**环形数组**

大部分无锁队列都是用环形数组实现的，简单高效，这里也不例外。假设队列长度为queue\_len，用read\_index表示可读的位置，用write\_index表示可写的位置。  
每次修改read\_index或write\_index的时候都需要将其归一化：

read\_index %= queue\_len

队列已使用空间used\_len的计算为：

write\_index >= read\_index ?

write\_index - read\_index : queue\_len - read\_index + write\_index

判断队列IsEmpty的条件为：

read\_index == write\_index

如果不做特殊处理，判断队列IsFull的条件和IsEmpty的条件一样，从而难以区分。所以我们将队列可写入长度设为queue\_len-1。这样判断长度为write\_len的数据是否可以写入的条件为：

// 注意是 < 而不是 <=

used\_len + write\_len < queue\_len

# 一写一读

先来考虑一写一读的场景，实现起来最简单。  
写操作：先判断是否可以写入，如果可以，则先写数据，写完数据后再修改write\_index。  
读操作：先判断是否可以读取used\_len > 0，如果可以，则先读数据，读完再修改read\_index。  
因为read\_index和write\_index都只会有一个地方去写，所以其实不需要加锁也不需要原子操作，直接修改即可。需要注意读写数据的时候都需要考虑遇到数组尾部的情况。