

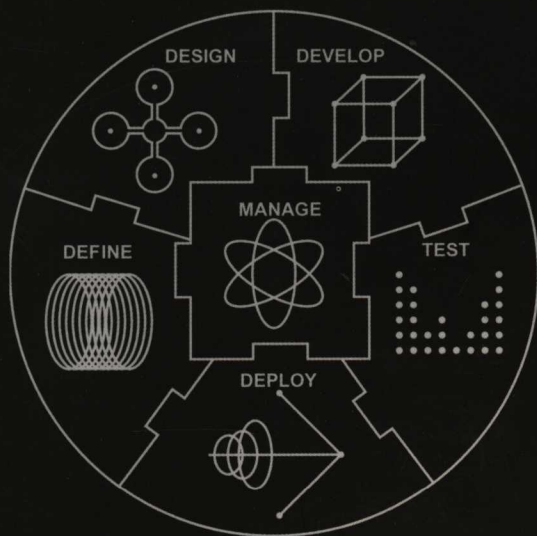
高等职业技术教育IT类双证教材

Borland 软件认证课程推荐教材

Delphi 7

高级应用开发教程

ATA教育公司 总策划
牛汉民 编 著



Delphi 高级程序员认证

 科学出版社
www.sciencep.com

高等职业技术教育 IT 类双证教材
Borland 软件认证课程推荐教材

Delphi 7 高级应用开发教程

(Delphi 高级程序员认证)

ATA 教育公司 总策划

牛汉民 编著

科学出版社

北京

内 容 简 介

本书是高等职业技术教育中 IT 类专业学生获取学历证书和国际著名软件厂商 Borland 软件认证证书的学习用教材。本教材依据课程教学大纲并结合 Delphi 7 高级程序员的的教学特点编写而成。

全书分 11 章, 主要内容包括多线程技术及其应用、组件和包、接口、COM 技术概述、类型库与类型信息、创建 COM 客户端应用程序、创建自动化服务器、创建活动服务器页面、ActiveX 控件的开发和使用、多层数据库应用程序开发和 ModelMaker 入门。为了便于教学使用, 本教材各章编入了大量教学范例, 并拟制了大量习题。本教材结构清晰, 应用实例丰富, 实现了理论学习和具体应用的充分结合。

本书可作为各大专院校、高等职业技术学院计算机软件开发专业课程和相关双证教学课程的教材, 也可供从事 Delphi 7 高级应用开发的程序设计人员学习、参考。

图书在版编目 (CIP) 数据

Delphi 7 高级应用开发教程 / 牛汉民编著, -北京: 科学出版社, 2005
(高等职业技术教育 IT 类双证教材. Borland 软件认证)

ISBN 7-03-016514-4

I. D… II. 牛… III. 软件工具-程序设计-高等学校: 技术学校-教材
IV. TP311.56

中国版本图书馆 CIP 数据核字 (2005) 第 139330 号

责任编辑: 杨 凯 刘晓融 / 责任制作: 魏 谨

责任印制: 刘士平 / 封面设计: 李 力

北京东方科龙图文有限公司制作

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

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

源海印刷有限责任公司印刷

科学出版社发行 各地新华书店经销

*

2006年1月第一版 开本: B5 (720×1000)

2006年1月第一次印刷 印张: 29 3/4

印数: 1~3 000 字数: 573 000

定价: 53.00 元

(如有印装质量问题, 我社负责调换〈新欣〉)

高等职业技术教育IT类双证教材
Borland 软件认证课程推荐教材

编审委员会成员名单

编 委 马肖风 王建国 罗晓中

组织实施 王 健 刘兴国

技术编审 尤克滨 李晓辉 王 健

前 言

Borland 公司自 1983 年成立以来,秉承“持久领先、卓越不凡”的服务宗旨,一直致力于加速整个应用程序的开发周期,不断推出其编译速度快、代码质量高而且易于调试的应用程序开发工具产品,逐步奠定了自己在世界 IT 行业应用程序开发工具领域的领先地位。当前广泛流行的 Borland 公司的软件产品,为我们提供了涵盖定义 (Define)、设计 (Design)、开发 (Develop)、测试 (Test) 和发布 (Deploy) 等各个环节在内的整个软件应用周期的解决方案。Delphi 7 就是 Borland 公司强力推出的重要的应用开发工具之一。

