## 学习总结文档

从上个月以来（2017年6月9日），开始学习《STL源码剖析》并编写自己的htl\_容器（相当于重新打一遍代码），到今天（2017年7月11日），已经有一个月多两天了。至此学习STL告一段落。

STL分几块内容，包括：

1. Allocator（空间配置器）
2. Iterator and Traits skill（迭代器以及traits编程技巧）
3. Containers（包括序列式容器(sequence container)和关联式容器(associative container)）
4. Algorithm（STL的算法）
5. Functor（仿函数）
6. Adapter（配接器）

目前学习进度：学习并深入的理解了1,2,3,4的内容，尚未深入5,6的内容（待以后学习）

接下来说说对前四章的学习体会总结：

***Allocator:***

首先是Allocator，空间配置器：空间配置器顾名思义就是用来管理空间配置的，空间是用来存储数据的。我们能自由管理空间的工具有malloc/free 以及new/delete。

malloc不负责其申请的空间的初始化，例如构造函数。new则在申请到空间后执行该对象的默认构造函数（如果有且为non trivial？）。Free不负责析构函数，delete则会在销毁前调用析构函数。对于一些有trivial构造函数和析构函数的类，new和delete明显做了多余的工作，因此SGI 的STL版本中使用分离申请空间和构造析构函数来提升效率，即对不需要调用默认构造函数初始化的类型，直接malloc分配，而对于那些需要调用默认构造函数初始化的类型，在malloc后需要再调用construct才能完成初始化，析构时类似调用destroy。

***Iterator和traits:***

然后是Iterator，迭代器就是一种类似指针的类型，而指针的各种行为中最常见以及最重要的是内容提领（dereference）和成员访问，因此迭代器类型中最重要的工作就是重载operator\*()以及operator->()。Operator++()—()等根据需要也要重载来正确移动。

Traits是利用模板的类型自动推导来获得Iterator所指对象的类型，即iterator的相应型别，详情见书本Page-87附件。

***Container:***

包括序列容器：vector, deque, list, / slist, queue, stack, priority\_queue, heap

包括关联容器：rb\_tree(底层), set, map, hashtable(底层), hashset, hashmap

// 其余不详

***Algorithm:***

算法包括很多，也不详。基本模式是，对一个区间执行一个方法，这个区间由迭代器指出，方法可以默认以及使用自定义的仿函数。

2017年7月11日 洪天龙