

Atomic Workshop Solutions

Exercise

- Why is the "dummy separator necessary?"
 - iHead represents the element before the oldest
 - iTail represents the element after the newest
 - There is a constraint that iHead and iTail must never represent adjacent elements
 - This entails that the list always contains at least one element
 - Hence, we add a "dummy" element to the newly constructed list

Exercise

- Explain why the race condition occurs
 - iHead and iTail can be accessed from different threads
 - iHead and iTail are not atomic variables
 - There is no ordering of accesses to the variables
- Can the race condition be avoided by using atomic variables?
 - iHead and iTail cannot be atomic, since `list<T>::iterator` is not trivially copyable
 - However, we can use `std::atomic_flag` to implement spin locks to impose ordering on the accesses
 - (We can of course also use mutexes and locks)