Delphi 7 为我们提供了一个高效率的应用开发环境,其应用范围十分广泛,不仅能够用来编写一般的控制台和 GUI (Graphic User Interface, 图形用户界面) 应用程序,也可以用来开发多媒体应用程序、数据库应用程序和 Web 服务器应用程序。使用 Delphi 7 所提供的各种最新技术,我们还可以非常方便地实现支持所有工业标准的企业级多层分布式应用架构。

Delphi 7 是专门为新一代电子商务领域的应用开发所提供的从应用设计到应用发布的完整解决方案,并引导开发人员开始迈向 .NET 这一全新的平台。基于 Delphi 6 的强大功能,Delphi 7 进一步加强和完善了高性能的数据访问中间件技术 DataSnap、基于组件的 Web 应用程序架构 WebSnap 以及 Web Services 开发平台 BizSnap,进一步改进了集成开发环境、编译器和调试工具,并提供了诸如 IntraWeb 组件、Rave 报表组件等在内的新的特性。为了加强对应用开发周期更加全面的支持,在 Delphi 7 中,还包含了建模工具——ModelMaker 和 Bold for Delphi。

Delphi 7 突破了基于 Windows 平台这一局限,它不仅仍然包含一直为广大开发人员所喜爱的 Windows 环境下的编译器,同时也包含了 Kylix 3 的 Delphi 语言版本以及一个全新的 .NET 编译器预览版本。可以看出, Borland 公司正在其开发工具产品中逐步实现完全平台无关性这一承诺。因此,对于广大开发人员来说,Delphi 仍将继续是我们今后从事应用开发的正确选择。

本书是用于 Delphi 7 高级程序员教学的基本教材。全书共分为 11 章。第 1 章“多线程技术及其应用”在介绍进程和线程的概念及其区别,进程的优先级与进程有关的函数的基础上,进一步讨论线程类 TThread 类的使用,包括线程的

控制、优先级、调用和线程的局部存储，同步问题的解决，创建使用多线程的步骤；第2章“组件和包”介绍了创建组件的一般概念和编写组件的一般步骤，讨论从现有组件中派生出新组件的方法、步骤以及创建图形控件的方法、步骤，讨论什么是组件包，组件包的创建和使用，注册组件、为组件添加属性编辑器和组件编辑器的方法、步骤，能独立编写自定义组件；第3章“接口”讨论了接口的声明、实现、使用和处理高级接口问题的方法；第4章“COM 技术概述”介绍了 COM 技术相关概念的含义、联系和区别，COM 的技术特性，重点讨论一个 COM 应用的基本组成及其技术原理与联系，如何以 COM 的基本设计提供的核心通信功能为基础的扩展技术；第5章“类型库与类型信息”介绍了类型库编辑器的组成和各种类型库元素，类型库所使用的数据类型和语言，对类型库的建立、修改、保存、注册和发布的步骤；第6章“创建 COM 客户端应用程序”说明在 COM 客户端应用程序中如何获取服务器的有关信息，以及自动化控制器和 ActiveX 应用程序的实现方法；第7章“创建自动化服务器”介绍了创建和使用一个新的自动化对象的步骤，定义和使用自动化对象的接口的方法；第8章“创建活动服务器页面”介绍为 ASP 创建、注册进程内和进程外服务器的方法，调试活动服务器页面应用程序的步骤和使用 ASP 内建对象的方法；第9章“ActiveX 控件的开发和使用”介绍 ActiveX 控件的组成元素、使用时要注意的问题、许可的内容、定制 ActiveX 控件的接口的规则、方法和创建 ActiveX 控件与 ActiveForm 的步骤，注册、测试和发布的方法；第10章“多层数据库应用程序开发”专门讨论了 DataSnap 技术和基于提供器的三层应用程序的工作步骤，支持 DataSnap 技术的远程数据模块和数据集提供器组件的使用；第11章“ModelMaker 入门”专门讨论了 ModelMaker 模型、ModelMaker 工具和各种 UML 模式图的创建、使用。

