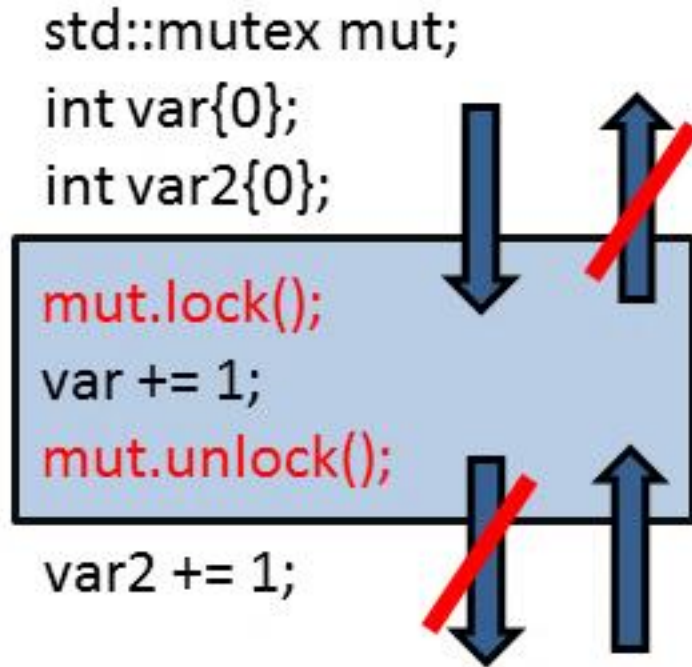


Synchronization and Ordering

Acquire-release semantics

- A release-operation on an atomic synchronizes with a acquire-operation on the same atomic and establishes an ordering constraint.
- Acquire-operation:
 - Read-operation (`load` or `test_and_set`)
- Release-operation:
 - Write-operation (`store` or `clear`)
- Ordering constraints:
 - Read- and write-operations can not be moved **before** an acquire-operation.
 - Read- and write-operations can not be moved **after** a release-operation.

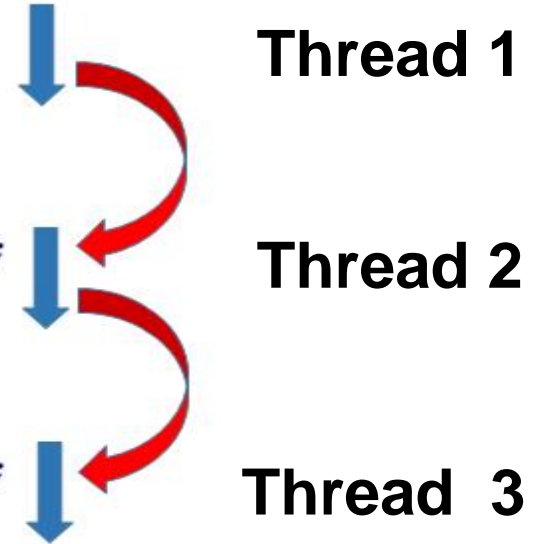
Synchronization and Ordering



- Acquire-operations
 - Locking of a mutex
 - Waiting for a condition variable
 - Starting a thread
- Release-operations
 - Unlocking of a mutex
 - Notification of a condition variable
 - `join`-call a thread

Synchronization and Ordering

```
void dataProducer() {  
    mySharedWork={1,0,3};  
    dataProduced.store(true, std::memory_order_release);  
}  
  
void deliveryBoy() {  
    while( !dataProduced.load(std::memory_order_acquire) );  
    dataConsumed.store(true, std::memory_order_release);  
}  
  
void dataConsumer() {  
    while( !dataConsumed.load(std::memory_order_acquire) );  
    mySharedWork[1]= 2;  
}
```



sequenced-before
synchronizes-with