把本书献给我的父母 Radosav 和 Slavka，以及我的姐姐 Natalija 和 Dušanka。感谢家人无私地奉献和关爱。特别感谢我的母亲，没有她我不可能成为一名程序员。

非常感谢我美丽的妻子 Lara 的付出，她用耐心和爱鼓励着我，无条件地支持我，告诉我不要放弃。我爱你。

感谢我的好友 Danijel，教我如何成为一个成功的商人，激励我每一天都能成为更好的程序员。

感谢黑山大学的老师们，没有他们我不可能成为现在这样的技术专家。特别感谢 Rajko Čalasan 不厌其烦地教我编程，Milo Tomašević 教我面向对象编程的专业知识，让我从此爱上 C++。特别感谢我最好的老师 Goran Šuković，他经常指导我，带我游历计算机科学的不同领域，让我每天都充满希望，以积极向上的态度学习新的知识。

## 审稿人简介

**Abhishek Gupta** 是印度班加罗尔的一位年轻的嵌入式软件工程师，开发自动车载信息娱乐系统软件多年。Abhishek 于 2011 年在印度理工学院的卡哈拉格普尔理工学院，完成了视觉信息和嵌入系统专业技术硕士的学习。他对视频处理非常感兴趣，喜欢从事嵌入式多媒体系统工作，擅长用 C 和 Linux 进行编程。

欲详细了解他的信息，请访问 [www.abhitak.wordpress.com/about-me](http://www.abhitak.wordpress.com/about-me)。

**Venkateshwaran Loganathan** 是一位杰出的软件开发人员，他工作至今从事过设计、开发和软件产品的测试工作，有 6 年多的工作经验。Venkateshwaran 早在 11 岁时就开始通过 FoxPro 学习计算机编程，从那以后，他学习并掌握了多种计算机语言，如 C、C++、Perl、Python、Node.js 和 Unix shell 脚本。他热衷于开源开发，为各种开源技术做出了贡献。

Venkateshwaran 现就职于高知特科技公司（Cognizant Technology Solutions），作为一名技术助理从事物联网领域的研究和开发工作。目前，他活跃于使用射频识别（RFID）设备发展未来技术的概念。在加入高知特科技之前，他已经在一些大型 IT 公司工作多年，如 Infosys、Virtusa 和 NuVeda。自从作为网络开发人员开始了他的职业生涯，他在网络、在线学习、医疗保健等各个专业领域都有涉猎。由于在工作上的突出表现，公司授予了他许多奖项和荣誉。

Venkateshwaran 在安那大学获得计算机科学与工程学士学位，目前正在攻读比尔拉技术与科学学院软件系统的硕士学位。除编程以外，他还钻研各种技术和软件技能，为新入职的工程师和在校学生授课。Venkateshwaran 喜欢唱歌和徒步旅行，热衷参与社会服务，喜欢和人打交道。欲详细了解他的情况，请访问网站：<http://www.venkateshwaranloganathan.com>，并给他发邮件：[anandvenkat4@gmail.com](mailto:anandvenkat4@gmail.com)。

Venkateshwaran 还写了一本书：《PySide GUI 应用程序开发》（*PySide GUI Application Development*），已由 Packt 出版社出版。

---

我感慨良多。首先，感谢我的母亲 Anbuselvi 和祖母 Saraswathi，感谢她们对我无私的付出，没有她们的支持和帮助，我不可能达到现在的水平。感谢我的兄弟和朋友们。在困难时期，他们一直都不离不弃地帮助我、祝福我。篇幅有限，无法一一列举所有帮助过我的人，我要在这里向他们表达最诚挚的感谢，感谢你们，我生命中的挚友们。

最重要的是，感谢全能的上帝，让我时刻沐浴在他无尽的祝福中。

---

**Walt Stoneburner** 是一位经验丰富的软件架构师，有超过 25 年的商业应用程序开发和咨询经验，对质量保证、配置管理和安全都有所涉猎。如果刨根问底，他还会承认自己喜欢统计学和编写文档。

Walt 对编程语言设计、协同应用程序、大数据、知识管理、数据可视化和 ASCII 艺术都很感兴趣。他说自己是壁橱极客。Walt 还评估软件产品和消费性电子产品，画漫画，运营一家针对肖像和艺术的自由摄影工作室 (CharismaticMoments.com)，写一些幽默的段子，用手表演一些小戏法，喜爱游戏设计。此外，他还是一名业余无线电爱好者。

