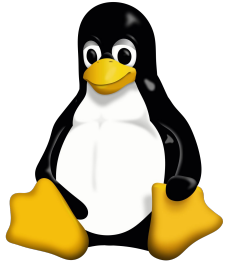
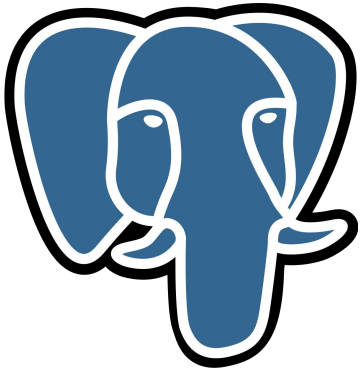
Статическая верификация системного программного обеспечения на языке Си

Илья Захаров
ИСП РАН
ilja.zakharov@ispras.ru

Системное ПО

- Надежность
- Безопасность
- Производительность





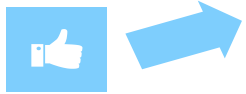
OpenSSL

Обеспечение качества

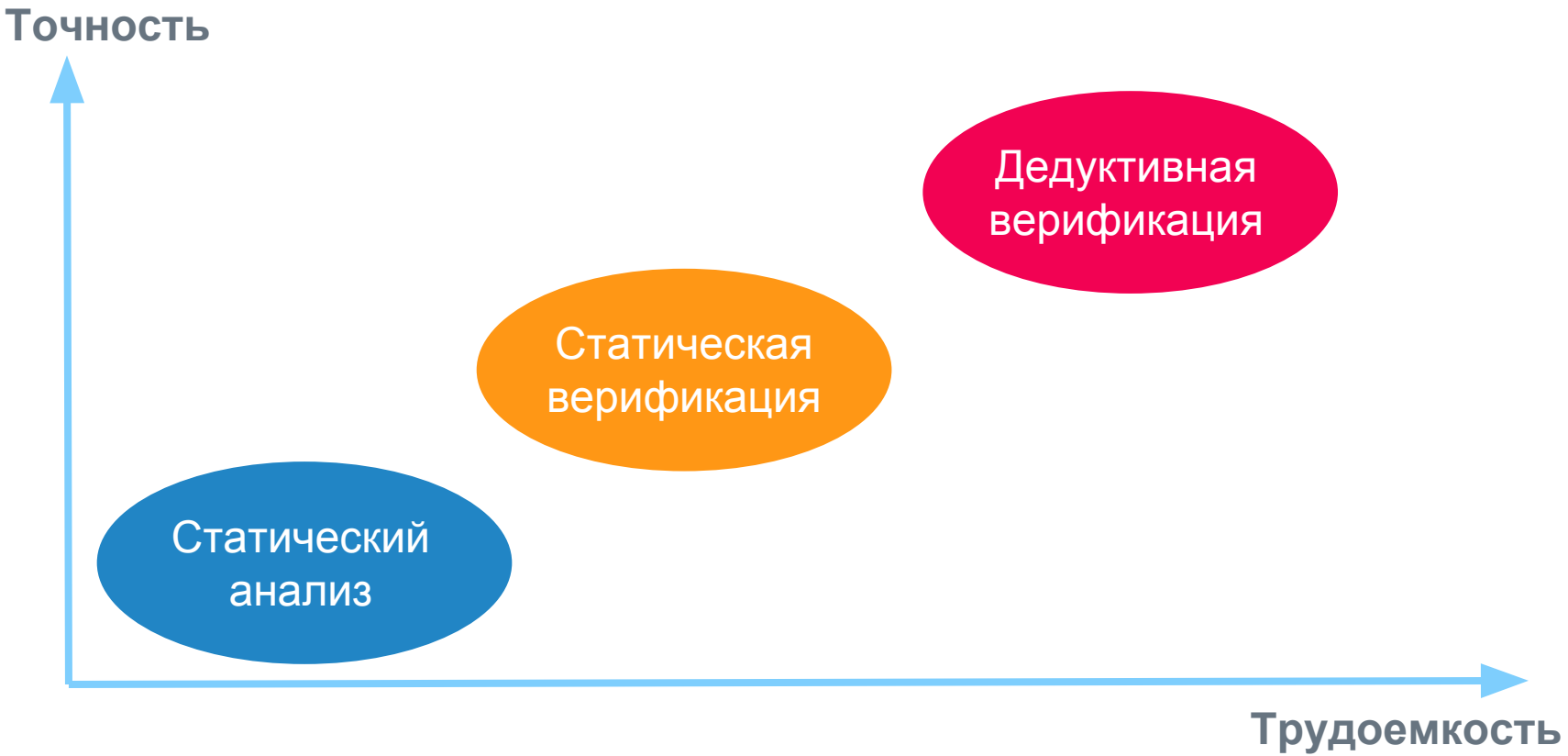
Динамические методы

Статические методы

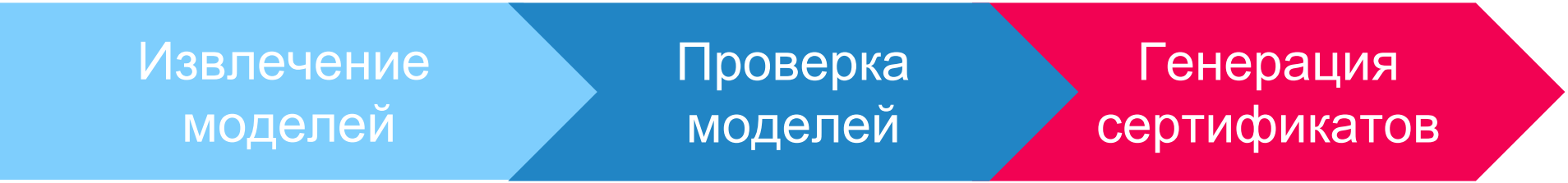