在本书的编写过程中，我们得到了 Borland 公司的授权和大力支持，尤克滨、李晓辉、王健等技术专家就本书的技术审核提供了热情帮助，在此一并表示衷心感谢。

ATA 公司为本书配套提供有教学 PPT 软件、多媒体演示光盘、教学大纲、考试大纲、模拟考试题，以及认证考试和证书授予，供合作院校教师和学生使用、参与。普通读者可以通过 ATA 教育公司网站 (<http://www.atlearning.com/>) 下载获得教学大纲、考试大纲等资源，还可与 ATA 公司电话联系相关事宜：010-65181122。

藉此特别说明，本书是由科学出版社出版的高等职业技术双证教育 IT 类专业 Borland 软件认证课程系列教材之一，针对 Borland 软件认证《Delphi 高级程

程序员认证》证书的考试。本系列还包括针对 Borland 软件认证《Java 初级程序员认证》、《Java Web 应用开发程序员认证》、《J2ME 应用开发程序员认证》、《Delphi 初级程序员认证》、《Delphi 程序员认证》、《Delphi 快速网络开发工程师认证》等证书考试的课程教材。读者可以通过 <http://www.sciencep.com> 了解其他教材的出版情况，并通过 <http://www.atlearning.com> 网站获得相关考试的信息。


北京全美教育技术服务有限公司

(ATA 教育公司)



www.atlearning.com



2005 年 10 月

目 录



第 1 章 多线程技术及其应用	1
1.1 进程与线程.....	1
1.1.1 进 程.....	2
1.1.2 线 程.....	2
1.1.3 进程与线程的区别.....	3
1.1.4 线程的优先级.....	4
1.1.5 与进程有关的函数.....	8
1.2 TThread 类.....	10
1.2.1 线程对象的状态.....	10
1.2.2 TThread 类.....	16
1.2.3 在应用程序中使用多线程.....	17
1.2.4 线程的控制.....	18
1.2.5 使用 Synchronize 方法.....	25
1.2.6 线程优先级的控制.....	26
1.2.7 测试线程代码的执行时间.....	27
1.3 多线程的管理.....	28
1.3.1 线程的局部存储.....	28
1.3.2 多线程的同步机制.....	30
1.4 多线程技术的综合使用.....	43
1.5 多线程与数据库操作.....	50
 练习 题.....	54
第 2 章 组件和包	57
2.1 创建组件的一般概念.....	57
2.1.1 组件库.....	57
2.1.2 组件的属性.....	58
2.1.3 组件的方法.....	59
2.1.4 组件的事件.....	59
2.1.5 组件的拥有关系.....	59
2.1.6 组件的父子关系.....	60



2.2 组件设计基础.....	60
2.2.1 确定是否有必要编写组件.....	60
2.2.2 编写组件的一般步骤.....	60
2.2.3 确定一个祖先类.....	61
2.2.4 创建一个组件单元.....	61
2.2.5 添加属性.....	63
2.2.6 加入事件.....	69
2.2.7 创建自定义的方法.....	75
2.2.8 构造器和析构器.....	75
2.2.9 测试组件.....	76
2.2.10 提供组件图标.....	77
2.2.11 注册组件.....	78
2.3 从现有的组件中派生出新组件.....	79
2.3.1 创建和注册组件.....	79
2.3.2 修改组件类.....	80
2.4 创建一个图形组件.....	81
2.4.1 创建和注册组件.....	81
2.4.2 发布继承的属性.....	82
2.4.3 添加图形功能.....	83
2.5 在设计期间使组件有效.....	89
2.5.1 注册组件.....	90
2.5.2 添加属性编辑器.....	92
2.5.3 属性类别.....	98
2.5.4 添加组件编辑器.....	100
2.6 组件包的使用.....	103
2.6.1 包与 DLL 的比较.....	104
2.6.2 包的建立.....	105
2.7 综合实例.....	106
2.7.1 预备知识.....	107
2.7.2 定义枚举属性和构造函数.....	107
2.7.3 定义组件缺省的大小.....	108
2.7.4 在 protected 部分覆盖定义 paint 过程.....	108
2.7.5 添加箭头高度和填充属性.....	110
2.7.6 添加类类型的属性 TPen 和 TBrush.....	111
2.7.7 定义新定制的事件.....	112
2.7.8 组件的测试.....	114