Walt 有一个名为 Walt-O-Matic 的技术博客：<http://www.wwco.com/~wls/blog/>，通过 [wls@wwco.com](mailto:wls@wwco.com) 或 [Walt.Stoneburner@gmail.com](mailto:Walt.Stoneburner@gmail.com) 可以直接与他取得联系。

他还参与了其他书籍的审稿：

- ▶ *AntiPatterns and Patterns in Software Configuration Management* (ISBN 978-0-471-32929-9, p. xi)
- ▶ *Exploiting Software: How to Break Code* (ISBN 978-0-201-78695-8, p. xxxiii)
- ▶ *Ruby on Rails: Web Mashup Projects* (ISBN 978-1-847193-93-3)
- ▶ *Building Dynamic Web 2.0 Websites with Ruby on Rails* (ISBN 978-1-847193-41-4)
- ▶ *Instant Sinatra Starter* (ISBN 978-1782168218)
- ▶ *Learning Selenium Testing Tools with Python* (978-1-78398-350-6)
- ▶ *Whittier* (ASIN B00GTD1RBS)
- ▶ *Cooter Brown's South Mouth Book of Hillbilly Wisdom* (ISBN 978-1-482340-99-0)



**Dinesh Subedi** 是 Yomari 私营有限责任公司的一位软件开发人员，目前从事数据仓库技术和商业智能开发。他毕业于尼泊尔加德满都工程学院（IOE）Pulchowk 学校并获得了计算机工程学士学位。他在 [www.codeincodeblock.com](http://www.codeincodeblock.com) 写了四年的博客，发表了许多与 C++ 软件开发相关的文章。

---

感谢我的兄弟 Bharat Subedi 在我审校本书时给予我的帮助。

---

---

# 前言

多线程编程正逐渐成为 IT 行业和开发人员关注的焦点。开发商希望开发出用户友好、界面丰富，而且能并发执行的应用程序。强大的 C++ 语言和本地 Win32 API 特性为多线程编程提供了良好开端。有了强大的 C++，可以轻松地创建不同类型的应用程序，执行并行，而且还能优化现有的工作。

本书是一本实践为主、通俗易懂的 Windows 多线程编程指导。你将学到如何从多线程方案中受益，增强你的开发能力，构建更好的应用程序。本书不仅讲解了创建并行代码时遇到的问题，而且还帮助读者详细理解同步技术。此外，本书还涵盖了 Windows 进程模式、调度技术和进程间通信方面的内容。

本书从基础开始，介绍了最强大的集成开发环境：微软的 Visual Studio。读者将学会使用 Windows 内核的本地特性和 .NET 框架的特性。除此之外，本书还详细讲解了如何解决某些常见的并发问题，让读者学会如何在多线程环境中正确地思考。

通过学习本书，读者将学会如何使用互斥量、信号量、临界区、监视器、事件和管道。本书介绍了 C++ 应用程序中用到的大部分高效同步方式。本书用大量的程序示例，以最好的方式教会读者用 C++ 开发并发应用程序。

本书使用 C++ 本地调用，演示如何利用机器硬件来优化性能。本书最终的目标是传授各种多线程概念，让读者能快速高效地进行并行计算和并发编程。

## 本书涵盖的内容

第 1 章，C++ 概念和特性，介绍了 C++ 编程语言和许多特性。本章重点介绍了程序的结构、执行流和 Windows OS 运行时对象。详细介绍了结构化方法和面向对象方法。

第 2 章，进程和线程的概念，详细介绍了进程和线程对象。本章涵盖了进程模式背后的思想和 Windows 进程的实现。除此之外，还介绍了进程间通信和典型的 IPC 问题，并简要介绍了在用户空间和内核中的线程实现。

第 3 章，管理线程，介绍了进程和线程背后的逻辑。本章涵盖了 Windows OS 特性，如协作式多任务和抢占式多任务。本章还详细介绍了线程同步以及同步对象和同步技术。

第 4 章，消息传递，重点介绍了消息传递技术、窗口处理器、消息队列和管道通信。

第 5 章，线程同步和并发操作，介绍了并行、优先级、分发器对象和调度技术。本章还解释了同步对象，如互斥量、信号量、事件和临界区。

第 6 章，.NET 框架中的线程，概述了 C++/CLI .NET 线程对象。本章简要介绍了托管方法、.NET 同步要素、.NET 线程安全、基于事件的异步模式和 `BackgroundWorker` 对象，以及其他主题。

