

Задача. *А. Алгоритм Петерсона.*

Будет ли алгоритм Петерсона для двух потоков гарантировать взаимное исключение, если в doorway-секции метода lock поменять местами взвод флажка ($want[thread_index].store(true)$) и запись в victim ($victim.store(thread_index)$)?

Решение. Приведем пример последовательного исполнения в модели чередования инструкций на одном процессоре, в котором захват мьютекса из двух потоков приводит к нарушению взаимного исключения. Будем управлять планировщиком.

Рассмотрим два потока A и B . Вспомним, что условие выхода, к примеру, потока A из зоны ожидания, это: $victim \neq A$ или $want[B] == false$, аналогично для B .

Последовательность действий. Пускай, в начальный момент времени $want[A] = want[B] = false, victim = A$.

Поступает запрос $lock(A) \rightarrow victim.store(A) \rightarrow$ планировщик переключается на запрос $lock(B) \rightarrow victim.store(B) \rightarrow want[B].store(true) \rightarrow want[A].load() == false \rightarrow B$ заходит в критическую секцию \rightarrow планировщик возвращается к $A \rightarrow want[A].store(true) \rightarrow want[B].load() == true \rightarrow victim.load() \neq A \rightarrow A$ заходит в критическую секцию.

Таким образом в критической секции оказалось два потока одновременно, а значит такой алгоритм не гарантирует взаимного исключения.