STL

1、容器是线程安全的吗？您怎么样在多线程的环境下使用容器？

通常多个线程读同一个STL容器是安全的，同时写、或一写一读不安全。

2、map与unordered\_map有什么区别？

map与unordered\_map都是存储键值对的容器，但map是用红黑树来实现，元素有序，查找、删除、插入的时间复杂度为o(logn)；

unordered\_map采用哈希表实现，元素无序，查找、插入、删除的平均复杂度为o(1),最坏情况下为O(n);

3、哈希表的底层实现，哈希平衡算法？

哈希表是由一个数组和一个哈希函数组成。

数组用于存储数据，哈希函数将键映射到数组的索引位置。

哈希冲突：多个键被映射到相同的索引

哈希冲突处理：

链地址法：

每个数组索引维护一个链表（或其它容器），所有哈希到相同索引的元素存储在链表中。

开放地址法：

在数组内寻找下一个空闲位置来存储冲突的元素。

哈希平衡算法：

负载因子是元素数量与数组大小的比值。

当负载因子高过阈值就扩容。当负载因子小于阈值就缩容。 都需要重新计算所有元素的哈希值并重新插入。

4、红黑树有什么好处？

红黑树是一种自平衡二叉搜索树，保证了插入、删除和查找操作的时间复杂度为O(logn)，有效的避免了退化为链表的情况。

红黑树广泛用于实现有序容器，如map和set，能够保持元素的有序性。

5、linux的线程调度

当一个新进程进入就绪队列时，调度器将根据虚拟运行时间插入到红黑树中

当程序被调度器运行并从就绪队列移除时，调度器从红黑树中删除该进程。

调度器通过查找红黑树虚拟运行时间的最小节点来选择下一个要允许的进程

6、Vector底层实现？

std::vector 是基于动态数组实现的，它通过动态分配内存和扩展机制来管理内存。存储元素在连续的内存块中。

vector提供高效的随机访问性能，插入和删除在末尾最为高效，但在中间进行插入删除操作的效率相对较低

8、vector默认大小？

vector在默认构造时是空的，大小为0，不会分配任何内存。

9、vector的扩容机制

容量不足时，通常会1.5倍或2倍的增长率扩展容量。扩容时会分配新内存，复制旧数据到新内存，释放旧内存。

10、迭代器++it、it++哪个好？