Экспертиза



Статические методы



Статическая верификация

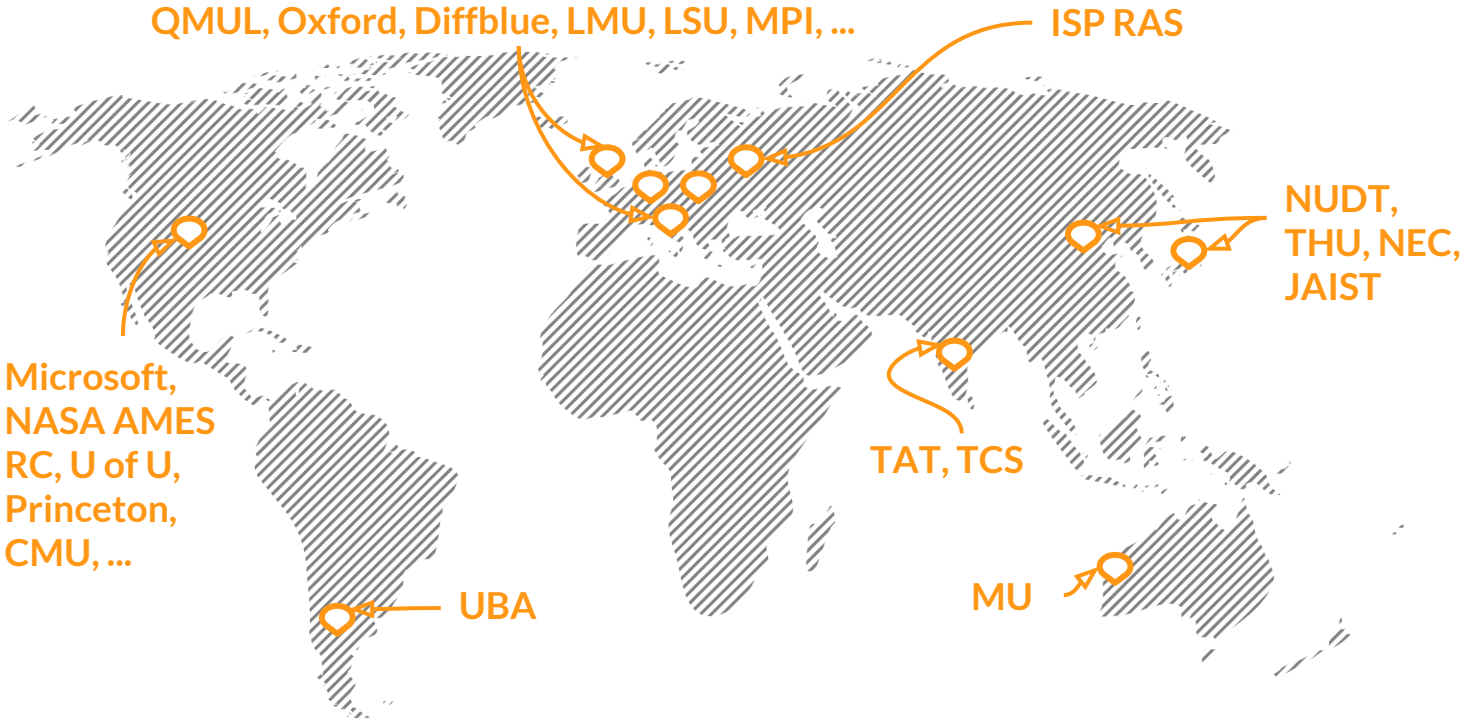


Фронтенды для Java, C

Методы BMC, CEGAR

Ошибочные пути, отчеты о покрытии, тесты

Разработка инструментов статической верификации



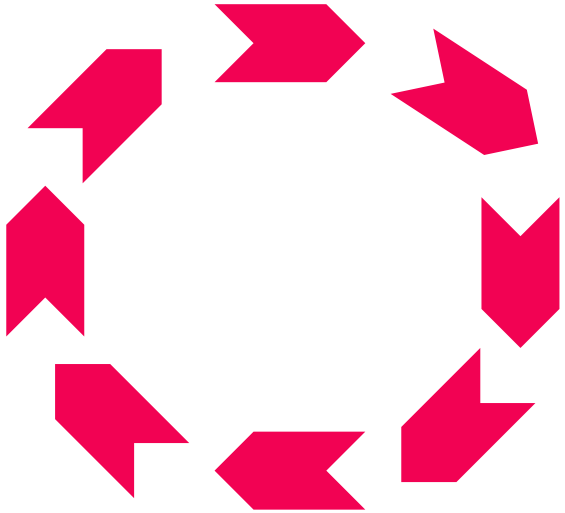
Свойства инструментов статической верификации

- ✓ Высокая точность анализа
- ✓ Генерируются сертификаты
- ✓ Автоматическая работа

- ✗ Требуются модели окружения
