

微软技术开发者丛书

微软资深 MVP，C# 权威技术顾问

全面论述 C# 6.0 的语法基础、程序结构、编程技巧及项目实战

配书 DVD 附赠 300 个实例源代码，500 分钟教学视频

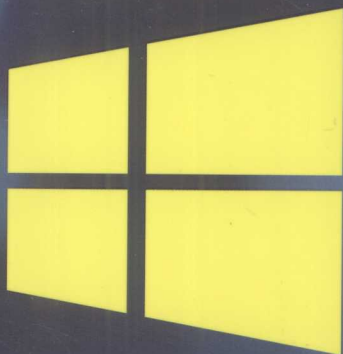
微软（中国）有限公司开发者体验与平台合作事业部推荐用书

微软全球执行副总裁沈向洋作序

# C# 6.0 学习笔记

从第一行 C# 代码到第一个项目设计

全程视频课堂



C# 6.0 Learning Notes

From the First Line of C# Code to the First Project Design

周家安 著  
Zhou Jia'an

清华大学出版社

配书 DVD 赠送  
500 分钟教学视频  
300 个实例代码



### 四十不惑创新不止

从飞鸽传书到指尖沟通，从钻木取火到核能发电，从日行千里到探索太空……曾经遥不可及的梦想如今已经变为现实，有些甚至超出了人们的想象，而所有这一切都离不开科技创新的力量。

对于微软而言，创新是我们的灵魂，是我们矢志不渝的信仰。不断变革的操作系统，日益完善的办公软件，预见未来的领先科技……40年来，在创新精神的指引下，我们取得了辉煌的成绩，引领了高科技领域的突破性发展。

IT行业不墨守成规，只尊重创新。过往的成就不能代表未来的成功，我们将继续砥砺前行。如果说，以往诸如个人电脑、平板电脑、手机和可穿戴设备的发明大都是可见的；那么，在我看来，未来的创新和突破将会是无形的。“隐形计算”就是微软的下一个大事件。让计算归于“无形”，让技术服务于生活，是微软现在及未来的重要研发方向之一。

当计算来到云端后，便隐于无形，能力却变得更加强大；当机器学习足够先进，人们在尽享科技带来的便利的同时却觉察不到计算过程的存在；当我们只需通过声音、手势就可以与周边环境进行交互，计算机也将从人们的视线中消失。正如著名科幻作家亚瑟·查尔斯·克拉克所说：“真正先进的技术，看上去都与魔法无异。”

技术是通往未来的钥匙，要实现“隐形计算”，人工智能技术在这其中起着关键作用。近几年，得益于大数据、云计算、精准算法、深度学习等技术取得的进展，人工智能研究已经发展到现在的感知、甚至认知阶段。未来，要实现真正的人机互动、个性化的情感沟通，计算机视觉、语音识别、自然语言将是人工智能领域进一步发展的突破口及热门的研究方向。

2015年7月发布的Windows 10是微软在创新路上写下的完美注脚。作为史上第一个真正意义上跨设备的统一平台，Windows 10为用户带来了无缝衔接的使用体验，而智能人工助理Cortana、Windows Hello生物识别技术的加入，让人机交互进入了一个新层次。

Windows 10 也是历史上最好的 Windows,最有中国印记的 Windows,不但有针对中国本土的大量优化,还会有海量的中国应用。Windows 10 是一个具有里程碑意义的跨时代产品,更是微软崇尚创新的具体体现,这种精神渗透在每一个微软员工的血液之中,激励着我们“予力全球每一人、每一组织成就不凡”。

四十不惑的微软对前方的创新之路看得更加清晰,走得也更加坚定。希望这套丛书不仅成为新时代之下微软前行的见证,也能够助中国的开发者一臂之力,共同繁荣我们的生态系统,绽放更多精彩的应用,成就属于自己的不凡。

沈向洋

微软全球执行副总裁

# 前言

## PREFACE

---

C# (C Sharp, “#”读作 Sharp, 初学者容易误读为“井”)是微软公司推出的一种完全面向对象、简单易学、现代化的新型编程语言。经过几个版本的演化, C# 的语法已然变成更加丰富和成熟, 并且具有许多其他高级程序设计语言所不具备的特性, 如 LinQ、异步等待等。

