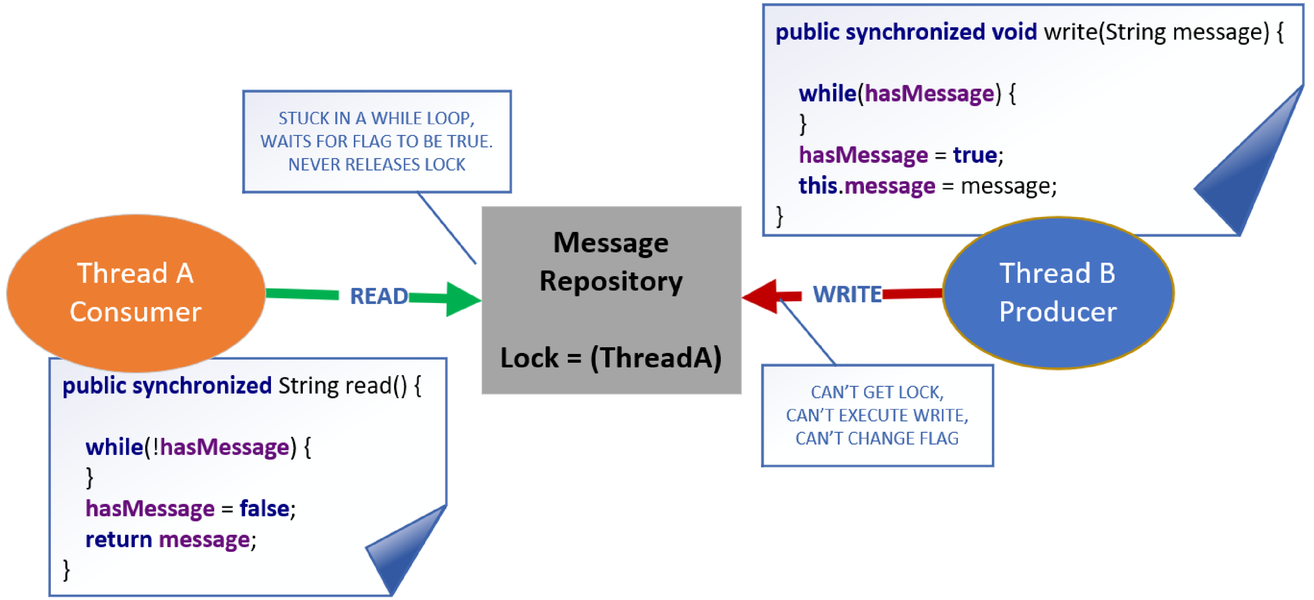
Producer Consumer Application

This kind of application has a class that produces data, so the Producer.

It also has a class that reads the data, or consumes it in some way, this is the Consumer.

Deadlock



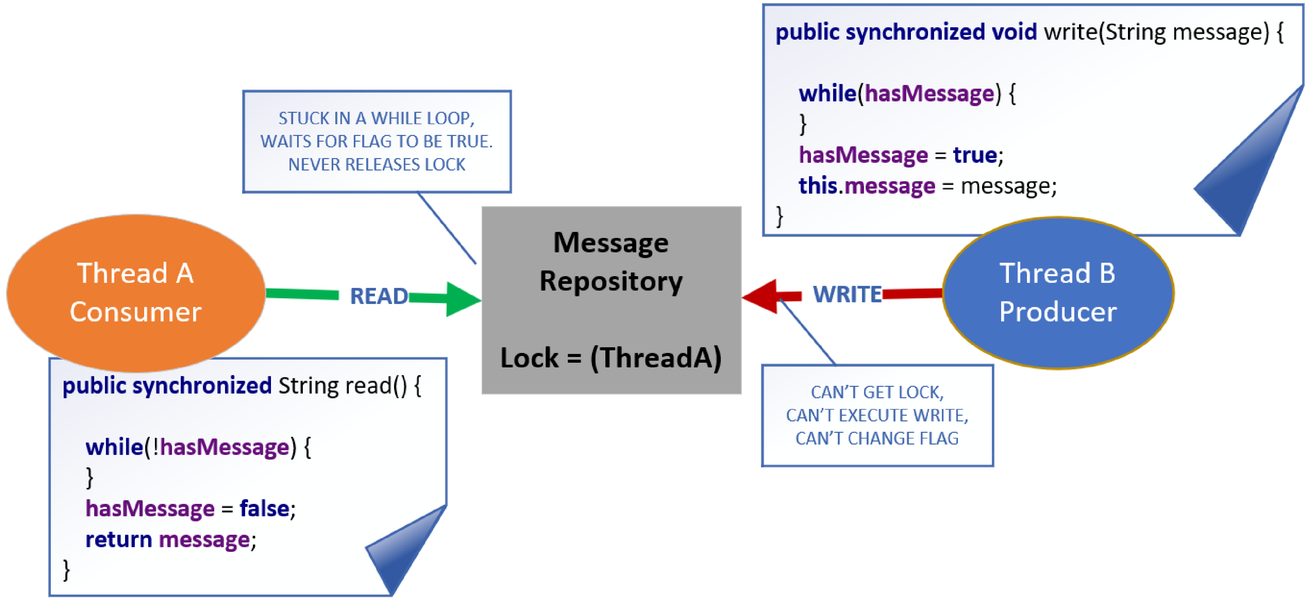
Thread A is our Consumer.

