2014, Москва

LinuxTesting.org

Методы модульной верификации ядра ОС Linux.

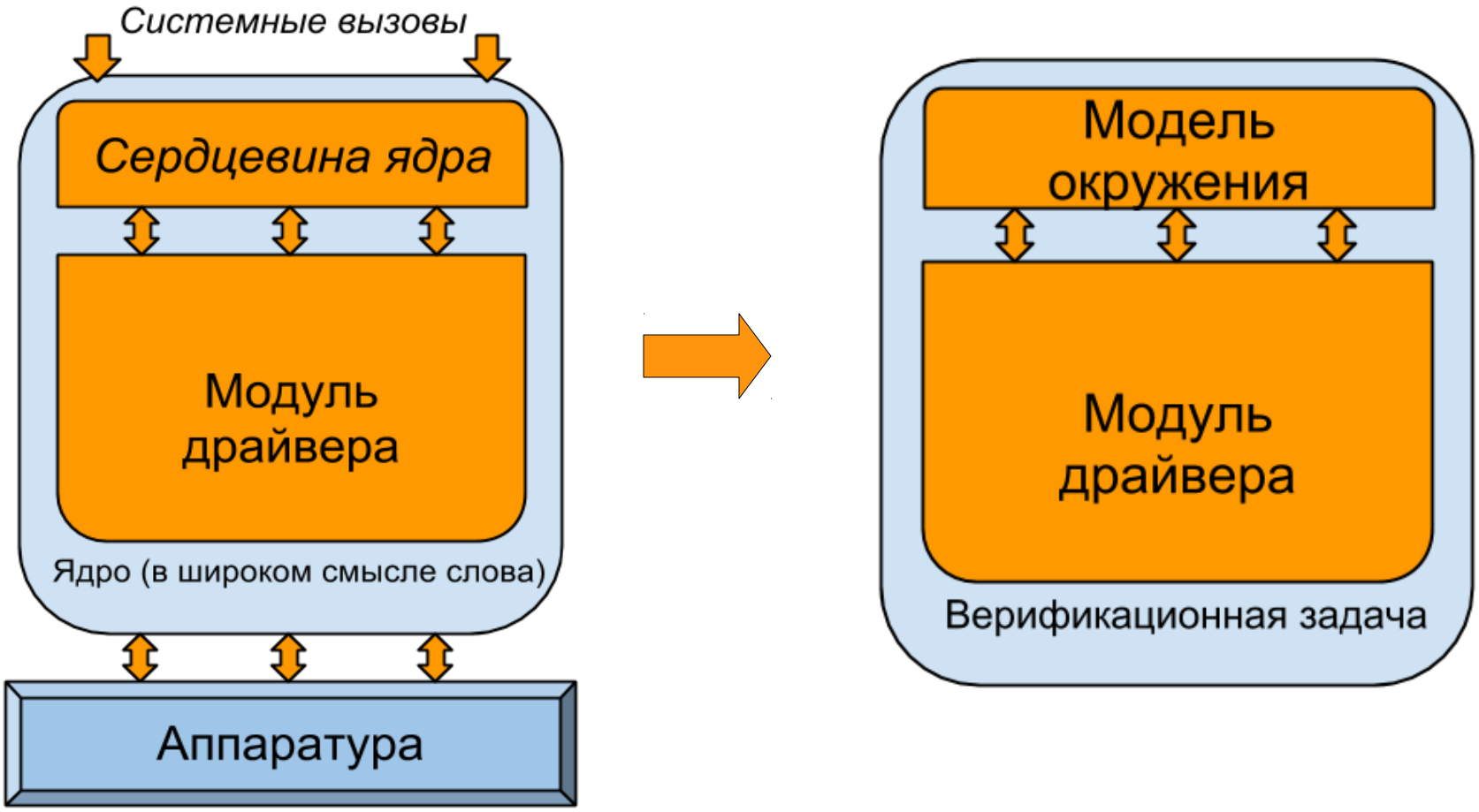
Илья Захаров 875 гр. ФУПМ
ilja.zakharov@ispras.ru

Научный руководитель:
д. ф-м. н., профессор Петренко А. К.