- ✗ Сложно анализировать результаты
- ✗ Сложно проверять более 10-100 тыс. строк кода

Процесс статической верификации

Декомпозиция исходного кода



Формализация требований

Моделирование окружения

Конфигурирование инструментов

Запуск инструментов

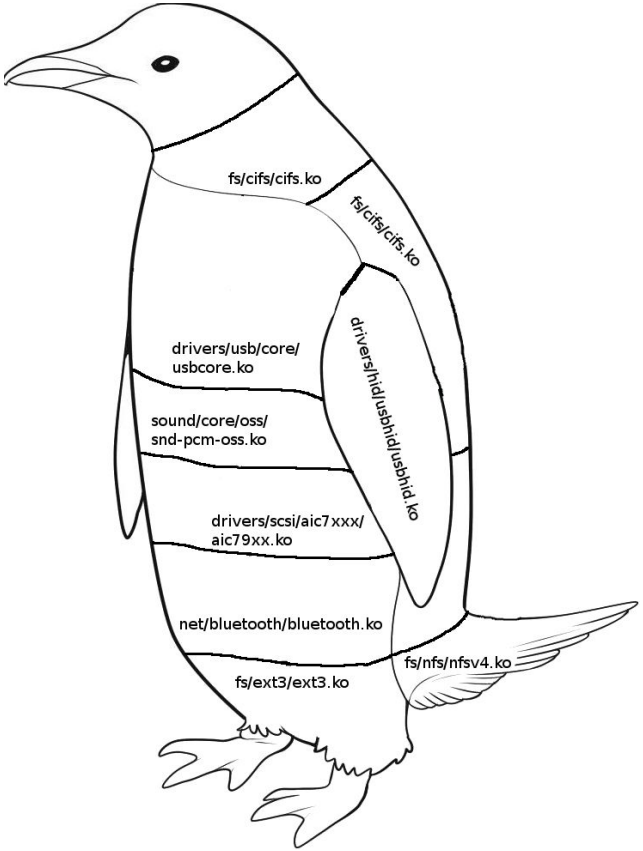
Анализ результатов

Системы статической верификации

- SDV
  Драйверы Microsoft Windows
- Avinux, DDVerify, LDV Tools
  Драйверы Linux
- DC2, embeddr
  Встраиваемое ПО