It can usually get in, to run the read method, because the hasMessage flag is usually true.

If the flag is false, it will execute it's while loop.

It's waiting on that hasMessage flag to change value, to exit the loop.

Deadlock

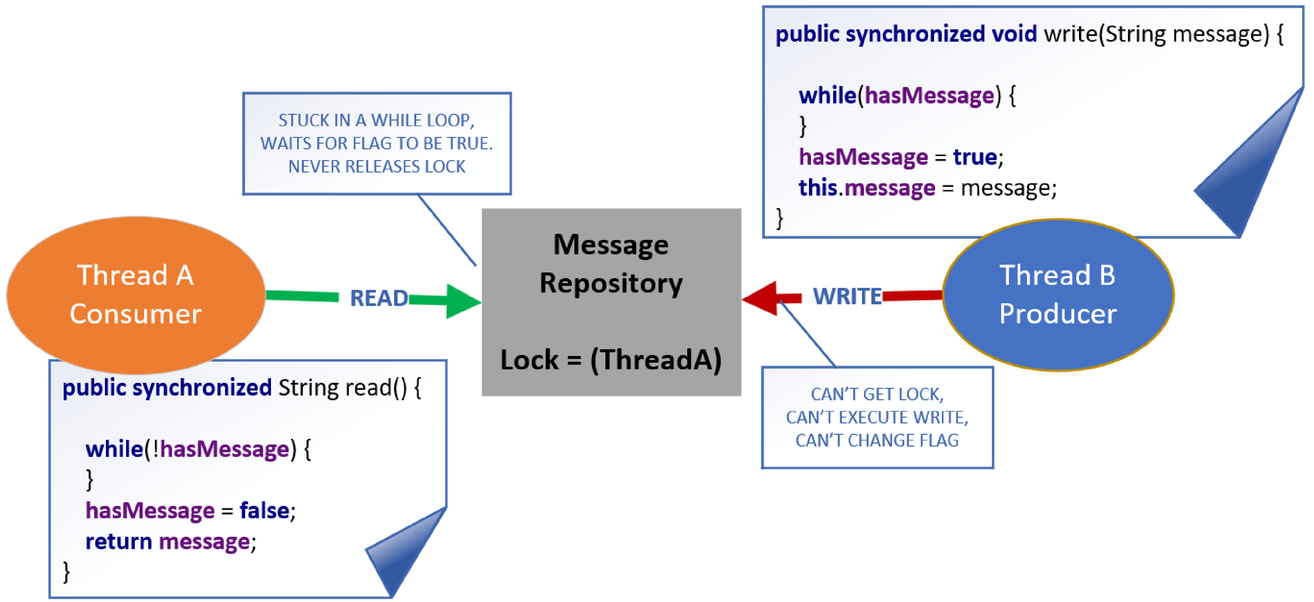


That flag is never going to change its value.

Thread A has acquired a lock on the shared resource, in this case the Message Repository, and Thread B can't get that lock.

Because Thread B is blocked, it can't change the flag, that would set the condition to let Thread A exit it's while loop, and release the lock.

Deadlock



The threads are stuck, one spinning indefinitely, the other blocked from doing anything.

This is a classic deadlock situation.