C# 与 .NET Framework 平台高度集成, 也就是说, C# 编写的应用程序必须依赖于 .NET 框架, 并可以与如 Visual Basic .NET、Visual C++ .NET 等语言编写的组件进行交互。 .NET 框架向开发者公开庞大的 API 库, 以帮助开发人员快速构建强大的应用程序。

### 本书包含了哪些内容

本书的内容基于最新的 C# 版本(目前最新版本为 6.0), 主要论述了以下方面的内容:

- 关于 C# 与 .NET 的简单介绍;
- 开发环境 Visual Studio 的使用技巧;
- C# 语法基础与应用程序的一般结构剖析;
- 面向对象编程基础;
- 开发控制台应用程序;
- 数据类型转换与字符串的格式化处理;
- 简单的数学运算;
- 泛型与集合;
- 变体的应用;
- 调试、纠错与单元测试;
- Lambda 表达式与 LinQ;
- 开发 Windows 窗体应用程序;
- WPF 基础知识与常用技术;
- 异步编程;
- 系统信息管理;
- 操作对目录、文件、流以及序列化技术;
- 访问数据库;
- 网络编程;

- 反射技术入门；
- 互操作性。

## 如何使用本书

本书采用通俗易懂的讲解风格,尽量避免出现晦涩难懂的描述与理论说教,降低入门者的理解难度。同时,书中针对每个知识点都附有简单的示例程序。读者在阅读本书的时候,可以先阅读每个知识点的说明,然后将附带的示例程序动手操作一遍。在完成书中示例代码后,读者应该运行一下应用程序,并观察程序代码的执行结果,然后思考一下为什么会得到这样的结果,不要做完示例就马上抛在脑后,而应当学会举一反三。

当读者了解书中的示例后,也可以进行“二次创作”,即适当地将示例代码进行修改,使其变成一个新的示例。如此一来,一个示例可以变成两个、三个,甚至更多的示例,读者从中也学会了如何灵活地去运用学到的知识。当然,该做法对于从未进行过任何程序开发的入门者而言会有些困难,刚开始的时候,读者可以抱着尝试的心态去做,不要在乎付出的劳动是否马上得到回报,“欲速则不达”,只要坚持不懈,一点一滴地积累和总结,相信每个有志于步入编程殿堂的读者都会收获颇丰。

在学习编写代码的过程中,读者不妨学会多问几个“为什么”,比如,这段代码为什么要先执行第 1 行,才能去执行第 2 行?如果把第 1 行代码与第 2 行代码调换后会发生什么?这个程序为什么要这样写?如果不这样写又会遇到哪些问题?……这种学习方法有助于帮助初学者形成清晰的思路,养成良好的编程习惯。因为程序代码都是人写的,它是人们事先安排好的一系列指令,交给计算机去执行,程序代码体现了人的主观意志,所以在编写代码的时候,思路非常重要。有了好的思路,并且在满足技术条件的情况下,才能开发出优秀的应用程序。若是思路不清晰,写出来的代码会非常混乱,将来维护起来也会十分艰难。

在本书的最后一章中,提供了两个综合示例,读者通过了解这两个示例的核心功能的实现过程,会初步认识到如何将各个知识点搭配运用,从而加深各个知识要点的印象。

## 本书适合我吗

本书内容侧重基础知识,适合以下读者对象:

- 希望通过自学走上编程之路却不具备基础知识的读者;
- 希望从其他编程语言转向 .NET 方向的读者;
- 对 C# 以及 .NET 相关技术感兴趣的任何读者;
- 可作为培训机构或高等院校开设 .NET 相关教学课程的参考教材。

## 学习本书内容,我能做什么

C# 与 .NET 框架紧密结合,除了可以开发常见的控制台、Windows 应用程序外,还可以用它来开发 Web 应用程序(如网站)、运行在移动设备上(如面向 Windows 10 设备)的应用程序。另外,通过开发环境附带的 Xamarin 及相关技术,还可以使用 C# 来开发运行在

Windows、Android、iOS 等移动终端上的应用程序。

## 勘误与支持

由于作者水平有限,而且编写本书的时间仓促,书中难免会有一些不妥之处,也可能出现错误和不准确的叙述,望读者朋友不吝指出和反馈。读者朋友可以通过以下途径与作者联系:

电子邮件: [csdev2012@foxmail.com](mailto:csdev2012@foxmail.com)