ISP RAS
Institute for System Programming of the Russian Academy of Sciences

Инструменты статической верификации Си программ, основанные на BMC и CEGAR

Модель окружения драйвера



Модель окружения модуля

- Модель взаимодействия с сердцевиной ядра
- Модель взаимодействия модулей ядра
- Модель взаимодействия с аппаратурой

Существующие подходы

- Microsoft SDV: аннотации, лимитированный набор поддерживаемых библиотек, размер ~ 100к строк
- DDVerify: вручную написанная модель для небольшого числа типов групп обработчиков на Си, подменяющая код библиотек
- Avinux: модель для отдельных обработчиков, большое внимание уделено инициализации параметров обработчиков
- LDV: модель на основе ручных спецификаций и спецификаций шаблонов групп обработчиков, можно проверять произвольный модуль

Преимущества и недостатки существующих подходов

	Модель взаимодействия с ядром	Модель взаимодействия между модулями	Простота сопровождения
DDverify	+/-	-	-
Avinux	-	+	-
LDV	+	-	+

Результаты проверки реальных ошибок

	Количество ошибок	Процент от общего количества ошибок
Найдено LDV	15	44 %
Ошибки в окружении	4	12 %
Нужна модель взаимодействия модулей	4	12 %
Проблемы инструментах верификации и правилах	11	32 %
Итого	34	100 %

Цель

Реализовать поддержку моделирования взаимодействия между модулями ядра ОС Linux рамках метода моделирования окружения в системе LDV

Пример драйвера

```
int usbpn_probe(struct usb_interface *intf, const struct usb_device_id *id){
    ...
}
static void usbpn_disconnect(struct usb_interface *intf){
    ...
}

static struct usb_driver usbpn_struct = {
    .name =         "ldv-test",
    .probe =        usbpn_probe,
    .disconnect =   usbpn_disconnect,
};

int __init usbpn_init(void){
    return usb_register(&usbpn_struct);
}

void __exit usbpn_exit(void){
    usb_deregister(&usbpn_struct );
}
```

