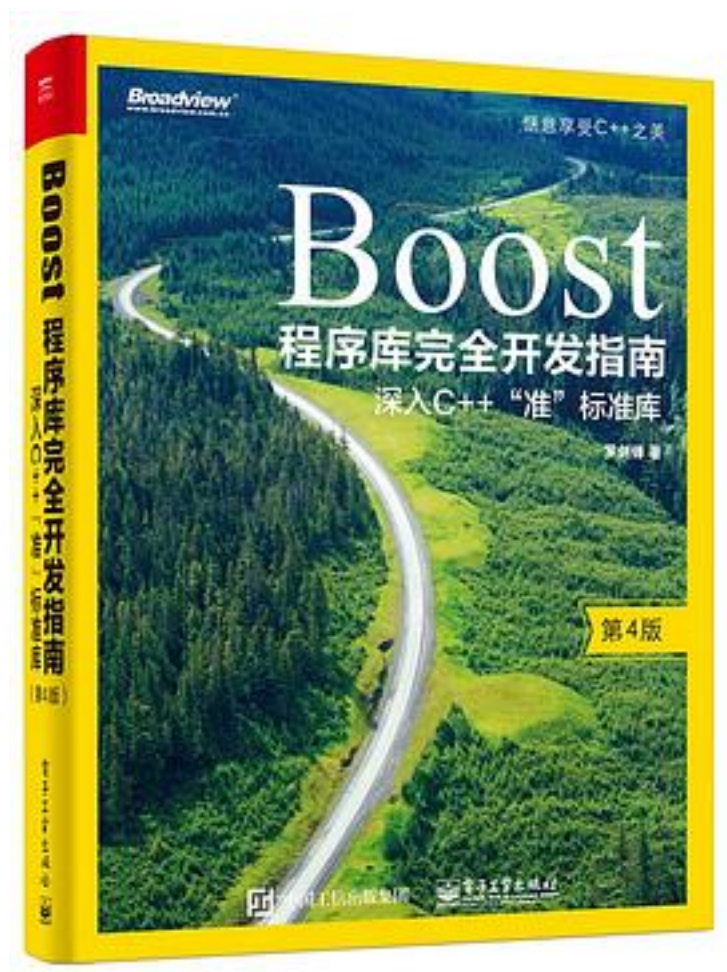


Boost程序库完全开发指南（第4版）



[Boost程序库完全开发指南（第4版）_下载链接1](#)

著者:罗剑锋

出版者:电子工业出版社

出版时间:2017-10

装帧:平装

isbn:9787121324604

Boost是一个功能强大、构造精巧、跨平台、开源并且完全免费的C++程序库，有着“C++“准”标准库”的美誉。

Boost由C++标准委员会部分成员所设立的Boost社区开发并维护，使用了许多现代C++编程技术，内容涵盖字符串处理、正则表达式、容器与数据结构、并发编程、函数式编程、泛型编程、设计模式实现等许多领域，极大地丰富了C++的功能和表现力，能够使C++软件开发更加简捷、优雅、灵活和高效。

《Boost程序库完全开发指南——深入C++“准”标准库（第4版）》基于2017年4月发布的Boost1.64版，介绍了其中的所有140余个库，并且结合C++11/14/17标准详细、深入地讲解了其中数十个库，同时实现了若干颇具实用价值的工具类和函数，可帮助读者迅速地理解、掌握Boost的用法并应用于实际的开发工作。

作者介绍:

罗剑锋（网名Chrono），1996年就读东北某大学，1997年开始接触C/C++。1998年参加全国计算机等级考试，获高级程序员资质。2003年毕业于北京理工大学，获计算机专业硕士学位。目前供职于某部委下属软件公司，任项目经理，主要研究方向为C/C++、设计模式、密码学、数据库、嵌入式系统开发。业余爱好是阅读、欣赏音乐和旅游。