博客: <http://www.cnblogs.com/tcjaan>

微博: <http://weibo.com/tcjaan>

## 鸣谢

本书能够顺利完成,离不开所有亲朋好友们的鼓舞和支持。感谢众多网友所给予的肯定,也感谢清华大学出版社给了我一个写作的机会,感谢盛东亮编辑,在书稿的写作过程中给予了我极大的帮助和支持。

作者

2016年8月

# 学习资源

RESOURCE

为提高读者学习效果,本书配套光盘,提供了如下内容:

## 1. 随书示例源代码

## 2. C# 语法规规范文档

## 3. 教学视频

### 第 1 章 (时长: 00:27:40)

- (1) 为什么要开发应用程序
- (2) dotnet 框架与 C# 的关系
- (3) 初识 VS
- (4) 解决方案与应用程序项目

### 第 2 章 (时长: 00:52:42)

- (1) 代码结构
- (2) 入口点
- (3) 命名空间
- (4) 变量与常量
- (5) 运算符
- (6) 代码流程控制

### 第 3 章 (时长: 01:02:42)

- (1) 类
- (2) 继承
- (3) 抽象类与接口
- (4) 方法与命名参数
- (5) 引用类型和值类型
- (6) ref 参数和 out 参数
- (7) 数组

### 第 4 章 (时长: 00:18:30)

- (1) 在控制台应用程序中输出文本的方法

- (2) 格式化输出的文本

- (3) 读取键盘输入

- (4) 获取命令行参数的方法

### 第 5 章 (时长: 00:16:49)

- (1) 串联字符串的方法

- (2) 拆分字符串

- (3) 字符串内插

- (4) 字符串查找

- (5) Math 类的使用

### 第 6 章(时长: 00:23:00)

- (1) 类型转换帮助器

- (2) 隐式转换

- (3) 强制转换

- (4) 自定义转换

- (5) as 运算符

### 第 7 章 (时长: 00:46:34)

- (1) 集合

- (2) 字典

- (3) 定义泛型类型

- (4) 泛型约束

- (5) 变体

### 第 8 章 (时长: 00:20:48)

- (1) 输出调试信息

- (2) 输出调试信息到日志文件

- (3) 断言

- (4) 禁用断言失败后显示的对话框

- (5) 单元测试

### 第 9 章 (时长: 00:12:46)

- (1) 匿名方法
- (2) lambda 表达式
- 第 10 章 (时长: 00:23:59)
  - (1) LINQ 基础
  - (2) 筛选与排序
  - (3) 分组
  - (4) 联合查询
- 第 11 章 (时长: 00:23:22)
  - (1) Windows 窗体应用程序项目简介
  - (2) 窗体设计器
  - (3) 共享事件处理程序
- 第 12 章 (时长: 00:37:28)
  - (1) WPF 应用程序的项目结构
  - (2) 应用程序模型
  - (3) 代码隐藏
  - (4) 呈现图像
  - (5) 网格布局
  - (6) 渐变画刷
- 第 13 章 (时长: 00:14:17)
  - (1) Thread 类示例
  - (2) Task 示例
  - (3) 异步等待
- 第 14 章 (时长: 00:12:06)
  - (1) 获取环境信息
  - (2) 获取进程列表
  - (3) 启动新进程
- 第 15 章 (时长: 00:14:22)
  - (1) 创建和删除文件
  - (2) 创建和删除目录
  - (3) 流的读写操作
- 第 16 章 (时长: 00:16:49)
  - (1) 二进制序列化
  - (2) XML 序列化
  - (3) JSON 序列化
- 第 17 章 (时长: 00:34:32)
  - (1) 访问 Access 数据库
  - (2) SQLLocalDB 的使用
  - (3) 动态创建和删除数据库
- 第 18 章 (时长: 00:19:50)
  - (1) 程序集信息
  - (2) 动态加载程序集
  - (3) 动态调用类型成员
- 第 19 章 (时长: 00:08:48)
  - (1) TCP 通信示例
  - (2) UDP 通信示例
- 第 20 章 (时长: 00:03:03)
  - 导入 Win32-API 函数



