B. Tricky Mutex

Калугин Дима

Март 2017

Гарантирует ли эта реализация взаимное исключение (mutual exclusion)?
Допустим, что оба потока оказались в критической секции. Это означает, что оба потока вышли из цикла while в методе lock(). Значит, для обоих было выполнено: thread_count.fetch_add(1) <= 0.
Пусть перед тем, как первый поток зашел в критическую секцию значение thread_count равнялось c. После того, как этот поток прошел условие цикла while() оно стало равняться $c + 1$.
Т.к. за одну итерацию цикла while значение thread_count не менятеся (т.к. на каждый вызов fetch_add(1) приходится один вызов fetch_sub(1)), то чтобы второй поток также попал в критическую секцию, необходимо чтобы было выполнено условие: $c + 1 <= 0$. Но это значит, что $c < 0$, но из алгоритма видно, что значение thread_count не может стать отрицательным. Противоречие.
Таким образом, данная реализация горантирует взаимное исключение.

Гарантирует ли эта реализация свободу от взаимной блокировки (deadlock freedom)?

Время	Поток 0	Поток 1	thread_count
1	lock: fetch_add(1)		1
2	Критическая область кода		1
3	Критическая область кода	lock: fetch_add(1)	2
4	unlock: fetch_sub(1)		1
5	lock: fetch_add(1)		2
6		lock: fetch_sub(1)	1
7		lock: fetch_add(1)	2
8	unlock: fetch_sub(1)		1

Далее, повторяя шаги 5-8, видим, что ни один из потоков не попадет в критическую зону. Таким образом получили deadlock.