

0-初识STL

-----比特科技整理-----

- 本节目标

1. 什么STL ?
2. STL的版本
3. 了解STL的6大组件
4. 学习STL三境界
5. 如何学习STL ?

- STL概念

STL是C++标准库的最重要的组成部分，STL（标准模板库）不仅是一个可复用的组件库，而且是一个包罗算法和数据结构的软件框架。

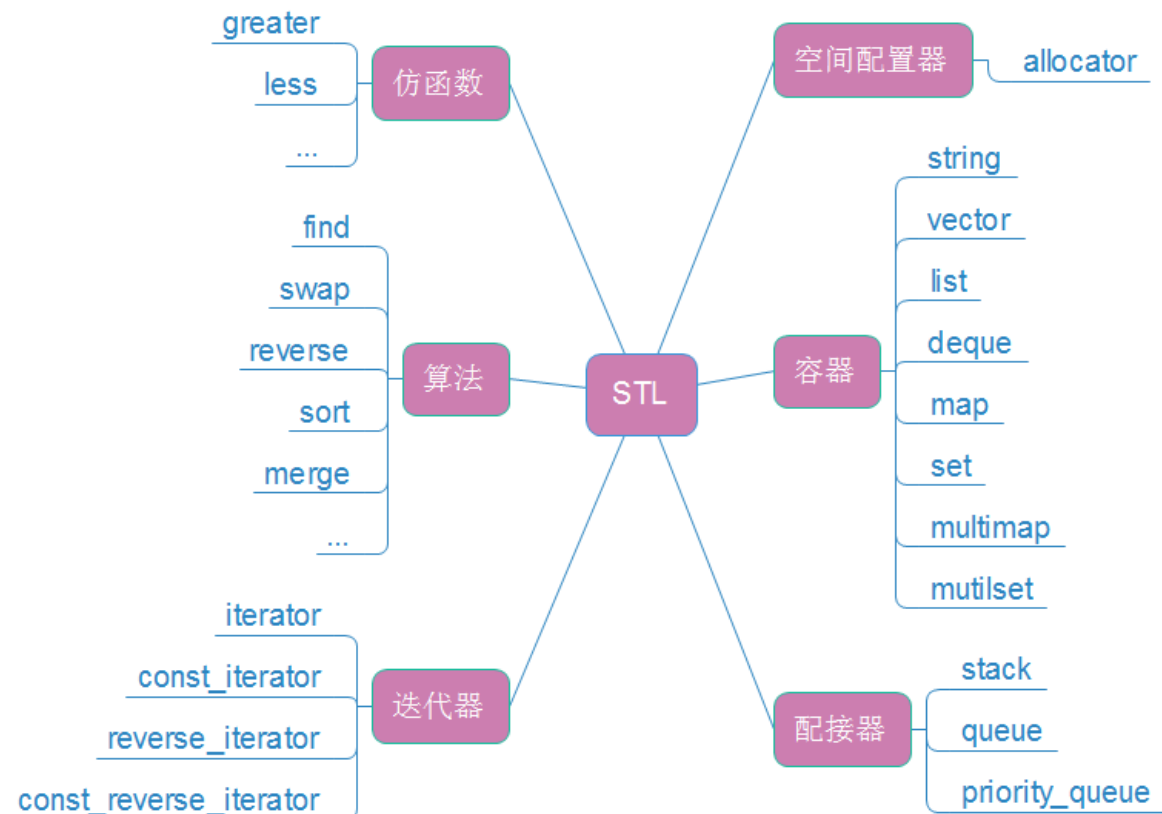
- STL版本

原始版本-Alexander Stepanov、Meng Lee在惠普实验室完成原始版本，本着开源精神，他们声明允许任何人任意运用、拷贝、修改、传播、商业使用这些代码，无需付费。唯一的条件就是也需要向原始版本一样做开源声明。

1. HP版本-也是所有STL的始祖
2. PJ版本-由P.J.Plauger开发--Windows-VS
3. RW版本-由Rouge Wave公司开发
4. SGI版本-有Silicon Graphics Systems, Inc公司开发-Linux

