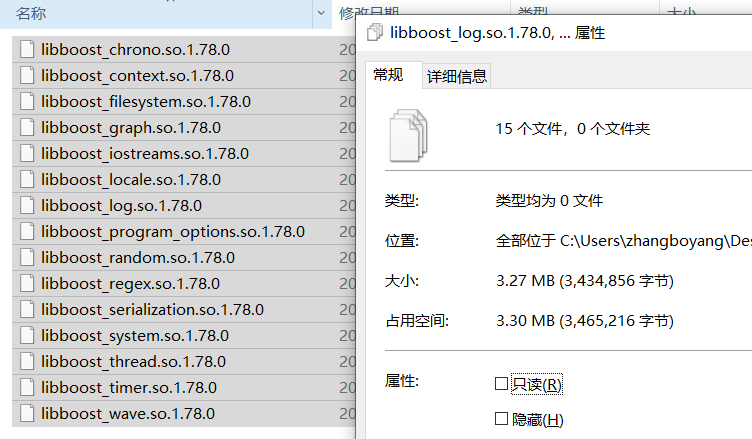
.so 动态编译

<https://www.boost.org/doc/libs/?view=condensed>

<https://blog.csdn.net/muyuxuebao/article/details/51972049> （中文版Boost开发指南）

Extension: determined according to the operating system's usual convention. On most unix-style platforms the extensions are .a and .so for static libraries (archives) and shared libraries, respectively. On Windows, .dll indicates a shared library and .lib indicates a static or import library. Where supported by toolsets on unix variants, a full version extension is added (e.g. ".so.1.34") and a symbolic link to the library file, named without the trailing version number, will also be created.



Chrono：一些实用的time组件。Minutes,seconds等（由duration实现），表示时间点的time\_point等。

Context：基础库，在单个线程上提供**协作多任务处理（**[callcc()](https://www.boost.org/doc/libs/1_78_0/libs/context/doc/html/context/cc.html#cc)/[continuation](https://www.boost.org/doc/libs/1_78_0/libs/context/doc/html/context/cc.html#cc) **）**。通过提供当前线程中当前执行状态的抽象，包括堆栈(带有局部变量)和堆栈指针、所有寄存器和CPU标志以及指令指针。需要C++11支持。

Filesystem：提供对路径、目录和文件操作的可移植性。这种高级抽象使C++程序员可以写出类似于其它编程语言脚本的代码。它提供了便于操作目录和文件的算法。编写要在不同文件系统平台间移植代码的困难工作由于这个库的帮助变得容易了。

Graph：Graph是一个处理图结构的库，它的设计受到STL的重要影响。它是泛型的，高度可配置，并且包括 多个不同的数据结构：邻接链表, 邻接矩阵, 和边列表。Graph还提供了大量的图算法，如Dijsktra最短路径算法，Kruskal最小生成树算法，拓朴逻辑排序，等等。

Iostreams：Boost.IOStreams提供了一个框架，用于定义流、流缓冲区和I/O过滤器。主要有三个作用：便于创建用于访问新源和接收器的标准 C++ stream和stream缓冲区。

提供一个框架来定义过滤器并将它们附加到标准stream和stream缓冲区。

提供即用型过滤器、源和接收器的集合。

例如，Boost.Iostreams 可用于创建stream以访问 TCP 连接或用作加密和数据压缩的框架。 该库包括用于访问内存映射文件、使用操作系统文件描述符访问文件、代码转换、使用正则表达式过滤文本、行尾转换以及 zlib、gzip 和 bzip2 格式的压缩和解压缩的组件。

Locale：为C++提供本地化和Unicode处理工具。通过std::locale类，保存有关特定区域性的所有所需信息的容器，如数字格式模式、日期和时间格式、货币、大小写转换等。

Log：日志库。应用程序将有关其执行的所有基本信息存储到日志中，当出现问题时，此信息可用于分析程序行为并进行必要的更正。日志记录还有其他非常有用的应用程序，例如收集统计信息和突出显示事件。

Program\_options：提供了程序选项配置(名字/值对), 程序选项通常是通过命令行参数或配置文件提供。这个库减轻了程序员手工分析这些数据的负担。

Random：这是一个对随机数的专业使用的库，包括大量的生成器和分配器，可适用于多个不同的领域，如仿真和加密。Rex：正则表达式是解决大量模式匹配问题的基础。它们常用于处理大的字符串，子串模糊查找，按某种格式tokenize 字符串，或者是基于某种规则修改字符串。由于C++没有提供正则表达式支持，使得有些用户被迫转向其它支持 正则表达式的语言，如Perl, awk, 和 sed。Regex提供了高效和强大的正则表达式支持，基于与STL同样的前提 而设计，这使得它很容易使用。

Serialization：这个库允许任意的C++数据结构存进来，再取出去，以及存档。例如，存档可以是文本文件或XML文件。Boost.Serialization是高度可移植的，并提供了非常成熟的特性，如类的版本、C++标准库中的通用类的序列化、共享数据的序列化，等等。

System：兼容boost1.69之前版本。

Thread： Thread是一个提供可移植线程的库，它包含大量线程的原始概念和高度抽象。

Timer：Timer库包含计时所需的特性，它的目标是尽可能做到跨平台的一致性。虽然每个平台都有特定的 API可以让程序 员用于计时，但对于高精度计时还没有可移植的方案。Boost.Timer通过提供最大可能的精度并同时保留可移植性 解决了这个问题，从 而可以让你自由地确定精度。

Wave：Boost.Wave 库是一个符合标准的、高度可配置的强制 C99/C++ 预处理器功能的实现，封装在一个易于使用的迭代器接口后面。