

Задача "Ticket spinlock [ticket]", Ткаченко Дмитрий, команда PQ.

Решение: Для начала обозначим $c = std :: thread :: hardware_concurrency()$ (максимальное количество одновременно работающих потоков в данной реализации). Теперь поймем, в каком случае *Ticketspinlock* сильно просядет (просядет число критических секций в секунду). Предположим, что исполняемые в данный момент времени потоки берут по билету. А затем планировщик сразу же сбрасывает их в конец очереди. Тогда, если число потоков равно c , то они сразу же после этого заходят в wait-секцию. Если же число потоков больше c , то следующая группа работающих потоков будет просить по билету, в то время как c ждущих потоков простаивают. Тем самым получаем, что если планировщик сбрасывает исполняемые потоки сразу же после того, как они взяли билет, все они будут простаивать в ожидании пока остальные возьмут билет, даже не зайдя еще в wait-секцию. Значит, число критических секций в секунду будет проседать при росте n , но первое наибольшее проседание произойдет при $n = c + 1$, так как группа из c одновременно исполняемых потоков будет ждать захода в wait-секцию пока еще один поток не получит билет. В $t - a - s$ спинлоке не возникает такой проблемы, так как потоки сразу же проверяют, могут ли они зайти в критическую секцию или нет (не тратят время на получение билета и ожидание, пока остальные потоки получают билет).

Ответ: При $n = std :: thread :: hardware_concurrency() + 1$.