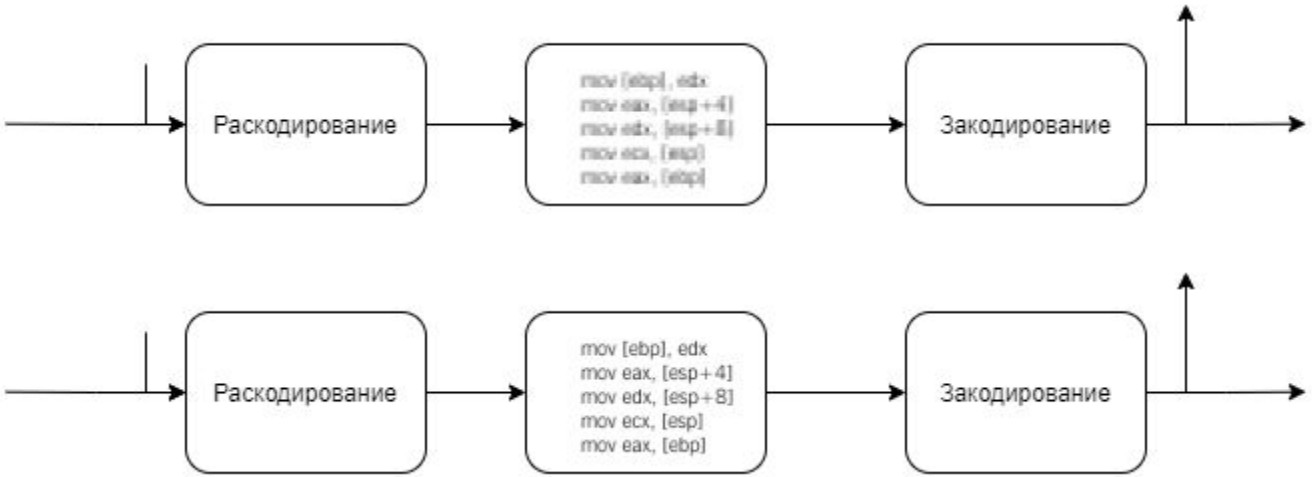
Синхронизация в многопоточных МАК-обфусцированных программах

Бабанов Пётр Андреевич

Научный руководитель:
Доцент к. т. н. Т. А. Брыксин
Научный консультант:
М. В. Баклановский
Рецензент:
ст. преподаватель ФГАОУ УрФУ А. Е. Сибиряков

Введение


Раскодирование
mov [ebp], edx
mov eax, [esp+4]
mov edx, [esp+8]
mov ecx, [esp]
mov eax, [ebp]
Закодирование
Раскодирование
mov [ebp], edx
mov eax, [esp+4]
mov edx, [esp+8]
mov ecx, [esp]
mov eax, [ebp]
Закодирование

Цель работы

Разработать программный модуль, позволяющий встраивать механизмы синхронизации в произвольный файл ассемблерного кода, сгенерированный компилятором МАК

