

# STL 各种容器的实现原理

---

1.vector 底层数据结构为数组，支持快速随机访问

2.list 底层数据结构为双向链表，支持快速增删

3.deque 底层数据结构为一个中央控制器和多个缓冲区，支持首尾（中间不能）快速增删，也支持随机访问

deque是一个双端队列(double-ended queue)，也是在堆中保存内容的.它的保存形式如下:

[堆1] --> [堆2] --> [堆3] --> ...

每个堆保存好几个元素,然后堆和堆之间有指针指向,看起来像是list和vector的结合品.

4.stack 底层一般用list或deque实现，封闭头部即可，不用vector的原因应该是容量大小有限制，扩容耗时

5.queue 底层一般用list或deque实现，封闭头部即可，不用vector的原因应该是容量大小有限制，扩容耗时  
(stack和queue其实是适配器,而不叫容器，因为是对容器的再封装)

6.priority\_queue 的底层数据结构一般为vector为底层容器，堆heap为处理规则来管理底层容器实现

7.set 底层数据结构为红黑树，有序，不重复

8.multiset 底层数据结构为红黑树，有序，可重复

9.map 底层数据结构为红黑树，有序，不重复

10.multimap 底层数据结构为红黑树，有序，可重复

11.unordered\_set 底层数据结构为hash表，无序，不重复

12.unordered\_multiset 底层数据结构为hash表，无序，可重复

13.unordered\_map 底层数据结构为hash表，无序，不重复

14.unordered\_multimap 底层数据结构为hash表，无序，可重复