Обработчики

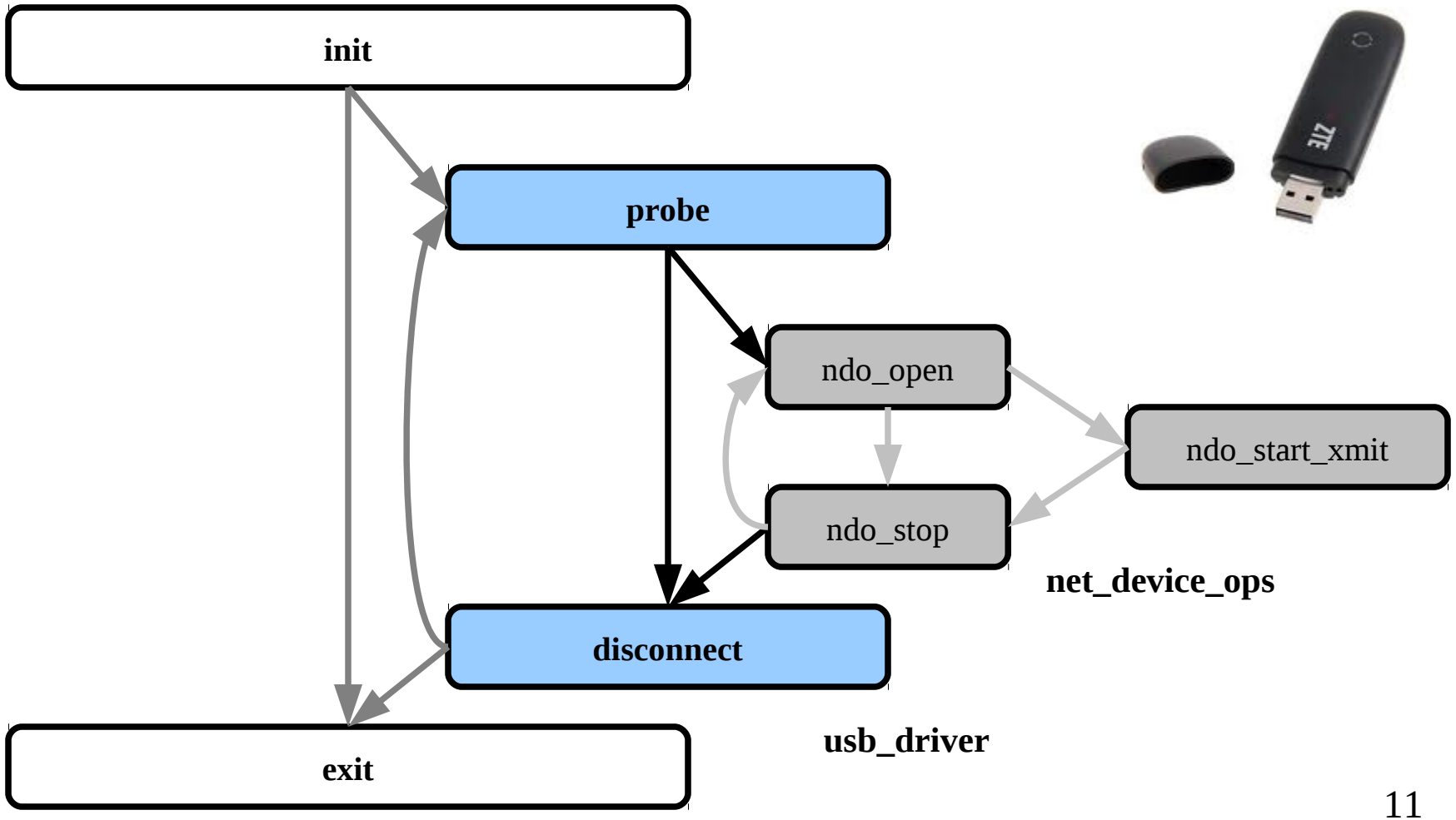
Функции инициализации и выхода

Функции регистрации и дерегистрации обработчиков

Группа обработчиков

- Включает: регистрация, обработчики, дерегистрация