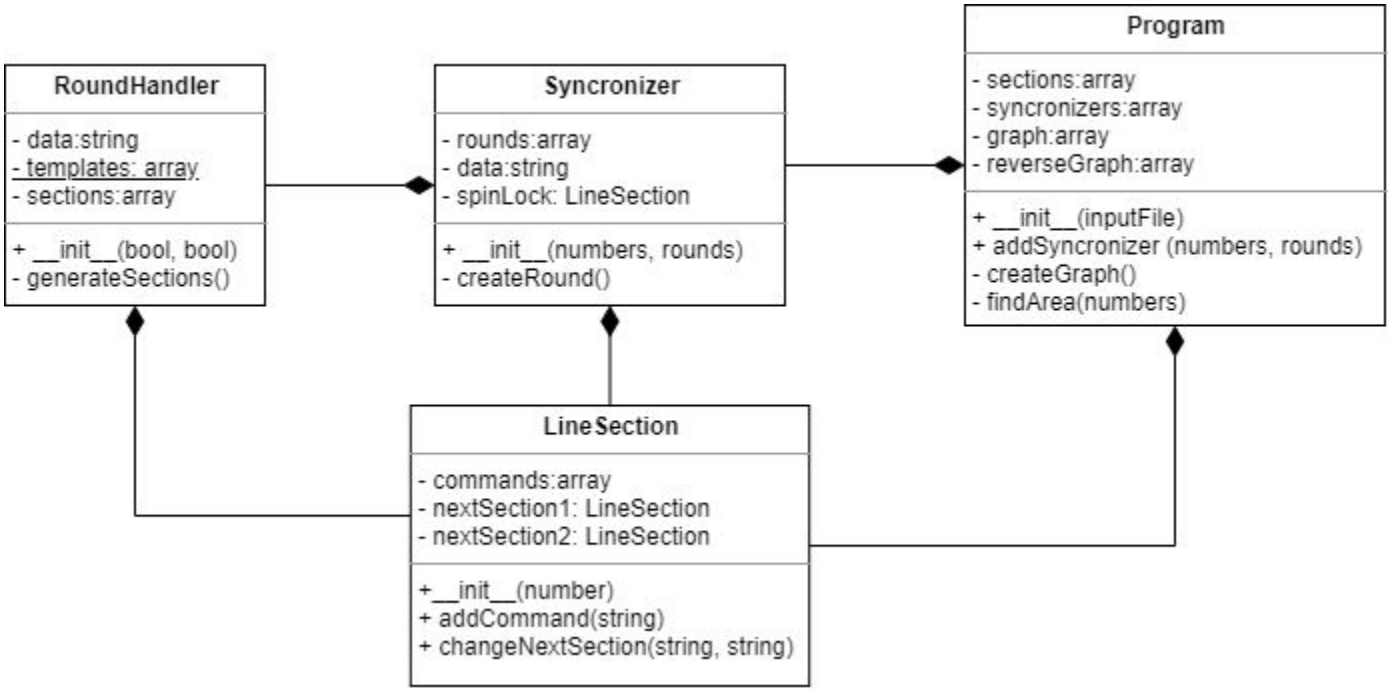
Задачи

- Изучить классификацию механизмов неблокирующей синхронизации и возможности их применения
- Разработать архитектуру требуемого модуля
- Реализовать прототип требуемого инструмента
- Провести тестирование полученного прототипа

Классификация механизмов неблокирующей синхронизации

- Obstruction free
  - При обнаружении конфликтующей операции она прерывается и выполняется позднее
  - Нельзя гарантировать конечное время работы ни одного процесса
- Lock free
  - Один поток не должен блокировать работу другого
  - Для одного потока гарантируется выполнение операции за конечное число шагов
- Wait free
  - Ни один поток не ожидает окончания работы другого
  - Для всех потоков гарантируется выполнение операций за конечное число шагов

Архитектура программного модуля



Требования к обфускаторам

- Одна точка входа
- Одна точка выхода
- Параметры, определяющие начало и протяженность защищаемого участка передаются в соответствии с соглашением о вызовах *cdecl*