第 7 章，理解并发代码设计，涵盖了诸如性能因素、正确性问题、活跃性问题等特性。通过本章的学习，用户能从另一个更好的视角理解并发和并行应用程序设计。

第 8 章，高级线程管理，重点介绍更高级的线程管理知识。本章详细介绍了线程池的抽象、定制分发对象，以及死锁的解决方案。最后，本章介绍了一个远程线程的示例，演示高级管理。

附录涵盖了 MySQL Connector C 和 WinDDK 的具体安装步骤。另外，还介绍了如何为驱动程序编译和 OpenMP 编译设置 Visual Studio。另外，还介绍了 DebugView 应用程序的安装步骤，演示了它的使用步骤。

## 本书必备软件

要运行本书中的示例，必须安装下面的软件。

▶ Visual Studio 2013

□ <http://www.visualstudio.com/downloads/download-visual-studio-vs#d-express-windows-8>

▶ Windows 驱动程序套件：WinDDK

□ <http://msdn.microsoft.com/en-us/windows/hardware/hh852365.aspx>

▶ MySQL Connector C

□ <http://dev.mysql.com/downloads/connector/c/>

## 本书的读者对象

本书主要面向中级和高级水平的读者。本书介绍的同步概念非常基础，因此也可作为对这方面技术感

兴趣的所有读者和开发人员的参考书籍。最后两章为水平较高的读者准备了一些高级知识，概述了并发设计和高级线程管理等主题。

## 本书的体例

本书通过不同的文本样式以区别不同类型的信息。这里介绍一下这些样式的示例，并解释其含义。

本书出现的代码、数据库表名、文件夹名称、文件名、文件扩展名、路径名、用户输入如下所示：“我们可以通过使用 `include` 指令包含其他上下文”。

代码块如下所示：

```
class CLock
{
public:
    CLock(TCHAR* szMutexName);
    ~CLock();
private:
    HANDLE hMutex;
};

inline CLock::CLock(TCHAR* szMutexName)
{
    hMutex = CreateMutex(NULL, FALSE, szMutexName);
    WaitForSingleObject(hMutex, INFINITE);
}

inline CLock::~CLock()
{
    ReleaseMutex(hMutex);
    CloseHandle(hMutex);
}
```

通过加粗的方式提醒读者注意代码块中的某些部分：

```
class CLock
{
public:
    CLock(TCHAR* szMutexName);
    ~CLock();
private:
    HANDLE hMutex;
};

inline CLock::CLock(TCHAR* szMutexName)
{
    hMutex = CreateMutex(NULL, FALSE, szMutexName);
    WaitForSingleObject(hMutex, INFINITE);
}
```

```
inline CLock::~~CLock()
{
    ReleaseMutex(hMutex);
    CloseHandle(hMutex);
}
```

复制和粘贴本书中的代码时，要特别注意。由于书页篇幅有限，一些代码没法都放在一行，不得不分成多行。我们尝试解决这个问题，但是在某些情况下不太可能。我们强烈建议读者在开始编译示例之前，检查每一个示例，特别是那些带双引号的字符串。由于某些特殊情况，代码被分成多行显示，但要读者要明白这些代码是不能分行的，否则无法通过编译。

在“操作步骤”中，当文中写道“添加第 1 章的现有头文件 CQueue.h”时，我们的意思是：读者要使用 Windows 资源管理器，导航至存放有 CQueue.h 文件的文件夹，并且把该文件以及与之相关的 CList.h 复制到当前项目的工作文件夹中。然后在 Visual Studio 中通过【新增】-【现有项】，把该文件添加至项目中。这样才能正确地编译运行示例代码。

新术语和重要的文字加粗显示。菜单上的选项以这种形式显示：打开【解决方案资源管理器】，右键单击【头文件】。



警告或重要的事项。



# 第 1 章 C++ 概念和特性简介

## 本章介绍以下内容：

- ▶ 创建一个 C++ 项目
- ▶ 程序结构、执行流、运行时对象
- ▶ 结构编程方法
- ▶ 理解面向对象编程方法
- ▶ 解释继承、重载和覆盖
- ▶ 理解多态
- ▶ 事件处理器和消息传递接口
- ▶ 链表、队列、栈示例

## 1.1 介绍

系统所执行的进程或抽象是所有操作系统的核心概念。现在绝大多数的操作系统在同一时间内都可以进行多项操作。例如，计算机在用户编辑 Word 文档时，还可以打印该文档、从硬盘缓冲区读数据、播放音乐等。在多任务操作系统中，中央处理单元（CPU）在程序中快速切换，执行每个程序只需几毫秒。