2.7.9 创建组件包发布组件.....	115
2.7.10 为组件添加位图图标.....	115
 练习题.....	117
第3章 接 口.....	120
3.1 接口的基本概念.....	120
3.1.1 接口的协议意义.....	122
3.1.2 接口和类的不同.....	123
3.1.3 接口的不变性.....	124
3.2 接口的声明和 GUID.....	124
3.2.1 接口的声明.....	124
3.2.2 GUID.....	125
3.2.3 接口的前向声明.....	127
3.3 接口的实现.....	128
3.3.1 接口变量的建立、使用及其引用对象实例的销毁.....	129
3.3.2 IInterface 的实现.....	134
3.3.3 接口指针的获取.....	137
3.4 高级接口问题.....	143
3.4.1 在一个类中实现多个接口.....	143
3.4.2 接口的委托实现.....	147
3.4.3 接口中的属性.....	153
3.4.4 接口在多态方面的应用.....	153
3.5 接口应用举例.....	157
 练习题.....	165
第4章 COM 技术概述.....	168
4.1 COM 技术背景.....	169
4.1.1 COM 技术的发展.....	169
4.1.2 COM 规范和 COM 实现.....	170
4.1.3 软件组件、组件软件与 COM.....	170
4.1.4 COM 的技术特性.....	175
4.2 COM 应用的体系结构.....	176
4.2.1 COM 接口.....	177
4.2.2 COM 服务器.....	179
4.2.3 COM 客户端.....	182
4.2.4 建立和使用提供加法服务的 COM 服务器.....	183

4.3 COM 扩展.....	187
4.3.1 自动化服务器.....	188
4.3.2 活动服务器页面.....	189
4.3.3 ActiveX 控件.....	189
4.3.4 活动文档.....	189
4.3.5 类型库.....	190
4.4 Delphi 对 COM 技术的支持.....	190
4.4.1 向导所支持的接口.....	190
4.4.2 向导所生成的实现类.....	191
4.5 创建并使用简单 COM 对象的范例.....	192
 练习题.....	206
第 5 章 类型库与类型信息.....	208
5.1 类型库编辑器.....	209
5.1.1 类型库编辑器的组成.....	209
5.1.2 类型库元素.....	213
5.2 类型库的建立、修改和发布.....	215
5.2.1 类型库所使用的数据类型和语言.....	215
5.2.2 类型库的建立和修改.....	224
5.2.3 保存和注册类型库信息.....	228
5.2.4 类型库的发布.....	230
5.3 类型库使用举例.....	230
 练习题.....	236
第 6 章 创建 COM 客户端应用程序.....	238
6.1 导入类型库信息.....	239
6.1.1 使用“导入类型库”对话框.....	240
6.1.2 使用“导入 ActiveX”对话框.....	241
6.1.3 导入类型库信息时所生成的代码.....	242
6.2 控制导入的对象.....	243
6.2.1 使用组件封套.....	243
6.2.2 使用数据感知的 ActiveX 控件.....	254
6.2.3 编写基于类型库定义的客户代码.....	256
6.3 创建无类型库服务器的客户端应用程序.....	260
6.4 OLE 数据类型.....	275
6.4.1 OLE 的基本数据类型.....	275

