

## Tricky Mutex

Данная реализация мьютекса гарантирует взаимное исключение: Действительно, поток может зайти в критическую секцию, только если  $thread\_count == 0$ . Каждый поток может вычитать из  $thread\_count$  только в двух случаях:

- 1) если перед этим увеличил  $thread\_count$  при проверке условия в *while*.
- 2) при *unlock()* но это значит, что до этого поток увеличивал  $thread\_count$

Таким образом, каждому уменьшению счётчика сопутствует некоторое его увеличение, которое было сделано до этого. Значит, для того, чтобы поток вошёл в критическую секцию, необходимо, чтобы все зашедшие в критическую секцию потоки вызвали *unlock()*

Но данная реализация мьютекса вовсе не гарантирует свободу от взаимной блокировки: действительно, пусть есть 3 потока, которые зашли в *lock()*, тогда один из них неизбежно зайдёт в критическую секцию. Теперь если планировщик всем остальным потокам даст выполнить только увеличение счётчика, то  $threads\_count \geq 2$  (даже если первый поток успеет сделать *unlock()*). Теперь пусть планировщик поступает следующим образом: каждому из потоков даёт выполнить ровно по две строчки кода (уменьшить счётчик, увеличить счётчик, проверить условие цикла). Тогда ни один поток не сможет зайти в критическую секцию, так как  $threads\_count \geq 1$  до увеличения счётчика.