从严格意义上来说，对于单处理器系统，处理器在一个单元时间内只能执行一个进程。操作系统以极快的速度切换多个进程，营造了一个多进程同时运行的假象。与多处理器系统中硬件支持的真正并行相比，单处理器系统的这种并行叫伪并行（*pseudoparallelism*）更合适。

多线程（*multithreading*）是现代操作系统中非常重要的概念。多线程即允许执行多个线程，对完成并行任务和提升用户体验非常重要。

在传统的操作系统中，每个进程都有自己的地址空间和一个执行线程，该线程通常叫主线程（*primary thread*）。一般而言，运行在同一个进程中的多个线程具有相同的地址空间（即进程的地址空间），在准并行上下文中，这些线程就像是多个单独运行的进程，只不过它们的地址空间相同。



伪并行是操作系统在单处理器环境下的特性。准并行地址空间概念是 Windows 操作系统的特性。在多处理器系统中，Windows 为每个进程提供了一个虚拟地址空间，比真正的物理地址空间大得多，因此叫做准并行上下文。

线程 (*thread*) 是操作系统中的一个重要概念。线程对象包含一个程序计数器 (负责处理在下次线程获取处理器时间时要执行什么指令)、一组寄存器 (储存线程正在操控的变量当前值)、一个栈 (储存与函数调用和参数相关的数据), 等等。虽然线程执行在进程的上下文中, 但是它们的区别很大。进程非常贪婪, 想占用所有的资源; 而线程比较“友好”, 它们彼此合作、交流, 而且共享资源 (如处理器时间、内存和变量等)。

## 1.2 创建 C++ 项目

本书所有的程序示例均在 Visual Studio IDE 中运行。下面, 针对 Visual Studio 介绍如何正确地设置 IDE, 并指出一些影响多线程应用程序的具体设置。

### 准备就绪

确定安装并运行了 Visual Studio (VS)。

### 操作步骤

运行 Visual Studio, 在【开始】界面选择【新建项目】, 会弹出一个有多个选项的窗口。在左边【模板】下面, 选择【C++】, 展开 C++ 节点, 有【CLR】、【常规】、【测试】、【Win32】等选项。然后, 执行以下步骤。

1. 选择 Win32。在中间栏有两个选项:【Win32 控制台应用程序】和【Win32 项目】。

目前, 我们使用【Win32 控制台应用程序】。【Win32 项目】用于有图形用户接口 (GUI) 的应用程序, 而不是控制台程序。如果使用控制台, 要在项目属性中设置其他选项。

2. 选择【Win32 控制台应用程序】, 并在窗口下方的【名称】右边为项目命名。我们把第 1 个 Win32 控制台应用程序项目命名为 TestProject。在【位置】右边选择储存该项目文件的文件夹。VS 将帮你创建一个文件夹, 把用户刚才在【位置】输入的文件夹作为将来创建项目的默认文件夹。现在, 读者应该看到 Win32 应用程序向导窗口。可以直接单击右下方的【完成】, 这样 VS 会自动创建所有需要的文件。或者, 选择【下一步】, 然后在附加选项中勾选【空项目】。如果这样做, 就要自己创建源文件和头文件, VS 不会自动生成所需的文件。

3. 如果在上一步骤的 Win32 应用程序向导窗口中直接选择【完成】, 在【解决方案资源管理器】中就可以看到 stdafx.h 和 targetver.h 头文件, 以及 stdafx.cpp 和 TestProject.cpp 源文件。stdafx.h 和 stdafx.cpp 文件是预处理头文件的一部分, 用于智能感应引擎。该引擎使用翻译单元 (Translation Unit, TU) 模型模仿命令行编译器, 用于智能感应。典型的翻译单元由一个源文件和包含在源文件中的多个头文件组成。当然, 其中还引用了其他头文件, 所以也包含这些被引用的头文件。智能感应引擎从一个特殊的子串开始, 给用户提供的信息 (如, 特定类型是什么、函数和重载函数的原型是什么、在当前作用域中变量是否可用等)。欲了解更多相关内容, 请查阅 MSDN

参考资料 (<http://msdn.microsoft.com>)。

4. TestProject.cpp 文件出现在中间的窗口，这就是编写代码的地方。以后，我们会在更复杂的项目中创建和使用更多的文件，现在先暂时介绍这么多。

## 示例分析

每个程序都必须有自己的主例程，即 main。当运行程序时，操作系统从调用 main 开始。这是执行 C++ 程序的起点。如果编写的代码遵循 Unicode 编程模型，就可以使用 main 的宽字符版本 wmain。当然，也可以使用定义在 TCHAR.h 中的 \_tmain。如果定义了 \_UNICODE，\_tmain 函数相当于 wmain 函数；如果没有定义 \_UNICODE，\_tmain 函数相当于 main 函数。