6.4.2 OleVariant 类型	276
 练习题	280
第 7 章 创建自动化服务器	283
7.1 创建自动化对象概述	283
7.1.1 自动化对象的设计	284
7.1.2 使用自动化对象向导	284
7.2 定义自动化对象的接口	287
7.2.1 为对象接口添加属性	288
7.2.2 为对象接口添加方法	288
7.2.3 向客户端发布事件	289
7.3 自动化接口	290
7.3.1 双重接口	291
7.3.2 调度接口	292
7.3.3 定制接口	293
7.4 编组数据	294
7.4.1 自动化兼容的类型	294
7.4.2 自动编组的类型限制	294
7.4.3 定制数据编组	295
7.5 进程内自动化服务器应用范例	295
7.5.1 创建一个执行长度单位换算的自动化服务器	295
7.5.2 长度单位换算的客户端应用程序	298
7.5.3 CreateOleObject 和 GetActiveOleObject	302
7.6 进程外自动化服务器应用范例	302
 练习题	311
第 8 章 创建活动服务器页面	314
8.1 创建活动服务器对象	315
8.1.1 使用 ASP 内建对象	316
8.1.2 为 ASP 创建进程内和进程外服务器	319
8.2 注册活动服务器对象	320
8.2.1 注册进程内服务器	320
8.2.2 注册进程外服务器	320
8.3 调试活动服务器页面应用程序	320
8.4 ASP 创建和使用范例	321

 练习题	328
第 9 章 ActiveX 控件的开发和使用	331
9.1 创建 ActiveX 控件概述	332
9.1.1 创建 ActiveX 控件和 ActiveForm 的步骤	332
9.1.2 ActiveX 控件的组成元素	333
9.2 设计 ActiveX 控件	334
9.3 基于 VCL 控件创建 ActiveX 控件	334
9.4 基于 VCL 窗体创建 ActiveForm	336
9.5 ActiveX 控件许可	337
9.6 定制 ActiveX 控件的接口	338
9.6.1 添加新的属性、方法和事件	339
9.6.2 使用类型库建立简单的数据绑定	341
9.7 为 ActiveX 控件创建属性页面	345
9.7.1 创建新的属性页面	345
9.7.2 在属性页面窗体中添加控件	346
9.7.3 在属性页面中的控件与 ActiveX 控件的属性之间建立关联	346
9.7.4 将属性页面连接到 ActiveX 控件	348
9.8 ActiveX 控件的注册和测试	348
9.8.1 ActiveX 控件的注册	348
9.8.2 ActiveX 控件的测试	349
9.8.3 一个 ActiveX 控件的开发和使用范例	349
9.9 在 Web 上发布 ActiveX 控件	359
9.9.1 发布 ActiveX 控件的步骤	359
9.9.2 设置发布选项	360
 练习题	366
第 10 章 多层数据库应用程序开发	369
10.1 DataSnap 系统基础	369
10.1.1 Datasnap 技术	369
10.1.2 三层应用程序的工作步骤	370
10.2 支持 DataSnap 技术的组件	371
10.2.1 远程数据模块	371
10.2.2 数据集提供者	372
10.3 开发 DataSnap 客户/服务器应用程序	375
10.3.1 建立 DataSnap 应用程序服务器	376