- Ограничения: на параметры, на порядок вызова обработчиков, на контекст

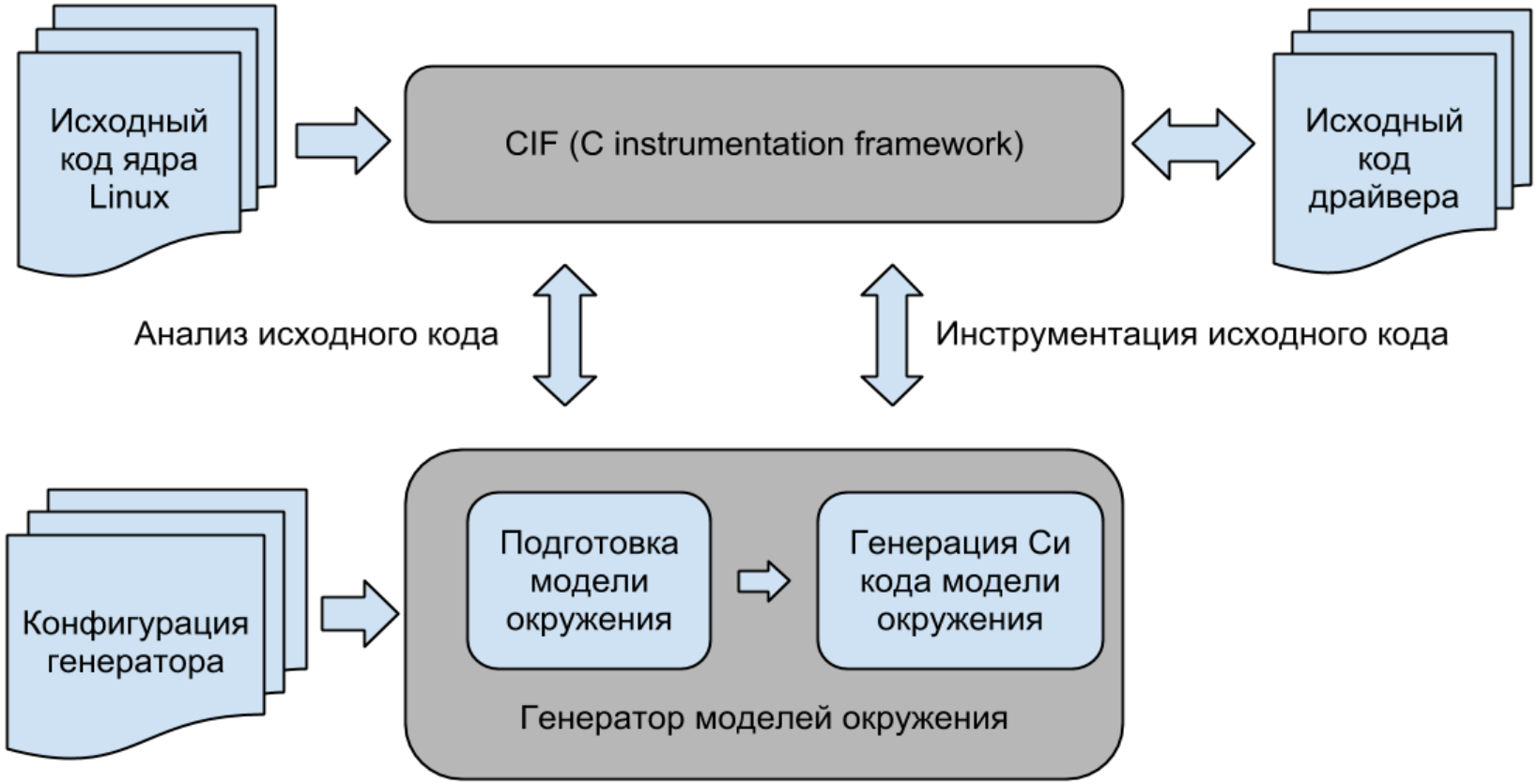
Совместная работа нескольких групп обработчиков



