Композиционное моделирование окружения для верификации программ на GNU C

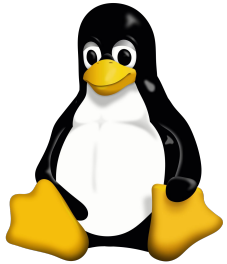
Илья Захаров, Евгений Новиков (ИСП РАН)

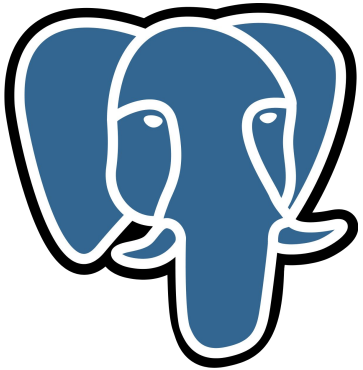
ОТКРЫТАЯ КОНФЕРЕНЦИЯ ИСП РАН ИМ. В.П. ИВАННИКОВА

МОСКВА, 22-23 НОЯБРЯ 2018 Г.

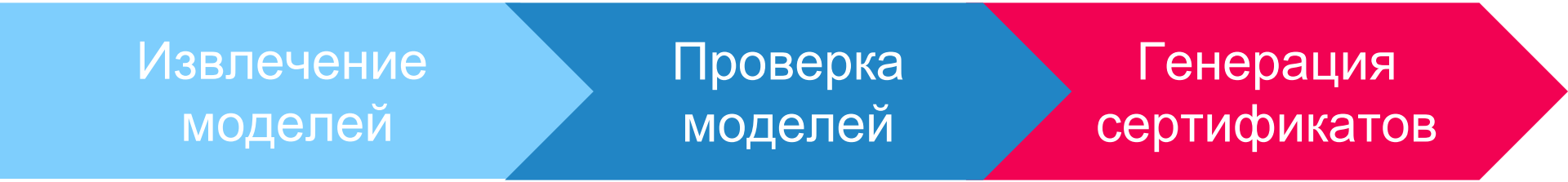
GNU C программы

- Надежность
- Безопасность
- Производительность





Статическая верификация



Фронтенды для Java, C, LLVM bytecode, Boogie

Методы BMC, CEGAR

Ошибочные пути, отчеты о покрытии, тесты

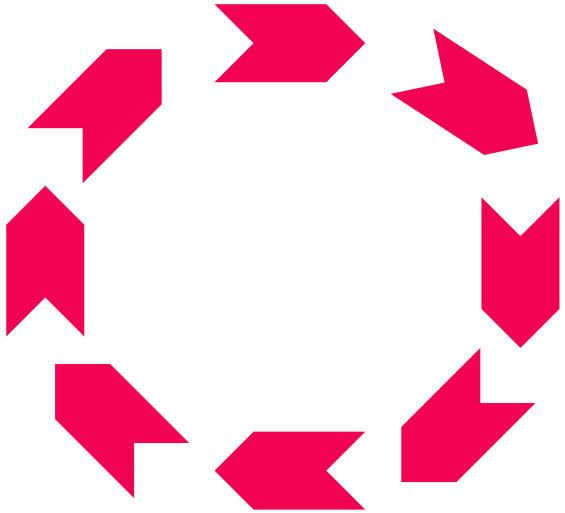
Свойства инструментов статической верификации

- ✓ Консервативный анализ
- ✓ Доказательства и сертификаты
- ✓ Автоматическая работа

- ✗ Программы 10-100 kloc
- ✗ Модели окружения
- ✗ Высокая трудоемкость применения