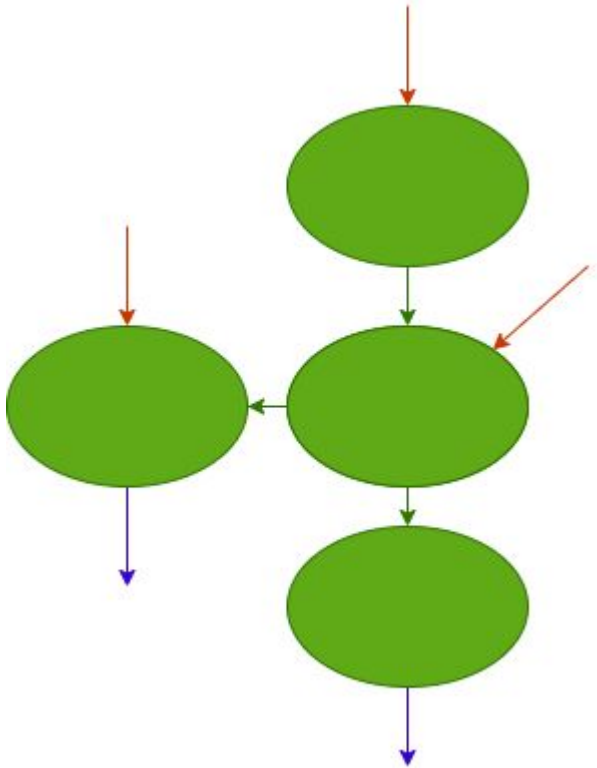
Модификация ранее созданного механизма синхронизации

- Увеличение числа участков, отвечающих за сохранение данных на входе в механизм синхронизации по числу ребер графа потока управления, входящих в модифицируемую область.
- Добавление сохранения адреса линейного участка, следующего после механизма синхронизации
- Замена команд возврата из механизма синхронизации

Добавление одного раунда кодирования

Область, для которой решается задача – набор защищаемых линейных участков и дуг графа потока управления, которые соединяют их.

Считаем, что набор защищаемых линейных участков образует компоненту связности на графе потока управления.

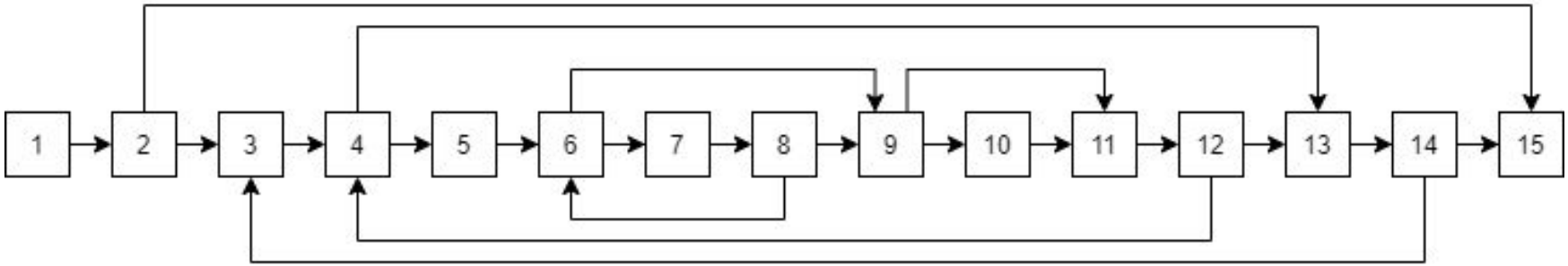


Добавление нескольких раундов кодирования

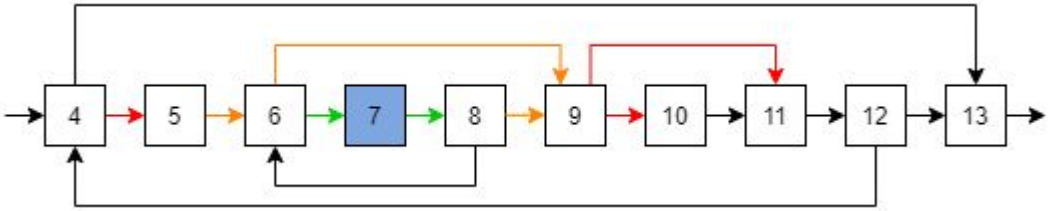
- Решаем задачу добавления одного раунда
- Расширяем область, для которой решаем задачу добавления одного раунда механизмом синхронизации уже добавленного раунда и соседними с ним вершинами
- Повторяем указанные действия для необходимого числа раундов