10.3.2	建立 DataSnap 客户端程序	377
10.3.3	与应用程序服务器的连接	379
10.3.4	调用服务器上的接口	381
10.3.5	在客户端纠错	382
10.3.6	更新数据	382
10.3.7	从应用程序服务器获取参数	384
10.4	DataSnap 的简单开发实例	384
10.5	公文包模式及应用范例	390
10.6	DataSnap 应用综合范例	394
	练习 题	416
第 11 章	ModelMaker 入门	419
11.1	ModelMaker 概述	419
11.1.1	ModelMaker 模型	420
11.1.2	ModelMaker 工具	420
11.1.3	UML 模式图	420
11.2	使用 ModelMaker	424
11.2.1	创建模型和添加类模式图	425
11.2.2	定义类	426
11.2.3	定义类的成员	429
11.2.4	使用模型	434
11.2.5	维护模型	439
11.3	使用 ModelMaker 创建一个组件	440
11.3.1	创建一个新的 ModelMaker 工程	440
11.3.2	创建新的类	441
11.3.3	为 TIntegerLabel 类添加属性和方法	442
11.3.4	完成方法的实现	442
11.3.5	创建和生成单元源代码文件	443
11.3.6	调试和安装组件	446
11.3.7	在 ModelMaker 中对组件进行改进	446
11.3.8	组件文档	448
11.3.9	模式图文档	450
	练习 题	453

第 1 章

多线程技术及其应用

本章教学目标

- ➔ 了解进程和线程的概念及其区别
 - ➔ 熟悉进程的优先级和与进程有关的函数
 - ➔ 掌握线程类 TThread 类的使用方法、创建和使用多线程的步骤
- 推荐课时安排 4 学时

Win32 操作系统为我们提供了在应用程序中执行多个线程的能力。在本章中，首先介绍 Win32 操作系统中进程和线程的概念，然后讨论 Delphi 所提供的 TThread 线程类、线程的同步问题，最后，说明多线程应用程序的开发。

1.1 进程与线程

Windows 95/98/NT/2000 和 XP 是多任务的操作系统。所谓多任务操作系统，简单地说，就是可以同时运行多个应用程序的操作系统。比如我们在 Windows 下运行 Delphi 的同时，还可以打开 Word、FrontPage、SQL Server 等应用程序使其投入运行。如图 1-1 所示。

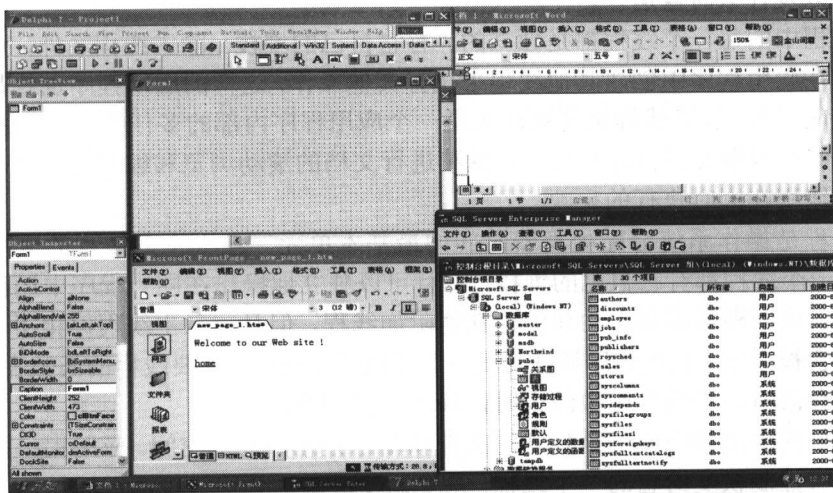


图 1-1 Windows 下运行的多个应用程序

实际上，多任务操作系统是将 CPU 时间片分配给每一个正在运行的应用程序来实现多任务的。在单 CPU 的 PC 机上，操作系统并不可能真正同时运行多个程序，它只是依据程序的优先级分配时间片来轮流运行每一个投入运行的程序。由于所划分的 CPU 时间片非常短，看起来就像多个程序在同时运行一样。

在多任务操作系统环境下开发应用程序，还需要进一步了解进程（Process）和线程（Thread）的概念。

1.1.1 进 程

进程是一个操作系统概念，可以认为进程代表的是一个正在运行的应用程序或一个应用程序的实例。在多任务的 Win32 环境中，可以同时激活多个进程，每个进程的代码和数据均可以获得 4GB 的进程地址空间。在进程的 4GB 地址空间中，除进程自身的代码和数据之外，还包括进程调用的动态连接库（DLL: Dynamic Link Library）的代码和数据等。每一个进程都是由自己惟一的实例句柄（Handle）——一个 32 位的整数——来标识和识别的。