Процесс статической верификации

Декомпозиция программы



Формализация требований

Моделирование окружения

Конфигурирование инструментов

Запуск инструментов

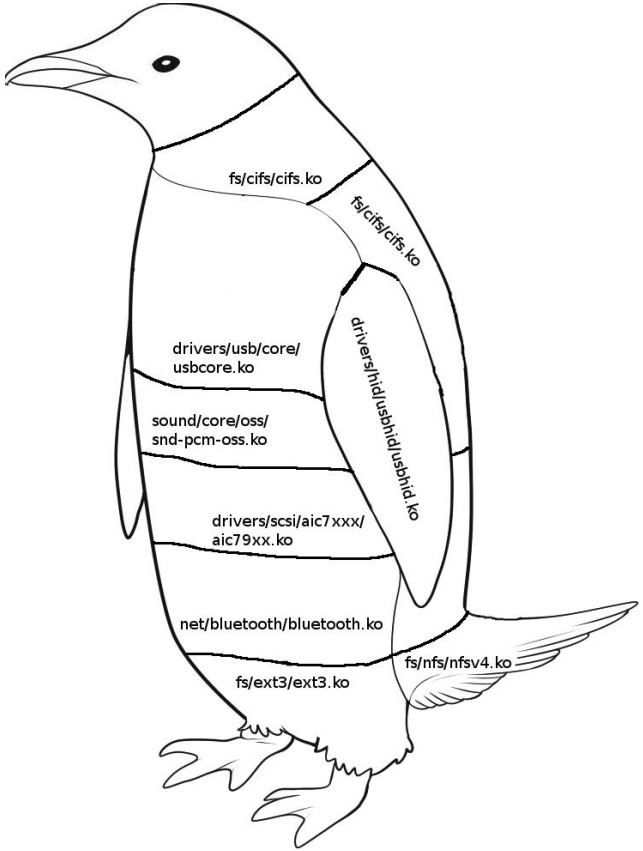
Анализ результатов

Klever – система статической верификации GNU C программ

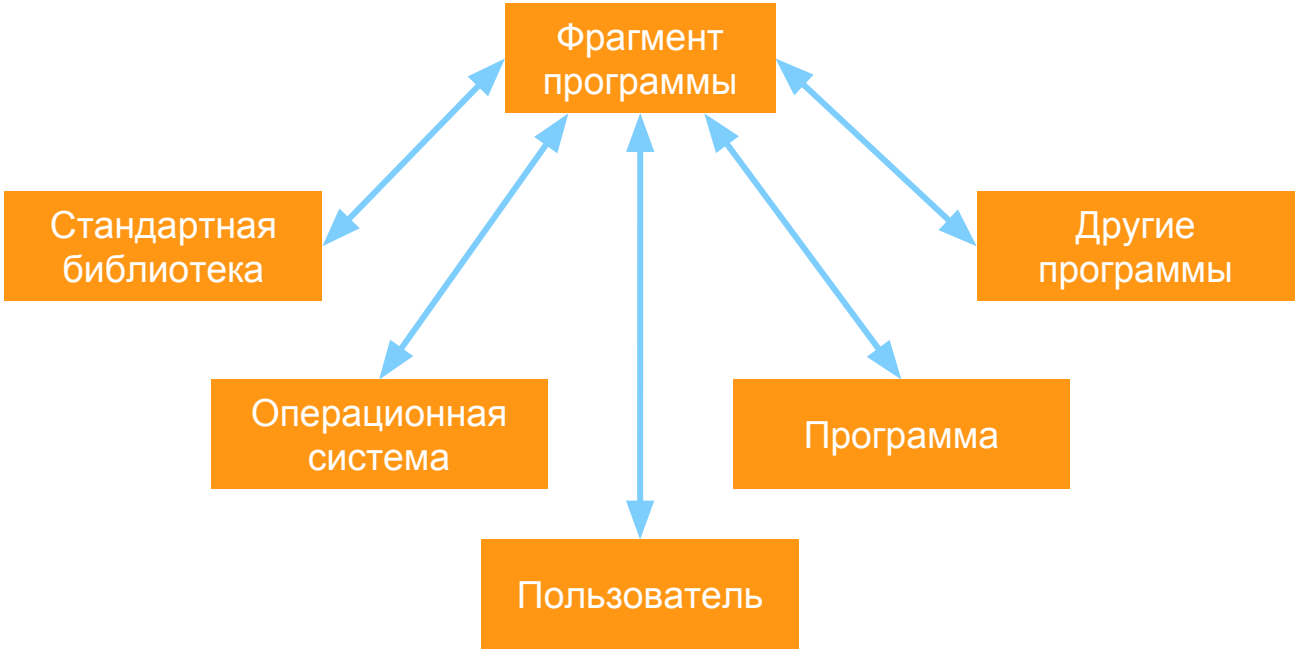
Применение

- ✓ Программные системы с повышенными требованиями к надежности и безопасности
- ✓ Программные системы большого размера
- ✓ Доказательство выполнения требований при некоторых предположениях
- ✓ Поиск ошибок, которые не удается найти другими методами

