

# Основанное на анализе кода fuzzy-планирование потоков с применением lincheck для обнаружения ошибок в многопоточном коде

## Проблема

Текущие проверки линейности алгоритмов, например с помощью [lincheck](#), заключаются в случайном многопоточном исполнении кода и сравнении результатов со всеми возможными последовательными исполнениями тех же операций. В ходе проверки даже самых простых студенческих lock-free алгоритмов выявилась полезность следующих функций:

1. Останавливать потоки после / перед CAS для повышения вероятностью некорректных ситуаций
2. Долгой остановки потока после / перед CAS с проверкой отработки остальных потоков по своим задачам

## Пути решения

Внедриться в JRE или байт-код и на основе понимания использования CAS явно влиять на планирование потоков

## Ближайшие задачи

1. Разобраться со способами внедрения через байт-код (типа Thread.sleep после / перед CAS)
2. Разобраться с возможностью влиять на планирование потоков в JRE

## Идеи на обсуждение

1. Обёртка на Atomic\*Reference с барьерами

From:  
<http://wiki.osll.ru/> - Open Source & Linux Lab

Permanent link:  
[http://wiki.osll.ru/doku.php/etc:users:kel:jre\\_fuzzy\\_threads\\_planning?rev=1580677383](http://wiki.osll.ru/doku.php/etc:users:kel:jre_fuzzy_threads_planning?rev=1580677383)

Last update: 2020/02/03 00:03