1.1.2 线 程

然而，进程本身并不执行代码，进程的代码是由所谓线程来执行的。线程是一种操作系统对象，代表进程中被执行的代码的路径。每一个 Win32 应用程序都至少拥有一个线程来完成应用程序的执行，该线程就是所谓主线程，也称为默认线程。当一个应用程序投入运行即一个进程被创建之后，系统就会为其创建一个主线程，由主线程在进程的环境中执行代码。

系统调度应用程序时，将控制权交给它的一个线程。系统调度决定了哪些线程运行以及什么时候运行。


进程中的每一个线程都是独立的，有它自己的堆栈。除主线程之外，可以根据需要在应用程序中创建其他线程来执行其他操作，这样的应用程序也就是所谓多线程应用程序。多线程又可被认为是一个应用程序内部的多任务。比如在 Word 中，我们可以在文档打印的同时，继续进行文档的滚动浏览和编辑等操作，这正是多个线程在同时执行的反映。

线程是操作系统分配处理器时间的最基本单元。在一个多线程的应用程序中，每一个线程都有它自己的堆栈，并且可以独立地操作在同一程序中运行的其他线程。可以根据需要启动和停止其他附加线程，但是一旦主线程停止了，整个程序就会被关闭。只要程序还在运行，主线程就在运行。只要线程的函数在运行，线程就存活着，当线程函数结束时，线程就被销毁。

典型的应用程序资源是打开的文件和动态分配的地址空间。应用程序在系统调度时将控制权交给它的一个线程，系统调度决定了哪些线程运行以及什么时候运行。在多处理器计算机中，系统调度可以将单独的线程分配到不同的处理器来

平衡 CPU 负载。

进程中的每一个线程都是独立的，除非用户使线程之间相互可见，否则，线程将独立于进程中的其他线程。

 **注** 进程本身并不直接执行代码，一个进程可以有多个线程，一个进程的所有线程都共享进程的虚拟地址空间，线程们可以共享进程的全局数据和资源。

多线程虽然在控制上比较复杂，但具有如下明显的优点：

(1) 避免瓶颈。使用单线程，应用程序在等待一个用时较长的操作时，CPU 必须停止所有其他的工作，这样导致 CPU 直到操作完成后才真正进入空闲状态；而使用多线程，应用程序可以在一个线程等待的过程中，转而去执行其他的线程，从而避免瓶颈的影响。

(2) 并行操作。应用程序的行为一般可以分为一些可以独立的、并行的操作，使用多线程可以让这些操作并行执行。此外，还可以为不同的任务指定不同的优先级，以便使 CPU 有更多的时间去执行更为紧迫的任务。

(3) 提高效率。如果运行应用程序的系统有多个处理器，可以将应用程序的任务分配到不同的处理器中，并让它们在不同的处理器上同步运行，以提高系统的性能。

1.1.3 进程与线程的区别

所谓进程，就是操作系统要完成的作业，一般是指调入内存中执行的程序。进程由以下几部分组成：

- 进程可以使用的一组私有的虚拟内存地址空间。
- 程序相关的代码和数据。
- 系统资源，如文件、操作系统的同步对象等等。这些系统资源是在程序执行时由操作系统分配给进程的。

线程本身并不直接执行代码，一个进程可以有多个线程，一个进程的所有线程都共享进程的虚拟地址空间，线程之间可以共享进程的全局数据和资源。进程至少包含一个线程，这个线程被称为基本线程，通过该基本线程可以创建多个其他线程。

线程是进程内的一个执行单元，它是操作系统分配 CPU 运行时间的基本实体，进程内的线程由数据结构、CPU 寄存器和堆栈构成。一个进程可以包含多个线程。每个线程都可以独立运行，并保持自己的堆栈和寄存器值。一个线程可以控制另一个线程的运行。