# 目录

## CONTENTS

---

丛书序(沈向洋)	1
前言	3
学习资源	7
<b>第 1 章 准备工作</b>	<b>1</b>
1.1 为什么要编写程序	1
1.2 C#与.NET 框架的关系	2
1.3 只需要一个工具——Visual Studio	3
1.3.1 下载和安装	3
1.3.2 认识 Visual Studio	3
1.3.3 如何使用快捷键	4
1.3.4 如何放置子窗口	4
1.3.5 自动隐藏窗口	5
1.3.6 巧用快速启动	6
1.3.7 个性化设置	7
1.4 创建第一个应用程序	12
1.5 解决方案与项目的关系	15
<b>第 2 章 语法基础</b>	<b>17</b>
2.1 代码的基本结构	17
2.2 命名空间	18
2.2.1 为什么要使用命名空间	18
2.2.2 如何访问命名空间中的成员	19
2.2.3 引入命名空间	20
2.2.4 using static 指令	20
2.3 Main 入口点	21
2.3.1 程序代码真的从 Main 方法开始执行吗	22

2.3.2	如何选择入口点	22
2.4	变量与常量	24
2.4.1	变量	24
2.4.2	变量的生命周期	25
2.4.3	常量	25
2.5	命名规则	26
2.6	运算符	27
2.6.1	示例：简单运算	28
2.6.2	示例：自增和自减运算	28
2.6.3	示例：位运算	29
2.6.4	示例：比较运算符	31
2.7	流程控制	32
2.7.1	顺序执行	32
2.7.2	有选择地执行	32
2.7.3	循环往复	36
2.8	写注释是个好习惯	41
2.9	基本的数据类型	42
2.10	不要错过《C#语言规范》文档	45
<b>第3章</b>	<b>面向对象编程</b>	<b>47</b>
3.1	类	47
3.1.1	字段	48
3.1.2	属性	48
3.1.3	方法	51
3.1.4	构造函数与析构函数	53
3.2	结构	57
3.3	引用类型与值类型	59
3.4	ref 参数与 out 参数	61
3.5	方法重载	66
3.6	静态类与静态成员	68
3.7	继承与多态	69
3.7.1	可访问性	69
3.7.2	继承	72
3.7.3	注意可访问性要一致	74
3.7.4	隐藏基类的成员	75
3.7.5	覆写基类成员	76

3.7.6 如何阻止类被继承 .....	79
3.8 抽象类 .....	79
3.9 接口 .....	82
3.9.1 定义接口 .....	83
3.9.2 接口与多继承 .....	83
3.9.3 实现接口 .....	84
3.9.4 显式实现接口 .....	85
3.10 扩展方法 .....	88
3.11 委托与事件 .....	88
3.11.1 定义和使用委托 .....	89
3.11.2 将方法作为参数传递 .....	92
3.11.3 使用事件 .....	93
3.12 枚举 .....	98
3.12.1 使用枚举类型 .....	98
3.12.2 如何获取枚举的值列表 .....	101
3.12.3 如何取得枚举中各成员的名字 .....	102
3.12.4 枚举的位运算 .....	103
3.13 特性 .....	106
3.13.1 自定义特性 .....	108
3.13.2 如何把特性应用到方法的返回值 .....	109
3.13.3 通过反射技术检索特性 .....	110
3.14 数组 .....	111
3.14.1 定义数组的几种方法 .....	111
3.14.2 多维数组 .....	113
3.14.3 嵌套数组 .....	115
3.14.4 复制数组 .....	118
3.14.5 反转数组 .....	119
3.14.6 更改数组的大小 .....	120
3.14.7 在数组中查找元素 .....	121
3.14.8 灵活使用 ArrayList 类 .....	124
<b>第 4 章 控制台应用程序 .....</b>	<b>126</b>
4.1 输出文本信息 .....	126
4.2 获取键盘输入 .....	129
4.2.1 Read 方法 .....	129
4.2.2 ReadKey 方法 .....	129