Метод построения модели в LDV

- Анализ исходного кода для извлечения групп обработчиков
- Сопоставление спецификаций и групп обработчиков
- Построение модели для конкретного модуля как параллельной композиции моделей отдельных групп обработчиков
- Трансляция модели в Си код

Генерация модели окружения в системе LDV



Достоинства метода

- **Применим для верификации практически всех динамически загружаемых модулей ядра**
- Высокая степень покрытия интерфейсов
- Точность моделирования позволяет успешно применять CEGAR инструменты
- Модель расширяема

Недостатки метода

- Нет возможности проверять библиотечные модули, предоставляющих обработчики другим модулям
- Слишком трудно специфицировать “специфичные” для подсистем группы обработчиков

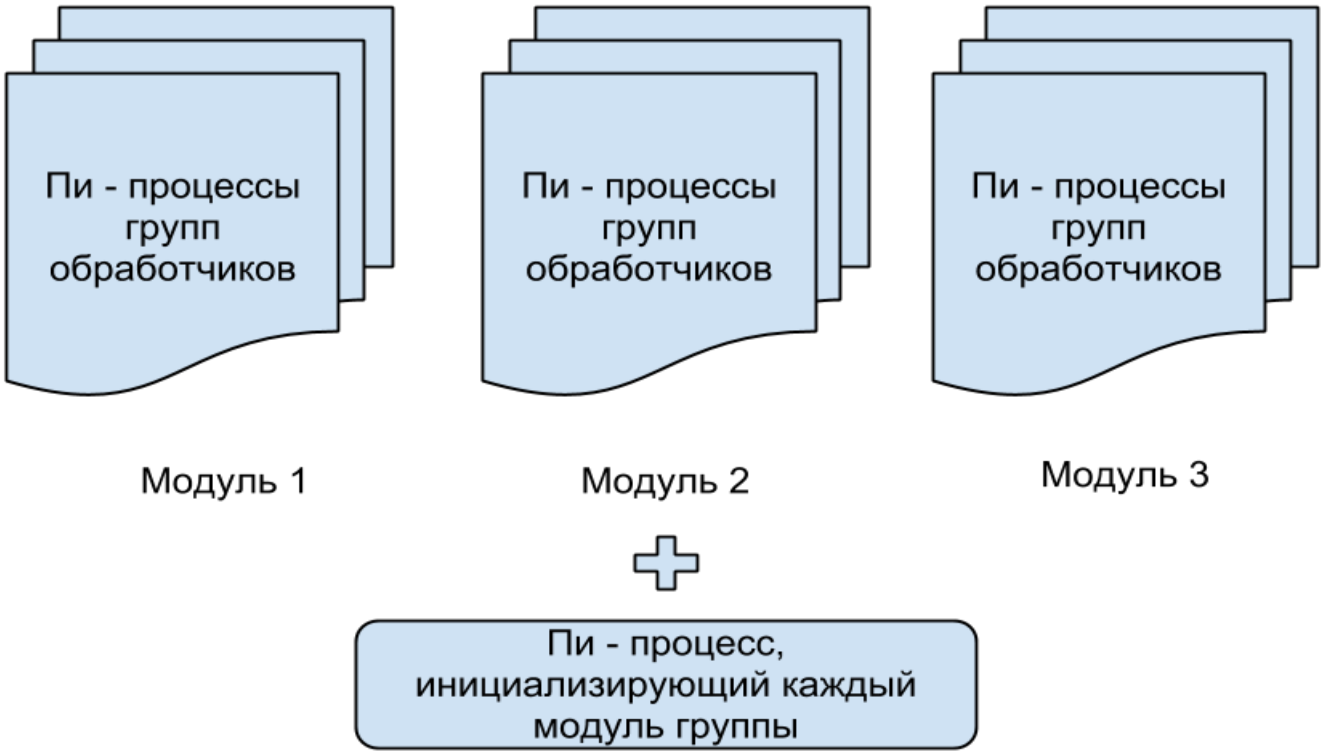
Результаты применения спецификаций

	Группы обработчиков	Типы групп обработчиков
Спецификация для типа	818	8
Спецификация для шаблона	5678	297
Простейшая модель	9876	434
Итого	16372	739

Совместная верификация нескольких модулей

- Увеличение полноты модели (экспортируемые функции, регистрация в экспортируемых функциях)
- Сохранить применимость существующего метода для моделирования окружения, опирающегося на моделирование окружения модуля ядра.
- Замена части модели окружения на реальное окружение

Модель окружения для группы модулей



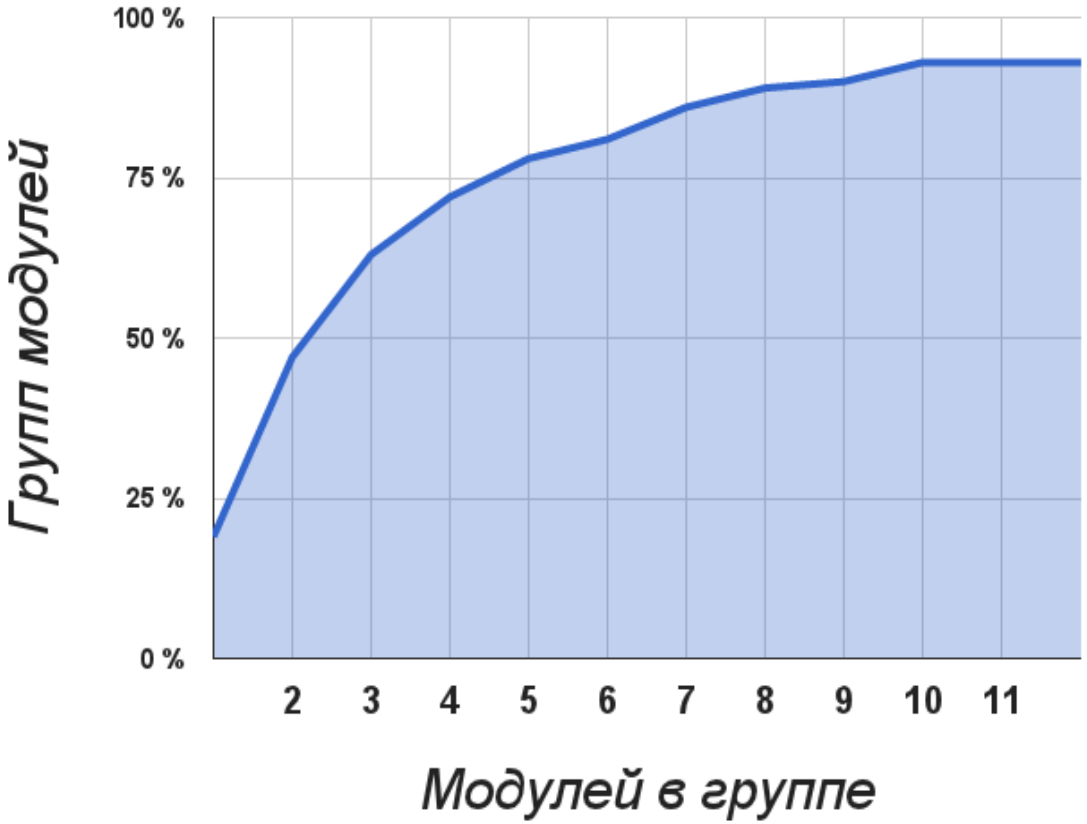
Как выбирать модули для верификационной задачи