目录: 第 0 章导读	1
0.1 关于本书	1
0.2 读者对象	1
0.3 术语与风格	2
0.4 语言标准	3
0.5 本书的结构	4
0.6 如何阅读本书	5
0.7 本书的源码	5
第 1 章总论	7
1.1 简介	7
1.1.1 获取方式	8
1.1.2 目录结构	8
1.1.3 使用方式	9
1.2 开发环境	9
1.2.1 操作系统和编译器	10
1.2.2 快捷安装	10
1.2.3 完全安装	10
1.2.4 定制安装	11
1.2.5 编译验证	11
1.3 构建工具	12
1.3.1 安装方式	12
1.3.2 构建脚本	12
1.3.3 构建语言	13
1.3.4 构建命令	14
1.4 总结	14
第 2 章时间与日期	15
2.1 timer 库概述	15
2.2 timer	16
2.2.1 用法	16
2.2.2 类摘要	17
2.2.3 使用建议	18
2.3 progress_timer	18
2.3.1 用法	18
2.3.2 类摘要	19
2.4 progress_display	19

2.4.1 类摘要	20
2.4.2 用法	21
2.4.3 注意事项	22
2.5 date_time 库概述	23
2.5.1 使用方式	23
2.5.2 基本概念	24
2.6 处理日期	24
2.6.1 日期	25
2.6.2 创建日期对象	25
2.6.3 访问日期	27
2.6.4 日期的输出	28
2.6.5 转换C 结构	29
2.6.6 日期长度	29
2.6.7 日期运算	30
2.6.8 日期区间	31
2.6.9 日期区间运算	33
2.6.10 日期迭代器	34
2.6.11 其他功能	35
2.6.12 综合运用	36
2.7 处理时间	38
2.7.1 时间长度	39
2.7.2 操作时间长度	40
2.7.3 时间精确度	42
2.7.4 时间点	43
2.7.5 创建时间点对象	44
2.7.6 操作时间点对象	44
2.7.7 转换C 结构	45
2.7.8 时间区间	46
2.7.9 时间迭代器	46
2.7.10 综合运用	47
2.8 date_time 库的高级议题	49
2.8.1 编译配置宏	50
2.8.2 自定义字面值	50
2.8.3 格式化时间	51
2.8.4 本地时间	51
2.9 总结	53
第 3 章内存管理	55
3.1 smart_ptr 库概述	55
3.1.1 RAII 机制	55
3.1.2 智能指针	56
3.2 scoped_ptr	57
3.2.1 类摘要	57
3.2.2 操作函数	58
3.2.3 用法	59
3.2.4 对比标准	61
3.3 shared_ptr	63
3.3.1 类摘要	63
3.3.2 操作函数	64
3.3.3 用法	65
3.3.4 工厂函数	67
3.3.5 应用于标准容器	68
3.3.6 应用于桥接模式	69
3.3.7 应用于工厂模式	70
3.3.8 定制删除器	72

- 3.3.9 高级议题.73
- 3.4 weak_ptr.76
 - 3.4.1 类摘要76
 - 3.4.2 用法77
 - 3.4.3 对象自我管理78
 - 3.4.4 打破循环引用79
- 3.5 intrusive_ptr.80
 - 3.5.1 类摘要80
 - 3.5.2 用法81
 - 3.5.3 引用计数器.82
- 3.6 pool 库概述.83
- 3.7 pool83
 - 3.7.1 类摘要84
 - 3.7.2 操作函数.84
 - 3.7.3 用法85
- 3.8 object_pool 86
 - 3.8.1 类摘要86
 - 3.8.2 操作函数.86
 - 3.8.3 用法87
 - 3.8.4 更多的构造参数88
- 3.9 singleton_pool 89
 - 3.9.1 类摘要89
 - 3.9.2 用法90
- 3.10 pool_alloc 91
- 3.11 总结91
- 第 4 章实用工具.93
 - 4.1 noncopyable.93
 - 4.1.1 原理94
 - 4.1.2 用法94
 - 4.1.3 实现95
 - 4.2 ignore_unused.96
 - 4.2.1 基本用法96
 - 4.2.2 模板用法97
 - 4.3 optional97
 - 4.3.1 类摘要98
 - 4.3.2 操作函数99
 - 4.3.3 用法.100
 - 4.3.4 工厂函数101
 - 4.4 assign.102
 - 4.4.1 list_inserter102
 - 4.4.2 operator+=.103
 - 4.4.3 operator ()104
 - 4.4.4 generic_list105
 - 4.4.5 初始化容器106
 - 4.4.6 重复输入108
 - 4.4.7 操作非标准容器109
 - 4.4.8 其他议题110
 - 4.5 tribool 110
 - 4.5.1 类摘要110
 - 4.5.2 用法.111
 - 4.5.3 为第三态更名.112
 - 4.5.4 输入/输出.113
 - 4.5.5 其他议题113
 - 4.6 operators114