在 TestProject 窗口上方，有各种各样的按钮和选项。其中有一个包含 Win32 可选项的下拉菜单，这个选项叫做【解决方案平台】。如果要创建 32 位可执行文件，就不用改动。如果要创建 64 位可执行文件，先展开下拉菜单，选择【配置管理器】，找到【活动解决方案平台】，选择【x64】选项。点击【确定】，然后关闭【配置管理器】窗口。

在创建 64 位可执行文件时，最重要的是更改项目属性中的设置。按下 Alt+F7，或者右键单击【解决方案资源管理器】中的 TestProject 项目，选择【属性】，弹出 TestProject 属性页窗口。在【配置属性】的【C/C++】的下拉菜单中选择【预处理器】。在【预处理器定义】中，把 WIN32 改成 \_WIN64 才能创建 64 位可执行文件。其他设置暂不更改。

无论创建 32 位还是 64 位的代码，都要正确设置代码生成。创建 C++ 项目时，可以选择该应用程序是否依赖用户 PC 上 C++ 运行时所需的动态链接库 (DLL)。如果创建的应用程序不仅在本机上运行，还要在其他 PC 上运行，就要考虑这一点。用 VS 在本机开发应用程序，所需的 C++ 运行时库已经安装，不会有任何问题。但是，在其他未安装 C++ 运行时库的 PC 上运行这种应用程序，就有可能出问题。如果确认不依赖 DLL，则需把【运行时库】选项改为【多线程调试 (/MTd)】的调试模式，或改为【多线程 (/MT)】发布模式。调试模式或发布模式在【解决方案配置】的下拉菜单中可任意切换。

对于本书的程序示例，其他选项都不需要改动，因为 32 位和 64 位的机器都能运行 32 位可执行文件。运行时库作为 C++ 软件包框架已经安装在 PC 中了，使用默认设置即可，应用程序在这样的 PC 中运行没有问题。

## 1.3 程序结构、执行流和运行时对象

编程范式是计算机编程的基本样式，主要有 4 种范式：命令式、声明式、函数式（或结构式）、面向对象式。C++ 是当今最流行的面向对象编程语言之一，集功能性、灵活性、实用性于一体。和 C 一样，程序员能很快地适应它。C++ 成功的关键在于，程序员可以根据实际需要做相应地调整。



但是，C++学起来并不轻松。有时，你会认为这是一门高深莫测、难以捉摸的语言，一门永远学不完也无法完全理解和掌握的语言。别担心，学习一门语言并不是要掌握它的所有细枝末节，关键要学会如何正确地用语言特性解决特定的问题。实践是最好的老师，根据具体情况尽可能多地使用相应的特性，有助于加深理解。

在给出示例前，我们先介绍一下查尔斯·西蒙尼的匈牙利表示法。他在1977年的博士论文中，使用元编程（*Meta-Programming*）（一种软件生产方法）在程序设计中制定了标准的表示法。文中规定类型或变量的第1个字母表示数据类型。例如，如果要给一个类命名，Test数据类型应该是CTest。第1个字母C表示Test是一个类。这个方法很不错，因为不熟悉Test数据类型的程序员会马上明白Test是一个类名。基本数据类型也可以这样处理，以int和double为例，iCount表示一个int类型的变量Count，而dValues表示一个double类型的变量Value。有了这些前缀，即使不熟悉代码也很容易识别它们的类型，提高了代码的可读性。

## 准备就绪

确定安装并运行了 Visual Studio (VS)。

## 操作步骤

根据以下步骤创建我们的第1个程序示例。

1. 创建一个默认的 C++控制台应用程序<sup>1</sup>，命名为 TestDemo。
2. 打开 TestDemo.cpp。
3. 输入下面的代码：

```
#include "stdafx.h"
#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Hello world" << endl;
    return 0;
}
```

## 示例分析

不同的编程技术使得 C++程序的结构多种多样。绝大多数程序都必须有#include或预处理指令。

#include <iostream>告诉编译器包含 iostream.h 头文件，该头文件中有许多函数原型。这也意味着函数实现以及相关的库都要放进可执行文件中。因此，如果要使用某个 API 或函数，就要包含相应的头文

<sup>1</sup> 译注：即弹出向导窗口后直接点【完成】。