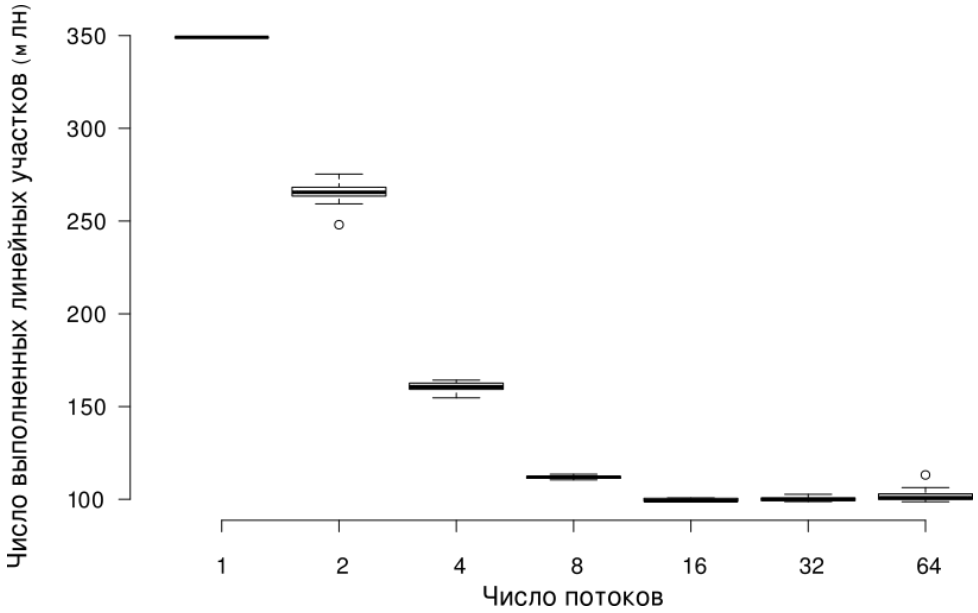
Тестирование

- Задача с тройным вложенным циклом
- Занимает большой объем памяти и



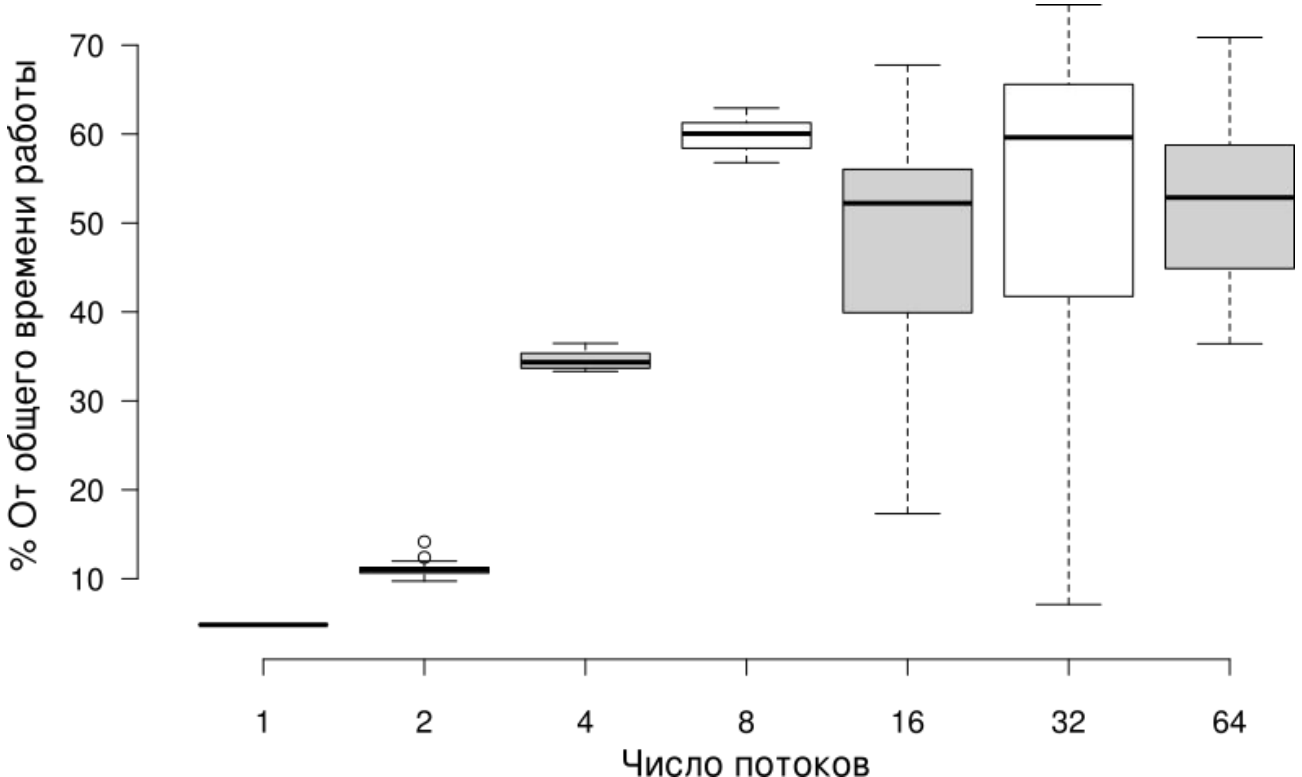
Результаты тестирования



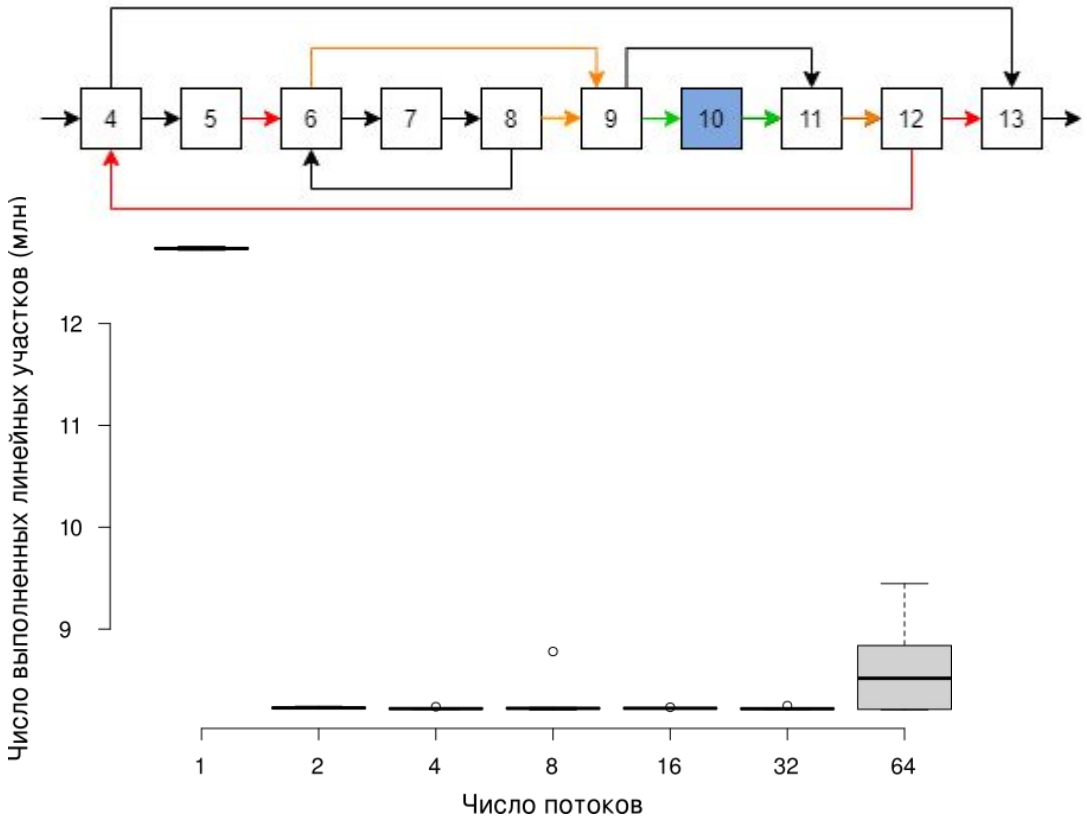