Декомпозиция исходного кода



Окружение фрагмента программы



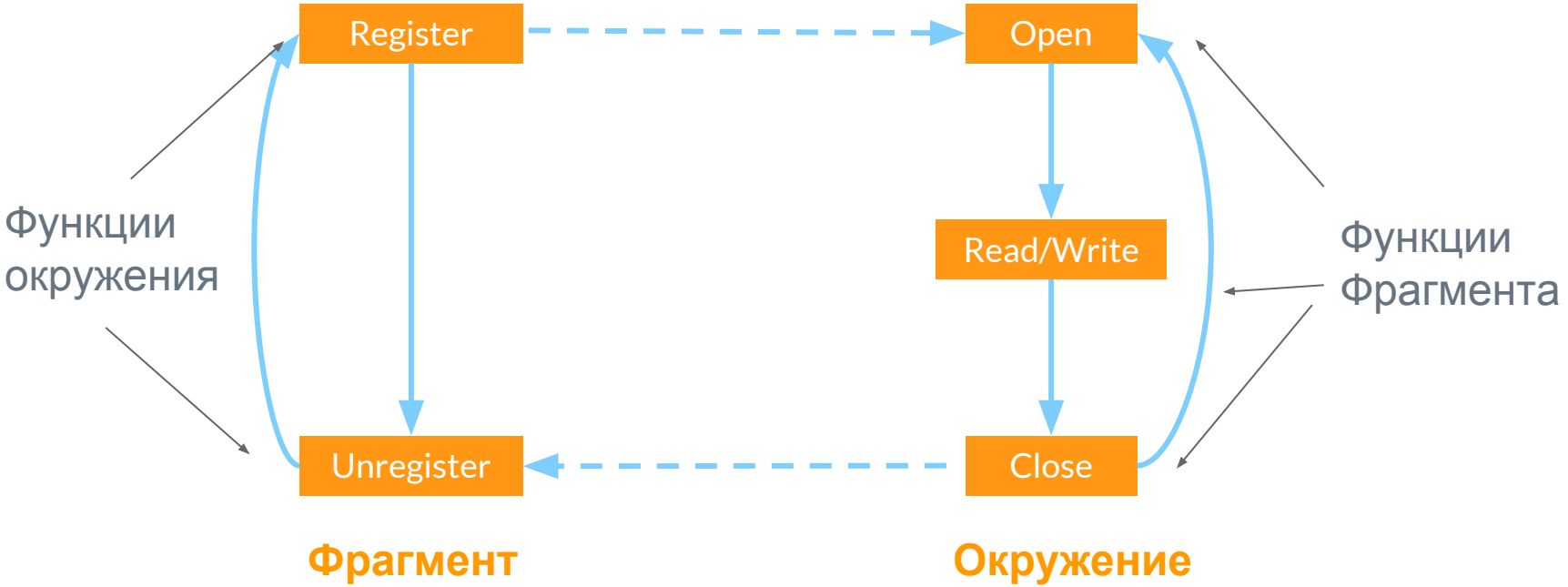
Модель окружения

Модель окружения

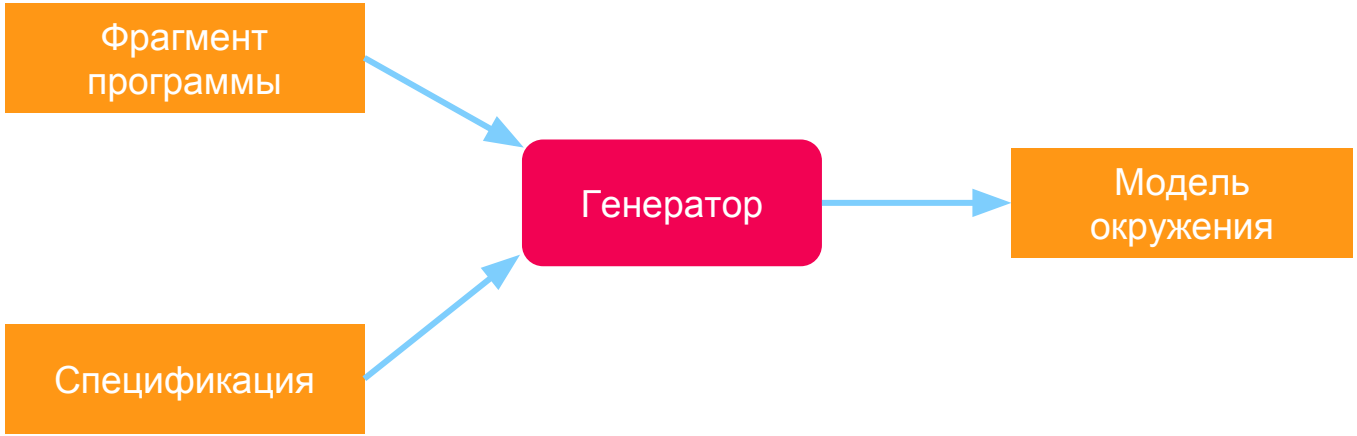
Примеры событий

- Вызов функций
  - из фрагмента окружением
  - окружением из фрагмента
- Изменение данных
  - фрагмента окружением
  - окружения фрагментом