1. Построение графа зависимостей модулей
2. Составить верификационные задачи фиксированного размера так, чтобы проверить все зависимости между модулями
3. Минимизировать число проверок одних и тех же модулей

Построение графа зависимостей

- Критерий - экспортируемые и импортируемые модулем ядра символы
- Извлечение зависимостей выполняется при помощи depmod
- Для каждого модуля рекурсивно определяются модули от которых он зависит

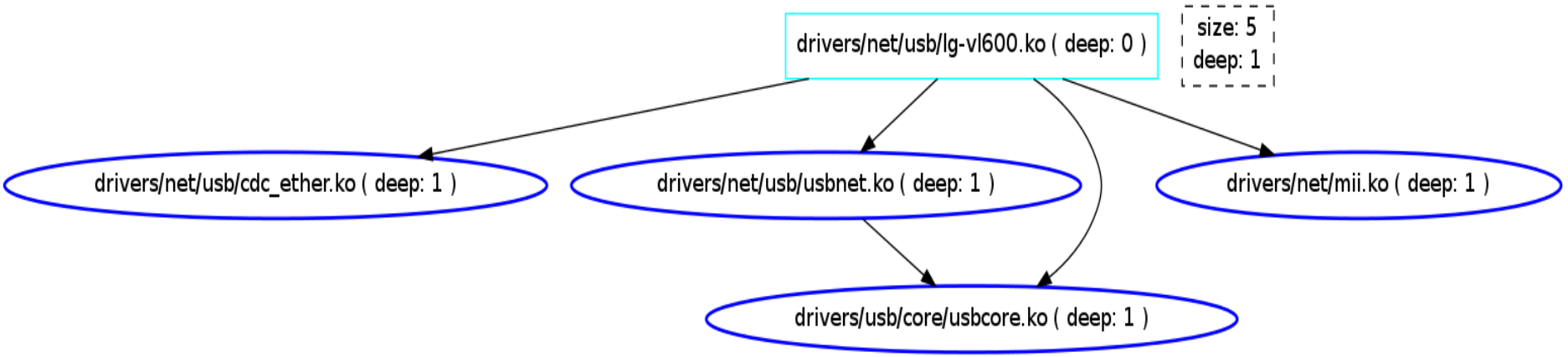
Размер групп модулей



Выделение верификационныз задач фиксированного размера

- Жадный алгоритм, отдающий приоритет непосредственно требуемым модулям
- Строим верифкационные задачи для “корневого” модуля, загружаемого в последнюю очередь