- 4.6.1 基本运算概念.116
- 4.6.2 算术操作符116
- 4.6.3 基类链118
- 4.6.4 复合运算概念.119
- 4.6.5 相等与等价121
- 4.6.6 解引用操作符.122
- 4.6.7 下标操作符123
- 4.6.8 布尔转型操作符124
- 4.6.9 二元操作符126
- 4.7 exception.126
 - 4.7.1 标准库中的异常127
 - 4.7.2 类摘要127
 - 4.7.3 向异常传递信息129
 - 4.7.4 错误信息类130
 - 4.7.5 包装标准异常.132
 - 4.7.6 使用函数抛出异常132
 - 4.7.7 获得更多信息.133
 - 4.7.8 高级议题134
- 4.8 uuid.136
 - 4.8.1 类摘要136
 - 4.8.2 用法.137
 - 4.8.3 生成器139
 - 4.8.4 增强用法141
 - 4.8.5 转换字符串142
 - 4.8.6 摘要算法143
- 4.9 config.144
 - 4.9.1 编译期字符串化144
 - 4.9.2 静态整型常量.145
- 4.10 utility 145
 - 4.10.1 二进制常量146
 - 4.10.2 调用函数名147
- 4.11 总结.148
- 第 5 章字符串与文本处理151
 - 5.1 lexical_cast 151
 - 5.1.1 函数声明152
 - 5.1.2 用法.152
 - 5.1.3 错误处理153
 - 5.1.4 转换对象的要求154
 - 5.1.5 应用于自定义类155
 - 5.1.6 对比标准156
 - 5.2 format 157
 - 5.2.1 简单的例子157
 - 5.2.2 输入操作符159
 - 5.2.3 类摘要160
 - 5.2.4 格式化语法161
 - 5.2.5 性能优化162
 - 5.2.6 高级用法162
 - 5.3 string_ref164
 - 5.3.1 类摘要164
 - 5.3.2 用法.166
 - 5.4 string_algo167
 - 5.4.1 简单示例168
 - 5.4.2 算法概述169
 - 5.4.3 大小写转换169

- 5.4.4 判断式 (算法) 170
- 5.4.5 判断式 (函数对象) 172
- 5.4.6 分类.172
- 5.4.7 修剪.173
- 5.4.8 查找.174
- 5.4.9 替换与删除176
- 5.4.10 分割177
- 5.4.11 合并179
- 5.4.12 查找 (分割) 迭代器179
- 5.5 xpressive181
- 5.5.1 使用方式181
- 5.5.2 正则表达式简介182
- 5.5.3 类摘要183
- 5.5.4 正则匹配185
- 5.5.5 正则查找187
- 5.5.6 正则替换188
- 5.5.7 正则迭代189
- 5.5.8 正则分词190
- 5.5.9 高级议题192
- 5.6 总结194
- 第 6 章正确性与测试197
- 6.1 assert197
- 6.1.1 基本用法197
- 6.1.2 禁用断言198
- 6.1.3 扩展用法199
- 6.2 static_assert 200
- 6.2.1 定义.201
- 6.2.2 用法.201
- 6.2.3 使用建议203
- 6.3 lightweight_test 203
- 6.3.1 测试断言203
- 6.3.2 用法.204
- 6.3.3 测试元编程205
- 6.4 test 205
- 6.4.1 最小化测试206
- 6.4.2 单元测试框架.207
- 6.4.3 测试断言208
- 6.4.4 测试主体209
- 6.4.5 测试实例210
- 6.4.6 测试夹具212
- 6.4.7 测试日志214
- 6.4.8 运行参数215
- 6.4.9 高级议题216
- 6.5 总结219
- 第 7 章容器与数据结构.221
- 7.1 array221
- 7.1.1 类摘要222
- 7.1.2 操作函数222
- 7.1.3 用法.223
- 7.1.4 能力限制224
- 7.1.5 初始化224
- 7.1.6 对比标准225
- 7.2 dynamic_bitset 225
- 7.2.1 类摘要226