Результаты тестирования

Нахождение защищаемого участка в раскодированном виде

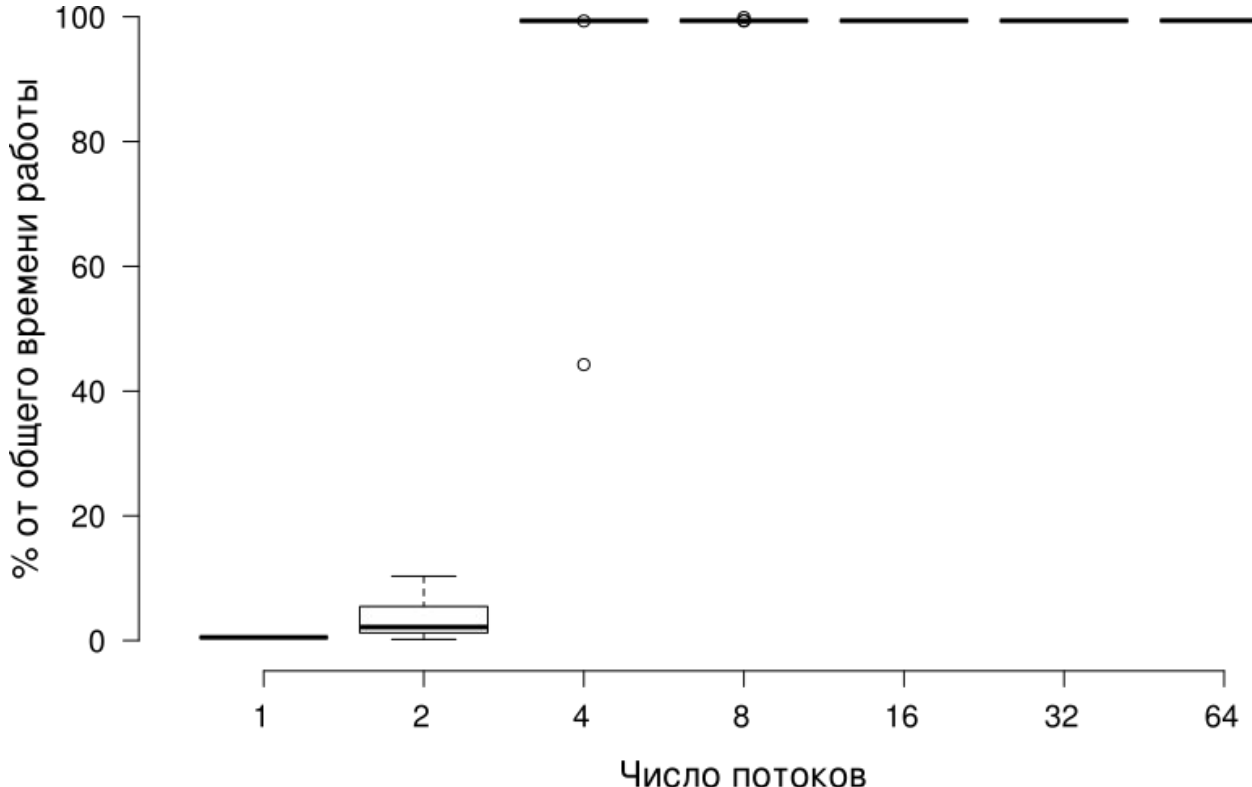


Результаты тестирования

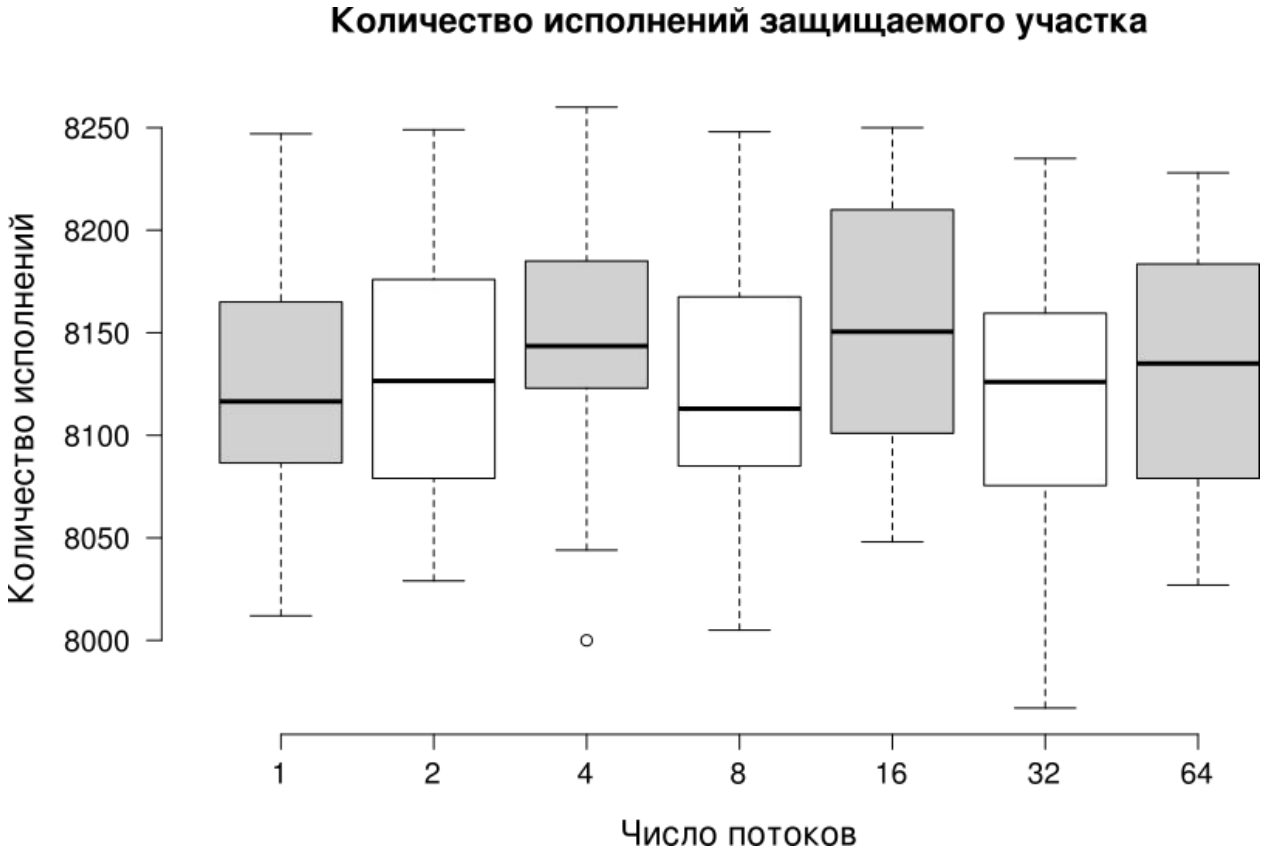


Результаты тестирования

Нахождение защищаемого участка в раскодированном виде



Результаты тестирования



Результаты

- Изучена классификация механизмов неблокирующей синхронизации
- Разработана архитектура программного модуля, решающего задачу автоматического встраивания механизмов синхронизации
- Реализован прототип данного модуля
- Проведено тестирование прототипа и исследование результатов, показавшее возможные уязвимости при применении используемого метода синхронизации.