4.2.3	ReadLine 方法	130
4.3	命令行参数	131
4.4	设置控制台窗口的外观	133
4.5	控制台窗口的大小和位置	134
<b>第 5 章</b>	<b>字符串处理与数学运算</b>	<b>139</b>
5.1	对象的字符串表示形式	139
5.1.1	重写 ToString 方法	139
5.1.2	扩展方法	141
5.2	字符串的串联与拆分	143
5.2.1	使用+运算符串联字符串	143
5.2.2	使用 Join 方法拼接字符串数组	144
5.2.3	使用 Concat 方法创建字符串	145
5.2.4	使用 StringBuilder 类创建字符串	145
5.2.5	使用 Split 方法拆分字符串	148
5.3	转义字符	149
5.4	英文字母的大小写转换	151
5.5	Parse 和 TryParse 方法	152
5.6	查找和修改字符串	154
5.6.1	查找索引	154
5.6.2	存在性查找	156
5.6.3	字符串的增删改	157
5.6.4	填充字符串	159
5.7	格式化字符串	160
5.7.1	IFormatProvider 接口	160
5.7.2	区域性相关的信息	161
5.7.3	实现字符串的格式化	162
5.8	字符串的复合格式化	169
5.9	字符串内插	171
5.10	生成随机数	172
5.11	Math 类与常用的数学运算	173
5.12	与日期时间有关的计算	174
<b>第 6 章</b>	<b>类型转换</b>	<b>176</b>
6.1	隐式转换	176
6.2	显式转换	179

6.3	可以为 null 的值类型 .....	180
6.4	引用类型的兼容性转换 .....	182
6.5	类型转换帮助器 .....	184
6.6	自定义转换 .....	187
<b>第 7 章</b>	<b>泛型、集合与变体 .....</b>	<b>190</b>
7.1	泛型类型 .....	190
7.1.1	泛型与类 .....	191
7.1.2	泛型与接口 .....	193
7.1.3	泛型与类型成员 .....	194
7.1.4	泛型与委托 .....	195
7.1.5	泛型约束 .....	198
7.2	集合 .....	202
7.2.1	普通集合 .....	202
7.2.2	泛型集合 .....	205
7.2.3	字典 .....	213
7.3	自定义排序 .....	218
7.4	变体 .....	223
7.4.1	协变与逆变 .....	223
7.4.2	类型参数的输入与输出 .....	225
7.4.3	必不可少的技巧 .....	229
<b>第 8 章</b>	<b>纠错与单元测试 .....</b>	<b>230</b>
8.1	实时纠正语法错误 .....	230
8.2	代码调试 .....	232
8.2.1	断点法 .....	232
8.2.2	输出信息法 .....	234
8.3	断言 .....	236
8.4	使用日志文件 .....	240
8.5	异常处理 .....	244
8.6	单元测试 .....	249
<b>第 9 章</b>	<b>匿名方法与 Lambda 表达式 .....</b>	<b>254</b>
9.1	匿名方法 .....	254
9.2	Lambda 表达式 .....	255
9.2.1	用于赋值 .....	256

9.2.2 用于参数传递	257
<b>第 10 章 集成化查询</b>	<b>262</b>
10.1 LINQ 基本语法	262
10.1.1 对查询结果类型的巧妙处理	263
10.1.2 延迟执行与强制立即执行	265
10.2 筛选	268
10.3 排序	269
10.4 分组	271
10.5 更复杂的查询	276
10.5.1 动态创建类型	276
10.5.2 联合查询	278
10.5.3 嵌套查询	284
<b>第 11 章 Windows 窗体应用程序</b>	<b>288</b>
11.1 Windows 窗体应用程序的基本结构	288
11.1.1 客户区域	288
11.1.2 控件	289
11.1.3 应用程序项目	289
11.1.4 Windows 消息循环	291
11.1.5 演练：手动创建一个 Windows 应用程序	294
11.2 ApplicationContext 类	297
11.3 窗体设计器	300
11.4 控件的基类——Control	304
11.4.1 示例：设置控件的位置和大小	306
11.4.2 示例：设置控件的前景色和背景色	309
11.4.3 示例：调整控件的 Z 顺序	311
11.5 常规控件	314
11.5.1 共享事件处理程序	314
11.5.2 制作“用户登录”窗口	316
11.5.3 CheckBox 与 RadioButton 示例	318
11.5.4 使用 ListBox 控件呈现对象列表	321
11.5.5 自行绘制列表控件的项	328
11.5.6 组合框	330
11.5.7 TextBox 的自动完成功能	332
11.5.8 多视图列表	334