- 7.2.2 创建与赋值227
- 7.2.3 容器操作228
- 7.2.4 基本运算229
- 7.2.5 访问元素230
- 7.2.6 类型转换231
- 7.2.7 集合操作232
- 7.2.8 综合运用233
- 7.3 unordered 234
 - 7.3.1 散列集合简介.234
 - 7.3.2 散列集合的用法236
 - 7.3.3 散列映射简介.238
 - 7.3.4 散列映射的用法239
 - 7.3.5 高级议题241
- 7.4 bimap243
 - 7.4.1 类摘要243
 - 7.4.2 基本用法244
 - 7.4.3 值的集合类型.245
 - 7.4.4 集合类型的用法246
 - 7.4.5 使用标签类型.247
 - 7.4.6 使用assign 库.249
 - 7.4.7 查找与替换249
 - 7.4.8 投射.251
 - 7.4.9 高级议题252
- 7.5 circular_buffer253
 - 7.5.1 类摘要253
 - 7.5.2 用法.254
 - 7.5.3 环形结构255
 - 7.5.4 空间优化256
- 7.6 tuple 257
 - 7.6.1 二元组257
 - 7.6.2 类摘要258
 - 7.6.3 创建与赋值258
 - 7.6.4 访问元素260
 - 7.6.5 比较操作261
 - 7.6.6 输入输出261
 - 7.6.7 连结变量262
 - 7.6.8 内部结构263
 - 7.6.9 使用访问者模式264
 - 7.6.10 高级议题266
- 7.7 any 269
 - 7.7.1 类摘要269
 - 7.7.2 访问元素270
 - 7.7.3 用法.271
 - 7.7.4 简化操作271
 - 7.7.5 保存指针272
 - 7.7.6 输出.273
 - 7.7.7 高级议题275
- 7.8 variant276
 - 7.8.1 类摘要276
 - 7.8.2 访问元素277
 - 7.8.3 用法.278
 - 7.8.4 访问器279
 - 7.8.5 高级议题281
- 7.9 multi_array.283

- 7.9.1 类摘要284
- 7.9.2 用法.285
- 7.9.3 改变形状和大小287
- 7.9.4 创建子视图287
- 7.9.5 适配普通数组.289
- 7.9.6 高级议题290
- 7.10 property_tree292
- 7.10.1 类摘要.293
- 7.10.2 读取配置信息294
- 7.10.3 写入配置信息296
- 7.10.4 更多用法297
- 7.10.5 数据格式298
- 7.10.6 高级议题300
- 7.11 总结.302
- 第8章算法305
- 8.1 foreach305
- 8.1.1 用法.306
- 8.1.2 详细解说307
- 8.1.3 更优雅的名字.308
- 8.1.4 支持的序列类型308
- 8.1.5 存在的问题309
- 8.2 minmax.310
- 8.2.1 用法.310
- 8.2.2 存在的问题311
- 8.3 minmax_element 311
- 8.3.1 基本用法312
- 8.3.2 其他用法312
- 8.4 algorithm.313
- 8.4.1 clamp 313
- 8.4.2 clamp_range314
- 8.4.3 hex/unhex314
- 8.5 总结316
- 第9章数学与数字317
- 9.1 math.constants317
- 9.1.1 基本用法318
- 9.1.2 高级用法318
- 9.2 integer319
- 9.2.1 整数特征319
- 9.2.2 标准整数类型.320
- 9.2.3 整数类型模板类322
- 9.3 rational325
- 9.3.1 类摘要326
- 9.3.2 创建与赋值327
- 9.3.3 基本运算327
- 9.3.4 类型转换328
- 9.3.5 输入输出328
- 9.3.6 分子与分母328
- 9.3.7 其他议题329
- 9.4 ratio.330
- 9.4.1 类摘要330
- 9.4.2 用法.331
- 9.4.3 数字单位331
- 9.4.4 字符串表示333
- 9.5 crc 333

