

## **Исследование технологий LLVM для организации управляемого исполнения кода на микроконтроллерах.**

С.А. Лисицын<sup>1,2</sup>

<sup>1</sup>Московский физико-технический институт (государственный университет)

<sup>2</sup>Лаборатория СЛОТС ФРТК МФТИ

[s.a.lisitsyn@gmail.com](mailto:s.a.lisitsyn@gmail.com)

Для многих классов современных информационных систем важными являются не только функциональная корректность, но также безопасность и расширяемость. Техническим средством, позволяющим разрешить это противоречие, являются платформы управляемого выполнения кода, такие как .NET и Java, которые производят дополнительные проверки во время выполнения и обеспечивают стабильность работы системы за счет защиты от ошибок исполняемых программ или злонамеренных действий их авторов. Такой подход стал де-факто стандартом для современных информационных систем, работающих как на высокопроизводительных серверах, так и на персональных вычислительных машинах. Отчасти этому способствовала унификация, или, по крайней мере, небольшое разнообразие вычислительных архитектур и операционных систем. К достоинствам такого подхода можно отнести: надёжность и безопасность исполнения, большая степень переносимости программ, развитые стандартные библиотеки, большое сообщество разработчиков и постоянное развитие платформ.

Такой подход, в тоже время, не лишен и недостатков - повышенных накладных расходов на функционирование самой платформы.

Попытки применить аналогичный подход для вычислительных устройств на основе микроконтроллеров предпринимались в течение последних двух десятков лет. Основными препятствиями на пути разработчиков были широкое разнообразие вычислительных архитектур и крайне ограниченные ресурсы микроконтроллеров. В то же время, существуют относительно успешные примеры платформ управляемого кода для микроконтроллеров. Самой известной является технология JavaCard. Как ясно из названия, это является переносом технологии Java в мир микроконтроллеров. К сожалению, она и подобные ей решения имеют ряд серьезных недостатков:

1. Проект носит ярко выраженный отраслевой характер, коммерческий и закрытый.
2. Ориентированность на подмножество языка Java, который плохо подходит для низкоуровневой разработки.

Видится необходимость в разработке платформы управляемого кода, которая, с одной стороны, предоставляла бы разработчику возможность использовать наиболее подходящий язык программирования для конкретной задачи, но при этом обеспечивала бы расширяемость кода и защиту для кода и данных во время выполнения.

Перспективным в этом отношении видится применение технологии LLVM, которая обеспечивает поддержку многих языков программирования и является зрелым проектом с открытым исходным кодом. Данная технология является универсальной системой анализа, трансформации и оптимизации программ и использует в качестве внутреннего представления программы LLVM IR - типизированный трёхадресный код в SSA-форме. Данное представление получается в результате обработки кода на front-end этапе компиляции, что обеспечивает поддержку сразу нескольких языков программирования.

Была исследована применимость технологий LLVM для организации управляемого исполнения кода на микроконтроллерах. По результатам работы был собран макет на ARM Cortex-M3 интерпретирующий эффективное бинарное представление (биткод) LLVM IR.

### Литература

1. Аулер Р., Лонес Б. LLVM: Инфраструктура для разработки компиляторов. 2015 – 342 с.
2. Абельсон Х., Сасман Д. Д. Структура и Интерпретация Компьютерных Программ. 2010 - 608 с.
3. Кормен Т. Алгоритмы: построение и анализ, 3-е изд. 2016 – 1328 с.

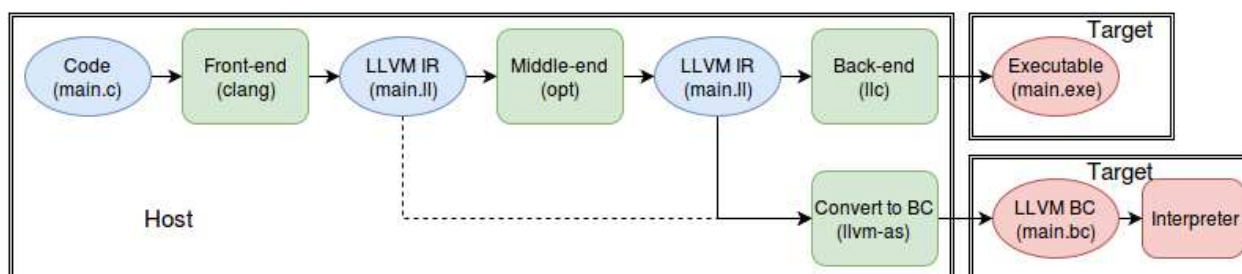


Рис. 1. Блок-схема интерпретации промежуточного представления компилятора (снизу) и компиляции (сверху).