Метод автоматической адаптации спецификаций правил корректности

Щепетков Илья Викторович

Студент

Факультет ФУПМ МФТИ, Москва, Россия

E-mail: shchepetkov@ispras.ru

Проект Linux Driver Verification [1] посвящен разработке системы статической верификации драйверов устройств LDV Tools. LDV Tools используется для проверки исходного кода драйверов ядра Linux на соответствие правилам корректности, каждое из которых определяет способы корректного использования определенной части программного интерфейса ядра. На данный момент LDV Tools содержит около 40 формализованных в виде спецификаций правил [2], которые уже помогли найти и исправить свыше 140 ошибок в драйверах ядра Linux[3].

Каждая спецификация содержит модельную реализацию соответствующей части программного интерфейса ядра Linux - набор функций, моделирующих поведение программного интерфейса и осуществляющих проверку корректности его использования. Также в спецификациях содержатся инструкции для LDV Tools по замене вызовов проверяемых функций вызовами их моделей. Программный интерфейс ядра постоянно изменяется [4], и поэтому возникает задача адаптации спецификаций правил корректности под конкретную версию ядра. Для большинства спецификаций правил эта задача решается с помощью автоматического отслеживания изменений в программном интерфейсе ядра и последующей ручной правкой спецификации [5]. Однако существуют правила с неявно заданным списком участвующих в них программных интерфейсов, зависящим от особенностей их реализации. Например, правило на запрет вызовов так называемых *might sleep* функций в контексте прерывания. Каждая функция, вызывающая *might sleep* функцию, также является *might sleep* и требует проверки корректности своих вызовов. Таким образом, изменения в реализации программного интерфейса могут сильно изменить общий список таких функций. Для подобных правил актуальной становится задача автоматизации адаптации спецификаций правил к различным версиям ядра.

Предлагаемый метод решения поставленной задачи заключается в специализации спецификации правила корректности под конкретную версию ядра на основе автоматически полученной информации

о внутренних свойствах функций программного интерфейса ядра, а также о зависимостях между ними в виде графа вызовов. Анализ внутренних свойств функций и построение графа вызовов предлагается осуществлять с помощью запросов по исходному коду инструментом C Instrumentation Framework [6], [7]. Данный метод был успешно реализован и опробован на четырех спецификациях правил корректного использования *might sleep* функций.

Литература

1. Мутилин В. С., Новиков Е. М., Петренко А. К., Хорошилов А. В. Конфигурируемая система статическои верификации модулеи ядра операционных систем. Институт системного программирования РАН, Москва. 2013.

2. Новиков Е. М. Развитие метода контрактных спецификаций для верификации модулей ядра операционной системы Linux. Диссертация на соискание ученой степени кандидата физико-математических наук, Институт системного программирования РАН, Москва. 2013.

3. Список ошибок, выявленных в модулях ядра ОС Linux с помощью конфигурируемой системы статической верификации Linux Driver Verification Toolkit: `http://linuxtesting.org/results/ldv`

4. G. Kroah-Hartman. The Linux Kernel Driver Interface. `https://www.kernel.org/doc/Documentation/stable_api_nonsense.txt`

5. Щепетков И. В. Диагностика синтаксической совместимости спецификаций правил корректности с программным интерфейсом ядра ОС Linux. Выпускная квалификационная работа на степень бакалавра. Институт системного программирования РАН, Москва. 2013

6. Новиков Е. М., Хорошилов А. В. Использование аспектно-ориентированного программирования для выполнения запросов по исходному коду программ. Труды Института системного программирования РАН, т. 23, стр. 371-386, 2012.

7. Новиков Е. М. Подход к реализации аспектно-ориентированного программирования для языка Си // Программирование. 2013. No4. С. 47-65