- 9.5.1 类摘要334
- 9.5.2 预定义的实现类334
- 9.5.3 用法.335
- 9.6 random336
 - 9.6.1 随机数发生器.336
 - 9.6.2 随机数分布器.338
 - 9.6.3 变量发生器343
 - 9.6.4 产生随机数据块344
 - 9.6.5 真随机数发生器345
- 9.7 总结348
- 第 10 章操作系统相关349
 - 10.1 system.349
 - 10.1.1 错误值.350
 - 10.1.2 错误类别350
 - 10.1.3 错误代码352
 - 10.1.4 错误异常354
 - 10.2 chrono.354
 - 10.2.1 时间长度355
 - 10.2.2 使用时间长度356
 - 10.2.3 时钟358
 - 10.2.4 时间点.360
 - 10.2.5 综合运用361
 - 10.3 cpu_timer363
 - 10.3.1 时间类型364
 - 10.3.2 cpu_timer 364
 - 10.3.4 auto_cpu_timer.366
 - 10.3.5 定制输出格式367
 - 10.4 filesystem.368
 - 10.4.1 类摘要.368
 - 10.4.2 路径表示.370
 - 10.4.3 可移植的文件名.372
 - 10.4.4 路径处理.372
 - 10.4.5 异常处理.375
 - 10.4.6 文件状态.376
 - 10.4.7 文件属性.377
 - 10.4.8 文件操作.378
 - 10.4.9 迭代目录.379
 - 10.4.10 实例1.382
 - 10.4.11 实例2.383
 - 10.4.12 实例3.385
 - 10.4.13 文件流操作.386
 - 10.5 program_options387
 - 10.5.1 概述387
 - 10.5.3 选项值.389
 - 10.5.4 选项描述器391
 - 10.5.5 选项描述器的用法392
 - 10.5.6 分析器.394
 - 10.5.7 存储器.395
 - 10.5.8 位置选项值396
 - 10.5.9 环境变量.398
 - 10.5.10 分组选项信息399
 - 10.5.11 高级用法401
 - 10.6 总结.404
- 第 11 章函数与回调.405

