

**Министерство науки и высшего образования Российской Федерации**  
**Федеральное государственное автономное образовательное учреждение высшего**  
**образования «Казанский (Приволжский) Федеральный Университет»**

Институт вычислительной математики и информационных технологий

Кафедра системного анализа и информационных технологий

Направление подготовки: 02.03.02 – Фундаментальная информатика и  
информационные технологии

Профиль: Системный анализ и информационные технологии

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**СИСТЕМА ФАЗЗИНГА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**  
**НА ОСНОВЕ ЭВОЛЮЦИОННОГО ПОДХОДА**

Обучающийся 4 курса  
группы 09-931

(Редькин В.С.)

Руководитель  
ст. преподаватель

(Долгов Д.А.)

Заведующий кафедрой системного анализа  
и информационных технологий  
д-р техн. наук, профессор

(Латыпов Р.Х.)

## **ОГЛАВЛЕНИЕ**

## **ВВЕДЕНИЕ**

Текст введения пишется вот здесь. Потом самое главное не забыть заменить этот образцовый текст перед демонстрацией научруку.

Абзац 2. Здесь начинается второй абзац введения. Потом самое главное не забыть заменить этот образцовый текст перед демонстрацией научруку.

1. вот так вот

2. будут выглядеть элементы списка

- а вот такими станут

- элементы маркированного списка. При этом стоит заметить, что текст по-прежнему выровнен по ширине и сохраняет отступы как в обычных абзацах

## **1. Трассировка**

Важным компонентом, значительно ускоряющим процесс фаззинга, является измерение покрытия кода программы при запуске очередного тестового примера. Существует несколько подходов для измерения покрытия:

### **1.1. Статическая инструментация**

Статическая инструментация программы, полагающаяся на применение специальных библиотек и компиляторов (например, afl-gcc), добавляющих в программу инструкции, на которые затем ориентируется фаззер для точного выяснения траектории выполнения программы.

Плюсом такого подхода является быстрота проведения фаззинга (например, в программе может быть искусственно выделена та или иная секция, подвергаемая тестированию в бесконечном цикле, за счёт чего отпадает необходимость в трате ресурсов на постоянный запуск новых процессов и загрузки библиотек).

Минус данного подхода состоит в необходимости наличия доступа к исходному коду программы и необходимости дополнительной работы с ним.

### **1.2. Динамическая инструментация**

Динамическая инструментация программы полагается на использование методов, схожих с таковыми, применяемыми в отладчиках - для сбора информации о траектории выполнения программы применяются точки останова, в которых записывается состояние регистра счётчика команд. В отличие от предыдущего подхода, в данном случае возможна работа с уже готовым исполняемым файлом, мы можем и не иметь исходного кода исследуемой программы.

Проблемой динамической инструментации является серьёзное влияние на скорость выполнения программы.

Для снижения этого влияния могут применяться различные методы, например Coverage-guided tracing.

## **ЗАКЛЮЧЕНИЕ**

Текст нашего умного заключения будет написан вот тут.