11.5.9	图像呈现控件 .....	341
11.5.10	实时报告进度 .....	348
11.5.11	选择日期和时间 .....	353
11.5.12	规范用户输入 .....	355
11.6	容器控件 .....	360
11.6.1	在容器内部显示滚动条 .....	360
11.6.2	分组面板 .....	362
11.6.3	网格布局面板 .....	362
11.6.4	选项卡 .....	365
11.6.5	自动排列内容的面板 .....	369
11.7	工具栏和菜单 .....	371
11.7.1	工具栏 .....	371
11.7.2	菜单栏 .....	376
11.7.3	上下文菜单 .....	379
11.7.4	自定义承载控件 .....	380
11.8	对话框 .....	384
11.8.1	“打开文件”和“保存文件”对话框 .....	384
11.8.2	字体选择对话框 .....	389
11.8.3	颜色选择对话框 .....	390
11.8.4	自定义对话框 .....	391
11.9	用户控件 .....	395
11.10	自定义控件 .....	397
<b>第 12 章</b>	<b>WPF 应用程序 .....</b>	<b>403</b>
12.1	WPF 应用程序项目的结构 .....	403
12.2	XAML 基础 .....	407
12.2.1	XAML 命名空间 .....	407
12.2.2	代码隐藏 .....	408
12.2.3	标记扩展 .....	409
12.2.4	内容属性 .....	410
12.3	依赖项属性 .....	413
12.3.1	定义和使用依赖项属性 .....	414
12.3.2	使用元数据 .....	416
12.3.3	使用已注册的依赖项属性 .....	422
12.3.4	只读依赖项属性 .....	425
12.3.5	附加属性 .....	427

12.4	路由事件	431
12.4.1	路由策略	434
12.4.2	注册路由事件的类处理程序	436
12.5	认识 WPF 控件	439
12.5.1	Control 类	440
12.5.2	内容控件	444
12.5.3	输入控件	452
12.6	数据绑定与视图呈现	455
12.6.1	用于数据绑定的 XAML 扩展标记	455
12.6.2	使用 InotifyPropertyChanged 接口	462
12.6.3	上下文绑定	467
12.6.4	绑定转换器	473
12.6.5	数据集合控件	476
12.6.6	数据模板	481
12.6.7	数据视图	491
12.6.8	ObservableCollection<T>集合	503
12.7	布局	505
12.7.1	Panel 类	505
12.7.2	网格布局(Grid)	507
12.7.3	栈布局(StackPanel)	511
12.7.4	停靠面板(DockPanel)	512
12.7.5	绝对定位(Canvas)	513
12.7.6	自动换行(WrapPanel)	514
12.7.7	Z 顺序	515
12.8	用户控件与自定义控件	516
12.9	样式与资源	525
12.9.1	样式中的触发器	527
12.9.2	资源的有效范围	529
12.9.3	合并资源字典	531
12.9.4	静态资源与动态资源	534
12.10	图形	535
12.10.1	Shape 类	535
12.10.2	基本图形	540
12.11	动画	542
12.11.1	演示图板	543
12.11.2	简单动画与关键帧动画	545



<b>第 13 章 异步编程</b> .....	549
13.1 为什么要使用异步编程 .....	549
13.2 使用 Thread 类进行异步编程 .....	552
13.3 线程锁 .....	554
13.4 通过委托执行异步操作 .....	558
13.5 并行任务 .....	560
13.5.1 启动简单的并行任务 .....	560
13.5.2 Task 与 Task<TResult>类 .....	565
13.5.3 创建可取消的 Task .....	567
13.6 async 和 await 关键字 .....	570
13.7 ThreadLocal<T> .....	575
13.8 AsyncLocal<T> .....	577
<b>第 14 章 系统信息管理</b> .....	579
14.1 管理进程 .....	579
14.1.1 管理正在运行的进程 .....	579
14.1.2 启动新进程 .....	583
14.1.3 重定向输入/输出流 .....	585
14.2 管理服务 .....	587
14.3 WMI 查询 .....	594
14.4 读取系统参数 .....	599
14.4.1 ComputerInfo 类 .....	599
14.4.2 Environment 类 .....	600
14.4.3 SystemInformation 类 .....	601
14.4.4 用于 WPF 的系统信息类 .....	603
14.5 写入事件日志 .....	604
<b>第 15 章 文件与流</b> .....	607
15.1 创建和删除目录 .....	607
15.2 创建和删除文件 .....	609
15.3 流 .....	610
15.3.1 读写文件 .....	611
15.3.2 内存流 .....	616
15.4 内存映射文件 .....	619
15.5 监控文件系统的变化 .....	623