- 11.1 ref405
 - 11.1.1 类摘要.406
 - 11.1.2 基本用法.407
 - 11.1.3 工厂函数.407
 - 11.1.4 操作包装.408
 - 11.1.5 综合应用.409
 - 11.1.6 对比标准.410
- 11.2 bind 411
 - 11.2.1 工作原理411
 - 11.2.2 绑定普通函数413
 - 11.2.3 绑定成员函数414
 - 11.2.4 绑定成员变量415
 - 11.2.5 绑定函数对象416
 - 11.2.6 对比标准416
 - 11.2.7 高级议题417
- 11.3 function421
 - 11.3.1 类摘要.421
 - 11.3.2 声明形式422
 - 11.3.3 操作函数423
 - 11.3.4 用法424
 - 11.3.5 使用ref 库.425
 - 11.3.6 用于回调426
 - 11.3.7 对比auto.428
 - 11.3.8 对比std::function429
- 11.4 signals2429
 - 11.4.1 类摘要.430
 - 11.4.2 操作函数431
 - 11.4.3 用法432
 - 11.4.4 返回值.434
 - 11.4.5 合并器.434
 - 11.4.6 管理信号连接436
 - 11.4.7 更灵活的管理信号连接437
 - 11.4.8 自动管理连接439
 - 11.4.9 应用于观察者模式441
 - 11.4.10 高级议题444
- 11.5 总结.449
- 第 12 章并发编程451
 - 12.1 atomic.451
 - 12.1.1 类摘要.452
 - 12.1.2 基本用法454
 - 12.1.3 整数用法455
 - 12.1.4 并发顺序一致性.456
 - 12.2 thread458
 - 12.2.1 mutex 459
 - 12.2.2 lock_guard461
 - 12.2.3 unique_lock 463
 - 12.2.4 lock adapter 465
 - 12.2.5 thread 467
 - 12.2.6 使用线程.469
 - 12.2.7 中断线程.472
 - 12.2.8 thread_group475
 - 12.2.9 call_once476
 - 12.2.10 condition_variable 477
 - 12.2.11 shared_mutex481

12.2.12 future 483
12.2.13 shared_future 486
12.2.14 高级议题 487
12.3 asio 491
12.3.1 概述 491
12.3.2 信号 497
12.3.3 定时器 501
12.3.4 网络通信 507
12.3.5 同步通信 514
12.3.6 异步通信 516
12.3.7 解析网络地址 521
12.3.8 使用协程 524
12.3.9 其他议题 526
12.4 总结 530
第 13 章 组件速览 533
13.1 算法 533
13.2 字符串和文本处理 534
13.3 容器与数据结构 535
13.4 迭代器 536
13.5 函数对象与高级编程 536
13.6 泛型编程 538
13.7 模板元编程 539
13.8 预处理元编程 540
13.9 并发编程 540
13.10 数学与数字 541
13.11 输入输出 542
13.12 操作系统相关 543
13.13 语言特性模拟 543
13.14 杂项 544
13.15 总结 546
第 14 章 设计模式 547
14.1 创建型模式 548
14.2 结构型模式 549
14.3 行为模式 552
14.4 其他模式 555
14.5 总结 557
第 15 章 结束语 559
15.1 未臻完美 559
15.2 锦上添花 560
15.3 工夫在诗外 563
15.4 临别赠言 564
附录A 推荐书目 565
附录B 标准简述 567
附录C 关键字浅谈 571
• • • • • ([收起](#))

[Boost程序库完全开发指南（第4版）_下载链接1](#)

标签

Boost

C++11

编程

STL

计算机

编程语言

2020

评论

[Boost程序库完全开发指南（第4版）_下载链接1](#)

书评

外面的4篇书评，清一色的五毛，拜托下次做的专业一点，不要那么明显。。。说明明显是因为这几个用户都满足以下条件： 1. 只读过这一本书 2. 只写过这一篇书评 3. 书评内容短而无意义 4. 没有上传头像 5. 没有好友 PS: 豆瓣写书评就必须得给星星，但这本书我没看过，没资格评价...

赖勇浩 (<http://laiyonghao.com>) 作为一个时不时要用一点 C++ 的程序员，我常常自嘲为斯德哥尔摩综合症患者，用 Python 写着懒散的代码时，会怀念以前编写 C++ 代码的那种被虐感。但当真正要写一些 C++ 代码的时候，又会怀念 Python 带给我的自在，这也许就是所谓的由奢入俭...

入门还行，如果你要做应用级开发,还是看文档吧.
里边的东西用法太简单了，好多也都是大概介绍下，复杂的应用下根本不够用的.
还是得看文档和源码，比如asio这一章....被坑的不浅...
计时器用法介绍太粗略了,谁知到每次计完数还要重新设置超时数值。
io_service介绍太简略了，wo...