Пример дерева зависимостей модуля



Строгая постановка задачи

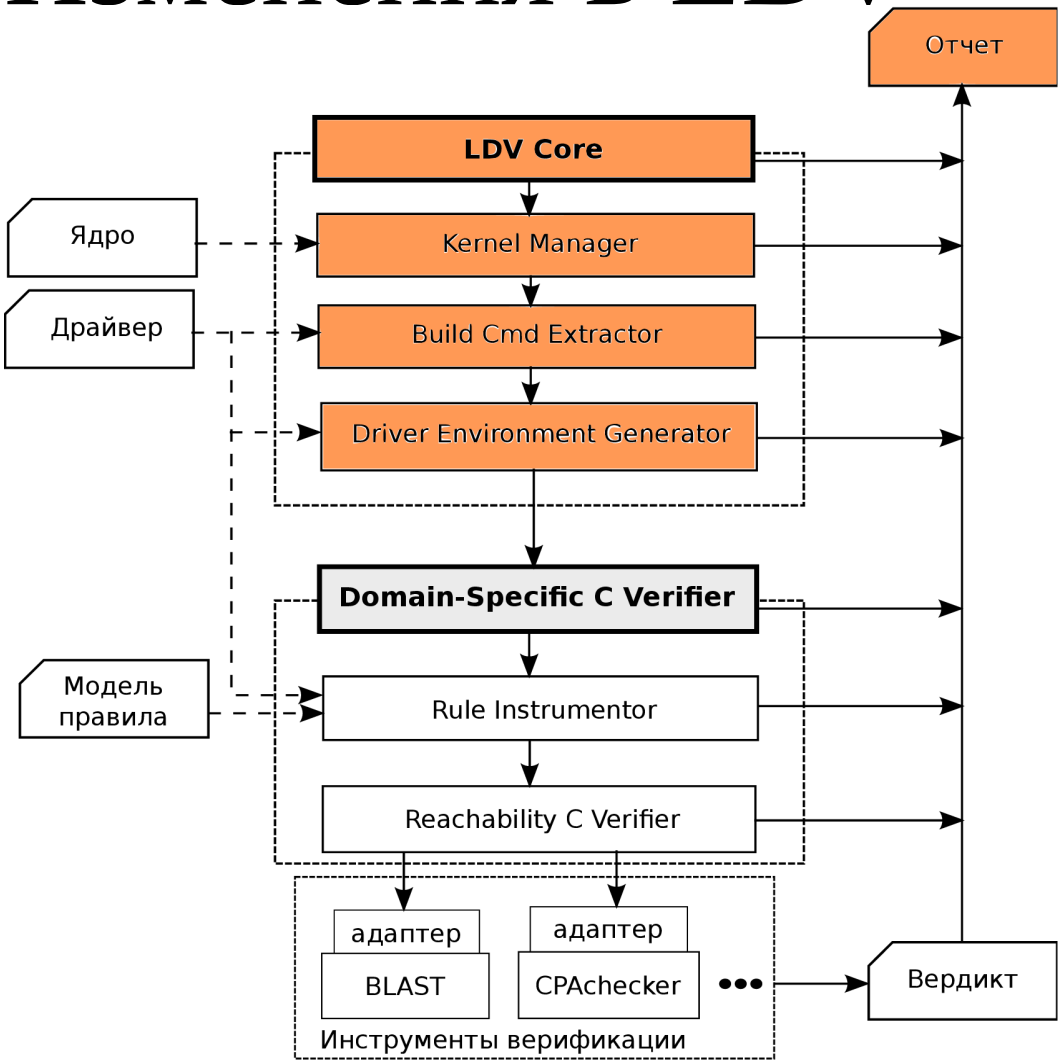
Выделить связные подграфы данного графа таким образом, чтобы

- каждое ребро входило в хотя бы один подграф
- число вхождений одной вершины в разные подграфы было минимальным
- размер каждого подграфа не должен превышать заданный размер
-

Метод с минимизацией проверок модулей

- Построение полного графа зависимостей модулей (считаем его неорентированным)
- Строим двойственный граф
- Решаем k-balanced graph partitioning
- Но: минимальный разрез не гарантирует минимального количества повторных проверок модулей из-за отображения вершин в несколько ребер при построении двойственного графа

Изменения в LDV



Результаты

- Разработана новая система выделение команд сборки
- Разработан способ построения верифкационных задач для совместной верификации нескольких модулей
- (почти) Разработан генератор моделей окружения для группы модулей

Направления дальнейшей работы

- Результаты экспериментов с верификационными задачами, состоящими из нескольких модулей
- Эксперименты по замене модели окружения на реальное окружение для некоторых проблемных групп обработчиков

LinuxTesting.org

Спасибо за внимание!