Klever – система статической верификации системного программного обеспечения

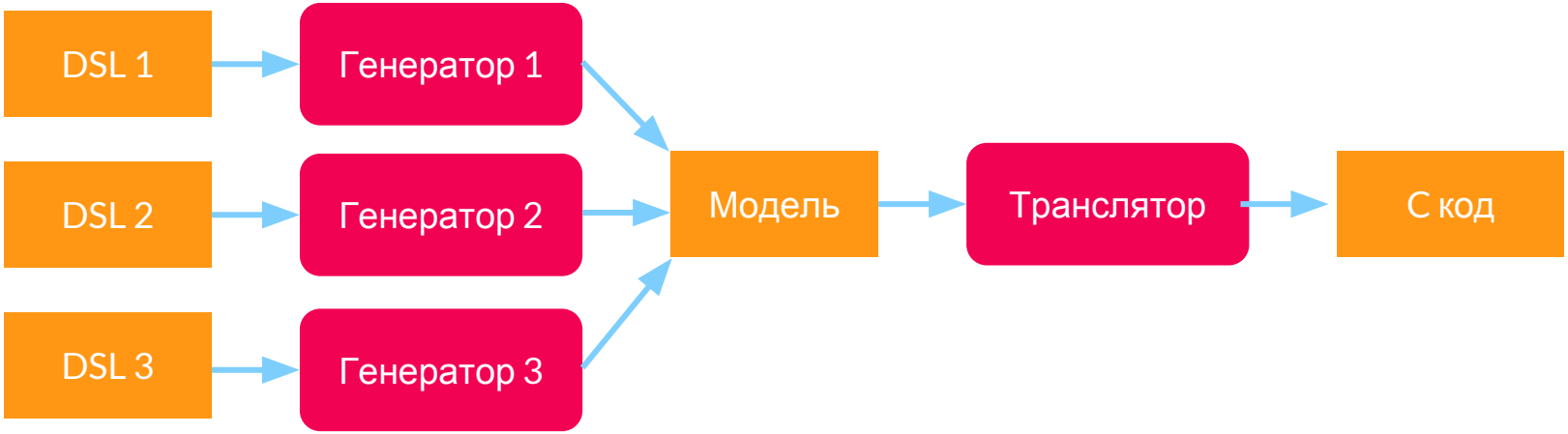
Декомпозиция исходного кода



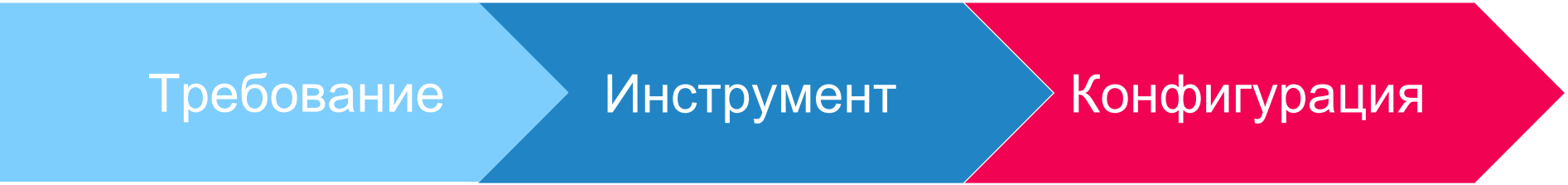
Формализация требований

- **Корректность работы с памятью**
- **Корректность работы с целочисленными типами**
- **Завершимость**
- **Отсутствие гонок в многопоточной программе**
- **Недостижимость ошибочного оператора**