这书吧，看看，对boost大概有个了解就行了。。。要真正地在项目中使用boost时不建议参考该书。该书中有些地方说得不明不白很容易让人误解，有的就完全搞错了，比如pool一章；有的该说的没有说看了也无法正确使用。而且，书的章节结构也不太好(我希望能是侯捷翻的那本《C++标准...

在读此书前，我对boost的了解几乎为0
，因为想系统的学习一下boost，所以买了此书。看完此书，总的来说对boost的大多数组件有了入门级的了解，本书对一些常用的组件比如date_time，bind/function,smart_ptr等都有介绍，也配有一些简单的实例，适合boost的初学者，但是对于有一...

以下读后感只针对[《Boost 程序库完全开发指南》第一版] (2010.9) 本书目前 (2018.8) 已被作者更新到了第四版。我没有仔细读过后续的版本，只是从简介中了解到作者精简了部分不常用功能的篇幅。对于 Boost User 来说，除了查阅 [Boost Library Documentation]，将本书作为使用 ...

勘误： P71 shared_ptr 应为：获得指向类型 Y 的指针 p 的管理权 P430 下画线
应为：下划线 P440 P496 const asio::error_code 应为： const system::error_code static
address (const char* str); 应为： static address from_string (const char* str); P505
shar...

只看了书上的thread,date_time等有数的几个章节,体验如下: 优点: 1.
目录很全,罗列出来了boost中实现的几乎所有的跨平台的组件; 2.
初碰boost时,可以大大缩短安装,编译,编写示例代码的时间. 缺点:
1.浏览目录的次数比浏览内容的次数还要多,一般是根据目录再翻过头去看boost...

应该说，这个书对于初学者很不错的。
当然了，看这个书只是对boost有个大概的了解。

实际应用时，还是多看看开发文档吧。对于这个库有个宏观的把握。
里面的一些东西用起来还是很方便的。
作者说在windwos下装boost挺麻烦，但我在ubuntu上 一条命令就搞定了...

我是boost初学者，看了这本书觉得帮助很大，至于有同学由于那个编译脚本的错误就指责这本书觉得有点过，这本书在国人写的技术书里算很严谨了。
内容可以看出来是作者经验的总结，尤其喜欢全书最后一句话：
“生活中不只有C++，代码和编程，还有更多的东西值得我们去体味，朋友...

作为一位c++开发人员，在拿到本书之前听名字我也像大家一样感觉很Nx，可是自从到手了，花了一个小时翻了一遍，心里有种失落的感觉，就把它丢在了废书堆里了。自己安装个boost，目录下的帮助文档比它强多了。本人认为，用上boost的程序员应该说c++应该基础不错，基础不好的不要...

刚刚接触boost，发现这是一本很好的工具书籍，讲的很好很实用，例子也比较典型和代表性，正在学习中，并且已经体验到利用boost开发带来的益处，很不错哦

我是本书的编辑，正与作者商量写本关于Boost高级应用相关的，希望各位Boost大侠能够多多给出宝贵意见，不胜感谢。

之前完全没搞过c++，在学了段时间的基本语法后就研究这个了，感觉还是蛮强大，用来辅助服务段程序开发真的是很不错，性能也很好，代码比较优雅，推荐一读哦，虽然我现在基本不写c++了，但对于想要了解的人来说此书还是不错的，（我现在主要用python了）

Boost是C++中很强大的一个库，但是有时候在写代码时候比较困惑不知如何利用这样强大的功能，目前我在读这本书，觉得是一本很好的工具书。希望更多的C++爱好者能够像我一样从中吸收到自己想要的知识！

一直以来都被boost的强大所吸引，但是国内的相关书籍真是少之又少，前两天无意中发现了这本书，让我更加认识到boost的强大力量。原来我之前所接触到的才是冰山一角，作者的底子很深，深入详细的介绍了boost库，很高兴能发现这本书，感谢作者！

一直以来，boost的资料都是网上查找，比较麻烦，最近买了一本，感觉不错，可以当成一本随手的手册，比较方便，推荐大家可以读一下。

[Boost程序库完全开发指南（第4版）_下载链接1](#)