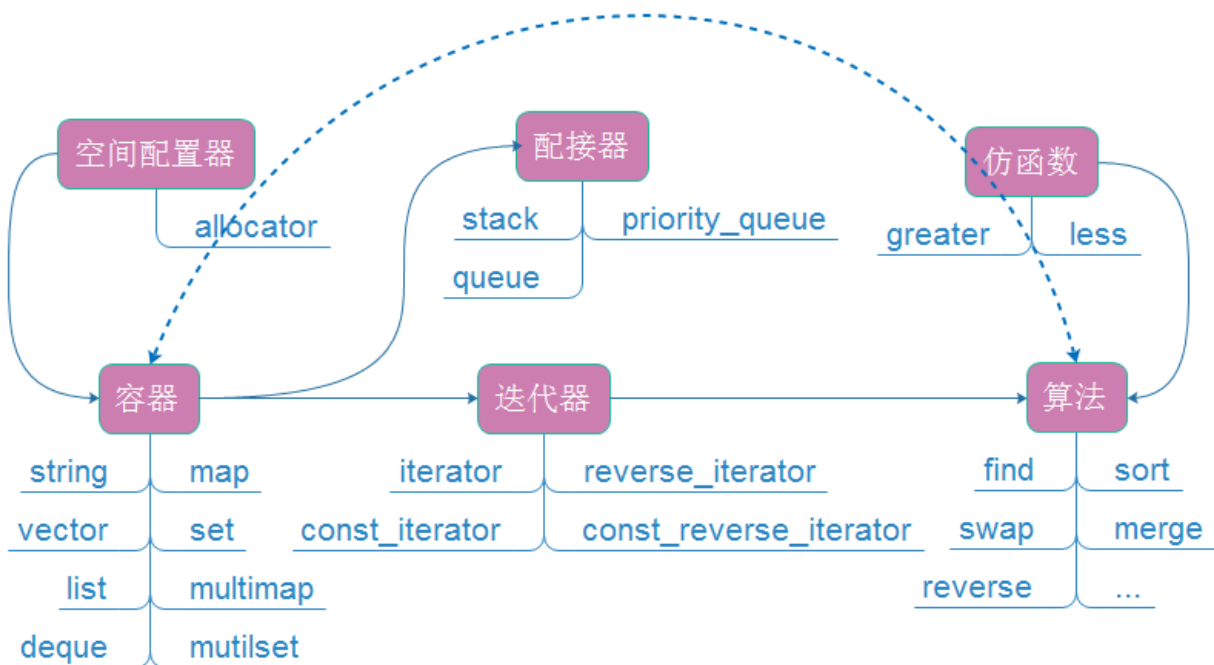
- STL的六大组件

1. 容器-各种数据结构（vector、list、map、set等）
2. 迭代器-扮演容器和算法的胶合剂
3. 空间配置器-负责内存空间的分配与管理
4. 配接器-一种修饰容器或者仿函数或者迭代器接口的东西



5. 算法-各种常见算法（sort、search、copy等）
6. 仿函数（行为类似函数的类，用作算法的某种策略）

- STL六大组件关系思维导图



- STL容器--序列式容器&关联式容器

Containers

<array>	Array header (header)
<bitset>	Bitset header (header)
<deque>	Deque header (header)
<forward_list>	Forward list (header)
<list>	List header (header)
<map>	Map header (header)
<queue>	Queue header (header)
<set>	Set header (header)
<stack>	Stack header (header)
<unordered_map>	Unordered map header (header)
<unordered_set>	Unordered set header (header)
<vector>	Vector header (header)

- C++库在线文档

<http://www.cplusplus.com/reference/>

- 学习STL的三境界

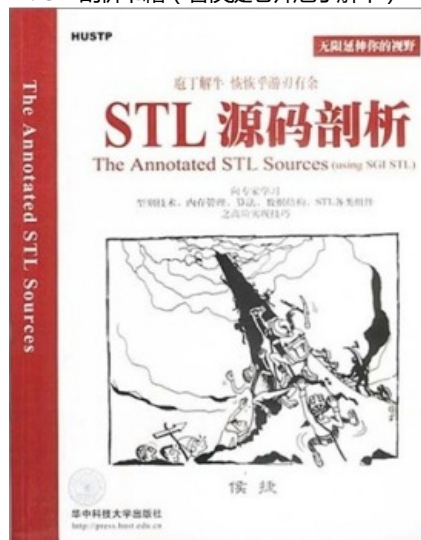
1. 理解STL的整体框架组成、熟练使用STL。
2. 熟悉整理框架以后，剖析STL核心代码，充实数据结构和算法知识，提升泛型编程的技巧，提升功力，一窥大家风范(参考《STL源码剖析》)
3. 对比自己实现的数据结构和算法，思考STL框架设计(为什么要这样设计?)，化繁为简，模拟实现部分STL，更有甚者扩展STL。

- 如何学习STL

1 : STL在线文档 (参考文档学习STL的使用)

<http://www.cplusplus.com/reference/>

2 : STL剖析书籍 (看侯捷老师庖丁解牛)



3 : STL源代码下载 (源码之前，了无秘密)

<http://www.sgi.com/tech/stl/download.html>

ps : 现在STL最新的源码已经更新到V3.3。我们剖析时使用V3.0版，因为相对而言，这个版本的代码可读性比较强，跟侯捷老师的《STL剖析书籍》书上匹配度比较高。

? stl203.tar.gz

? stl.tar.gz

? stl30.tar.gz

? stl32.tar.gz

- 为什么我们要去学习STL源码？

人们常说，不要从轮子重新造起，要站在巨人的肩膀上。面对扮演轮子角色的这些 STL 组件，我们是否有必要深究其设计原理或实现细节呢？答案因人而异。从应用的角度思考，你不需要探索实现细节（然而相当程度地认识底层实现，对实务运用有绝对的帮助）。从技术研究与本质提升的角度来看，深究细节可以让你彻底掌握一切；不论是为了重温数据结构和算法，或是想要扮演轮子角色，或是想要进一步扩张别人的轮子，都可因此获得深厚扎实的基础。

天下大事，必作于细！

我开玩笑地对朋友说，这本书的出版，给大学课程中的“数据结构”和“算法”两门授课老师出了个难题。几乎对所有可能的作业题目（复杂度证明题除外），本书都有了详尽的解答。然而，如果学生能够从庞大的 SGI STL 源码中干净抽出某一部分，加上自己的包装，做为呈堂作业，也足以证明你有资格获得学分和高分。事实上，追踪一流作品并于其中吸取养分，远比自己关起门来写个三流作品，价值高得多——我的确认为 99.99 % 的程序员所写的程序，在 SGI STL 面前都是三流水准 ☺。

ps : STL库的实现需要考虑各种情况和整体各模块的框架设计。很多东西都符合二八原则，STL也一样的，从整体框架来看，它只有20%是它的核心代码，所以在刚开始剖析STL源码时要注意去抓住这20%核心代码，先抛开旁枝末节的实现，先从整体上理解框架组成和这样设计的意义，等到具体剖析部分，再去抓细节，分析它的实现。