Ilja Zakharov
Ilja.zakharov@ispras.ru

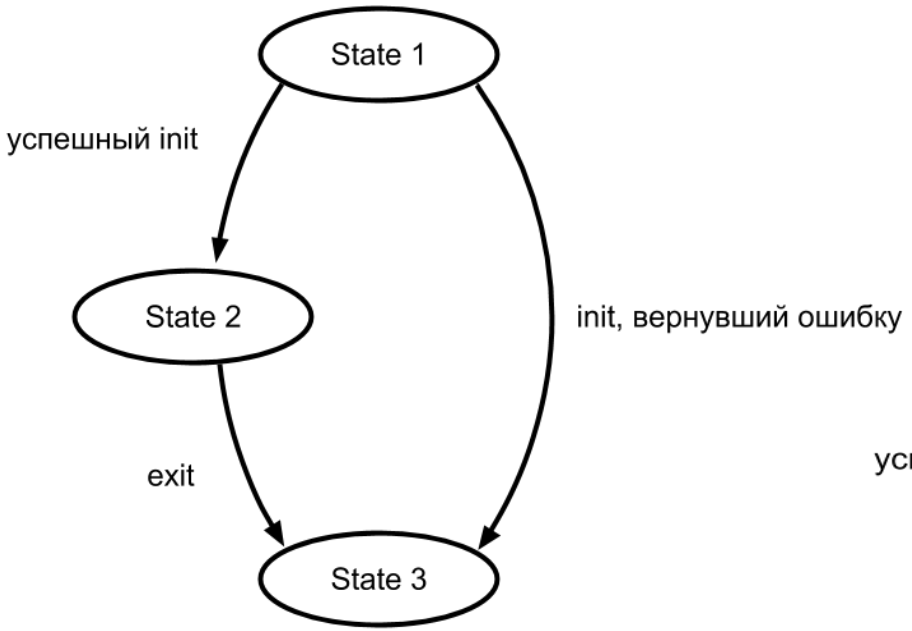
http://linuxtesting.org/project/ldv,
gsoc2013/glaurung/5001

ISP RAS
Institute for System Programming of the Russian Academy of Sciences

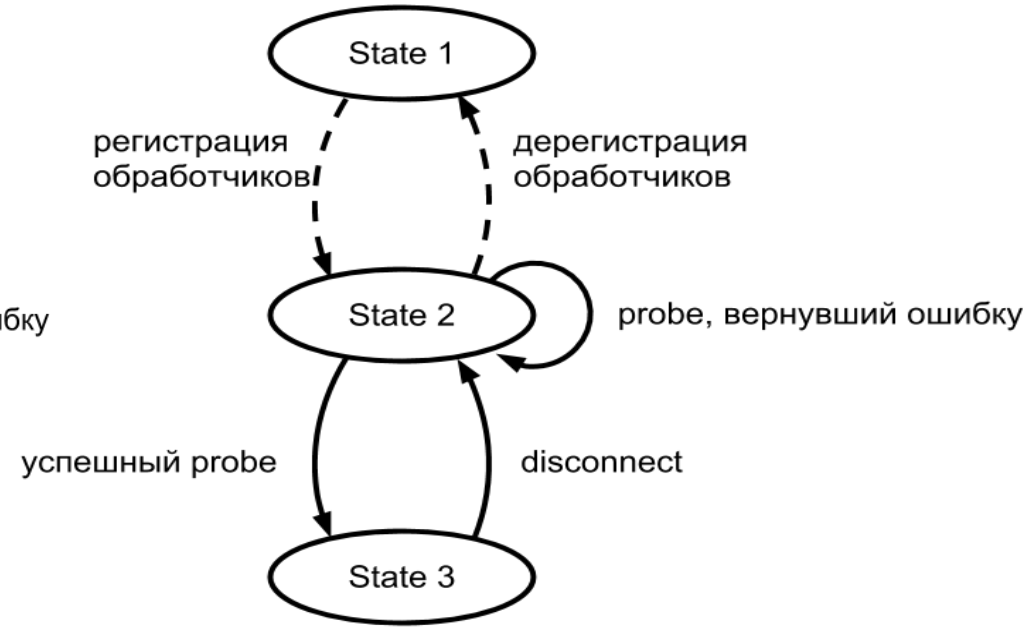
Проблемы в моделировании

- Передача данных через сердцевину ядра (при регистрации и передаче данных в другие группы обработчиков)
- Вызов по функциональным указателям обработчиков из других модулей

Представление модели в виде параллельной композиции Пи-процессов



Пи-процесс для функций init и exit



Пи-процесс для usb_driver