Моделирование сценариев



Моделирование окружения



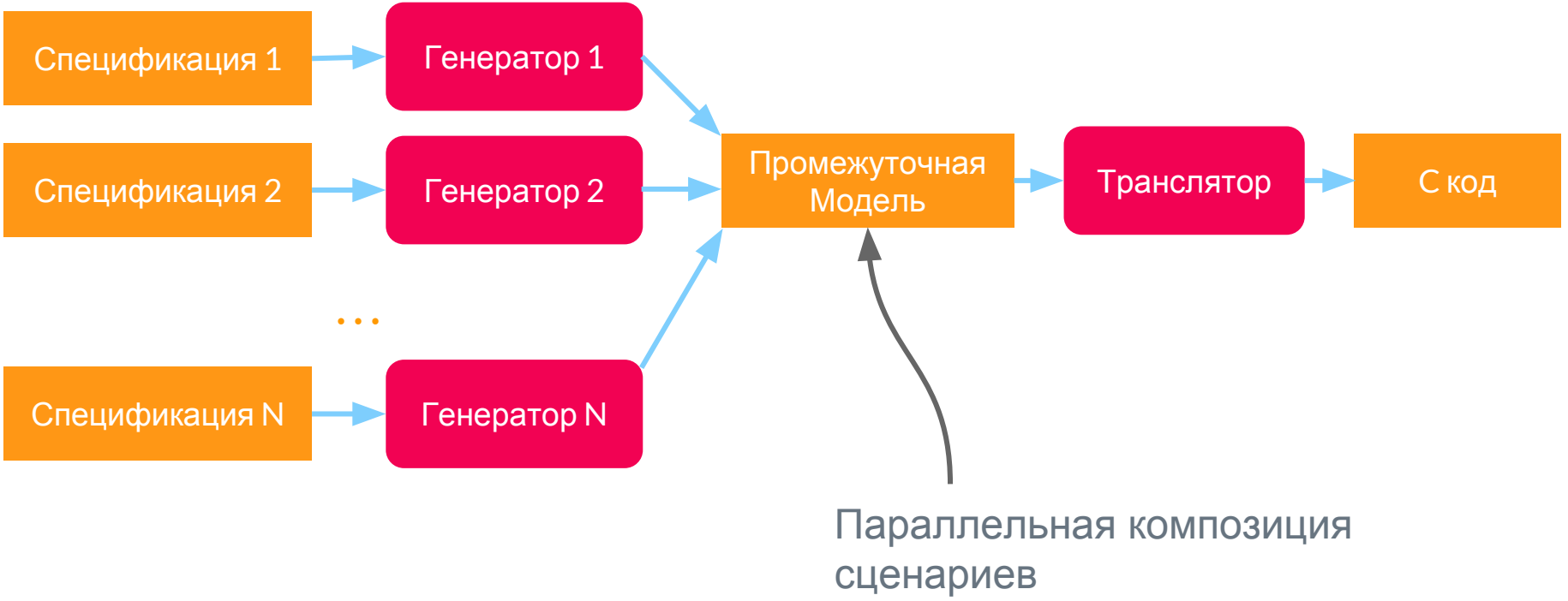
Генерация модели окружения

- Спецификация отделена от кода
- Адаптация к инструменту верификации
- Язык спецификации проще GNU C
- Проще переиспользовать части модели для разных фрагментов

Использование одного языка для описания сценариев

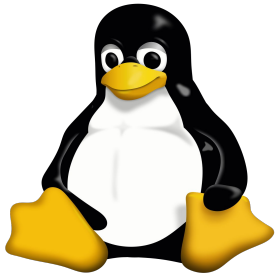
- ✗ Сценарии имеют разную сложность моделирования
- ✗ Перегруженность синтаксиса

Моделирование окружения



Практические результаты

- Драйверы устройств Linux
- Подсистемы монолитного ядра Linux
- Апплеты BusyBox



Реализация метода

- ✓ 3 генератора сценариев для модулей и подсистем ОС Linux
- ✓ 2 генератора для других программ, включая BusyBox
- ✓ Транслятор для генерации параллельного и последовательного GNU C кода

Практические результаты

Проект	Фрагменты	Покрытие (строки)	Покрытие (функции)	Ошибки
Драйверы (Linux 3.14)	3619	45%	31%	30%
Подсистемы (Linux 3.14)	357	55%	45%	43%
BusyBox 1.28.3	268	93%	86%	20%

Исправленные ошибки в ядре Linux: http://linuxtesting.org/results/ldv

Спасибо за внимание