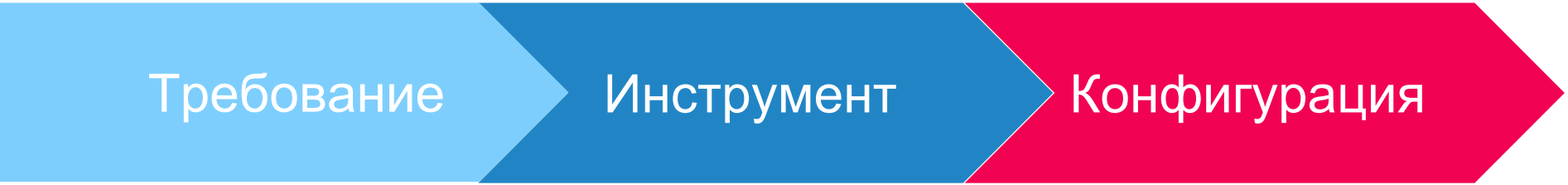
Моделирование окружения



Конфигурирование инструментов



Конфигурирование инструментов



Корректность работы с API

CPAchecker

CEGAR, предикатный анализ, битовая точность

Запуск инструментов

- Ограничение потребления вычислительных ресурсов
- Распределенное или локальное выполнение

Анализ результатов

Анализ ошибочных путей

Анализ покрытия

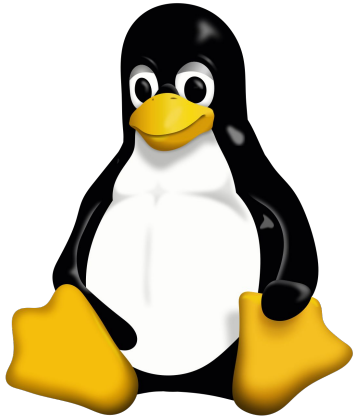
Накопление результатов

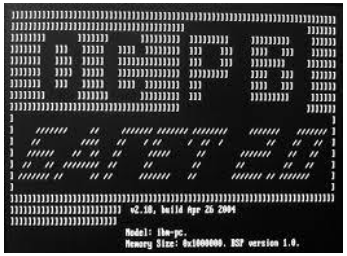
Сравнение результатов

Применение

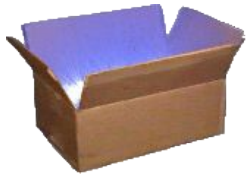
- ✓ Верификация и поиск ошибок
- ✓ Поиск уязвимостей
- ✓ Сертификация
- ✗ Разработка

Проекты

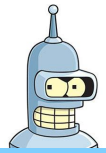




TIZEN™



NGINX



Пример ошибки

```
static int samsung_i2s_remove(struct platform_device *pdev)
{
-       struct i2s_dai *pri_dai, *sec_dai;
+       struct i2s_dai *pri_dai;
        pri_dai = dev_get_drvdata(&pdev->dev);
-       sec_dai = pri_dai->sec_dai;
-
-       pri_dai->sec_dai = NULL;
-       sec_dai->pri_dai = NULL;
```

Пример ошибки

→ static int samsung_i2s_probe(struct platform_device *pdev)

→ static int samsung_i2s_remove(struct platform_device *pdev)

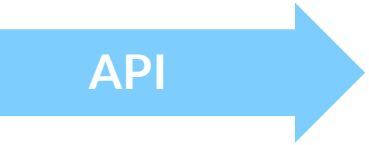
Пример ошибки

```
static int samsung_i2s_probe(...)
{
   struct i2s_dai *pri_dai, *sec_dai = NULL;
   ...
   pri_dai = i2s_alloc_dai(pdev, false);
   ...
   if (quirks & QUIRK_SEC_DAI) {
      sec_dai = i2s_alloc_dai(pdev, true);
      ...
      sec_dai->pri_dai = pri_dai;
      pri_dai->sec_dai = sec_dai;
      ...
  dev_set_drvdata(&pdev->dev, pri_dai);
```

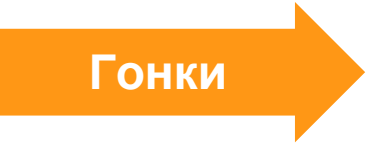
```
static struct i2s_dai *i2s_alloc_dai(...)
{
   ...
   i2s = devm_kzalloc(...);
   ...
   i2s->pdev = pdev;
   i2s->pri_dai = NULL;
   i2s->sec_dai = NULL;
```

Спасибо за внимание

Доля истинных ошибок



60%



30%

10%