

А. А. Пронин¹, Е. Е. Лапшева¹

¹ ФГБОУ ВО "СГУ имени Н.Г. Чернышевского", г. Саратов, Российская Федерация

ТЕСТИРОВАНИЕ ТРАНСЛЯТОРА ИЗ ЯЗЫКА ПРОГРАММИРОВАНИЯ КУМИР В ЯЗЫК ПРОГРАММИРОВАНИЯ C++

Аннотация. В данной работе приведены результаты разработки автоматизированной системы тестирования для ранее разработанного в рамках ВКР студента транслятора программного кода из языка программирования КуМир в язык C++. Автоматизация достигается путем разработки системы автоматизированного тестирования и создания набора тестов, обеспечивающих полное покрытие инструкций исходного языка программирования, а также путем создания унифицированной среды для сборки транслятора и запуска тестов.

Ключевые слова: КуМир, C++, Docker, Python, тестирование ПО, автоматизация процессов, github actions.

А.А. Pronin¹, Е.Е. Lapsheva¹

¹ Saratov State University, Saratov, Russian Federation

TESTING THE TRANSLATOR FROM THE KUMIR PROGRAMMING LANGUAGE TO THE C++ PROGRAMMING LANGUAGE

Abstract. This paper presents the results of the development of an automated testing system for a translator of program code from the KuMir programming language to the C++ language, previously developed within the framework of the student's thesis. Automation is achieved by developing an automated testing system and creating a set of tests that provide full coverage of the instructions of the source programming language, as well as by creating a unified environment for building the translator and running tests.

Keywords: KuMir, C++, Docker, Python, software testing, process automation, github actions.

Введение

В современном школьном образовании среди прочих можно выделить следующие тенденции:

1. растущий интерес к изучению робототехники;
2. глубокая интеграция языка программирования КуМир в школьную информатику и государственную итоговую аттестацию по этому предмету.

В связи с этим будет востребована система программирования роботов с использованием этого языка программирования. Ключевой составной частью системы программирования роботов является транслятор программы с языка КуМир [1] в язык C++.

Высокая доступность плат Arduino позволяет уменьшить затраты на приобретение и обслуживание комплектов для программирования роботов, облегчить обучение и освоение робототехники. Обучающийся средней школы сможет применять полученный ранее опыт программирования на языке КуМир для разработки роботов. В качестве аппаратной платформы для данной задачи выбран электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов [2] - платы Arduino, ввиду следующих причин:

- низкой стоимости устройств, периферийных модулей;
- простоты разработки аппаратных устройств на базе плат Arduino;
- высокой модульности систем;
- высокой распространенности Arduino среди робототехников.

В рамках ВКР студента был разработан транслятор из языка программирования КуМир в язык C++. Работа транслятора была поверхностно протестирована. На данный момент нельзя достоверно утверждать, что работа транслятора полностью корректна.

В статье описан процесс тестирования транслятора с целью поиска недостатков и ошибок в работе транслятора.

Существующие способы тестирования компиляторов

Создание средств и систем обеспечения надежности ПО такого класса как компиляторы, представляет собой весьма сложную задачу, особенно, если речь идет о тестировании компиляторов.

В ходе предварительного анализа для реализации системы тестирования разработанного транслятора, определили 4 основных способа тестирования компиляторов:

- наборы небольших статических тестов (regression, unit и т.п.);
- приложения-тестеры;
- coverage-тестирование;
- fuzzing-тестирование [3].

Среди готовых решений для тестирования ПО распространены следующие:

– IBM Rational Functional Tester[4] предоставляет функции автоматического тестирования для функционального, регрессионного тестирования, тестирования графических пользовательских интерфейсов и тестирования, ориентированного на данные. Данная программа не распространяется бесплатно, для использования необходимо приобрести лицензию;

– IBM Rational Test Workbench[5] также доступен по платной лицензии, с пробным периодом бесплатного использования в тридцать дней;

– Система автоматизированного тестирования TestComplete[6], предоставляющая, в том числе инструмент для регрессионного тестирования.

На этапе анализа существующего ПО для тестирования оказалось, что все рассмотренные варианты не смогут быть применены ввиду чрезмерно объемного интерфейса, а также отсутствия бесплатной подписки.

Транслятор из языка программирования КуМир в язык программирования C++ - это разработка с открытыми исходными кодами. На данный момент язык программирования КуМир поддерживает функциональный стиль программирования без создания пользовательских структур или классов.

Ввиду наличия исходных кодов транслятора и ограниченности набора инструкций языка КуМир решено разработать автоматизированную тестовую среду с набором статических тестов.

Процесс тестирования

Под тестированием компилятора понимается сборка транслятора из исходных кодов, трансляция тестовой программы и сравнение результатов работы транслятора с ожидаемым результатом. Решено автоматизировать процесс тестирования для повышения качества и снижения сложности процесса.

Поскольку язык программирования и среда КуМир - кроссплатформенные проекты, в начале процесса автоматизации стоило решить вопрос с внешними условиями - операционной системой, набором используемых пакетов и библиотек. Первым шагом к автоматизации тестирования явилось создание централизованной среды для сборки компилятора, а именно разработка dockerfile файла, содержащего набор инструкций для сборки транслятора из исходных кодов. В качестве операционной системы выбрана Ubuntu ввиду широкой распространенности, большого сообщества разработчиков и активной поддержки пакетов с данной системой в репозитории docker.

Далее встал вопрос автоматизации запуска транслятора. Для решения этой проблемы разработан скрипт на языке Python 3.

Для автоматизации запуска тестовой системы в репозитории разрабатываемого транслятора была настроена служба автоматического запуска, а именно Github Actions при отправке изменений и создании запроса на внесение изменений в целевую ветвь.

Тесты

Для тестирования разработан набор тестов, покрывающих основные инструкции языка программирования КуМир, а именно: работа с переменными (инициализация, присваивание), арифметические и логические выражения, ветвления, циклы, а также функции и процедуры. Тестирование построено на принципе сравнения ожидаемого результата с результатом работы транслятора. Тесты сгруппированы по подмножествам инструкций исходного языка. Группы тестов приведены на рис 1.

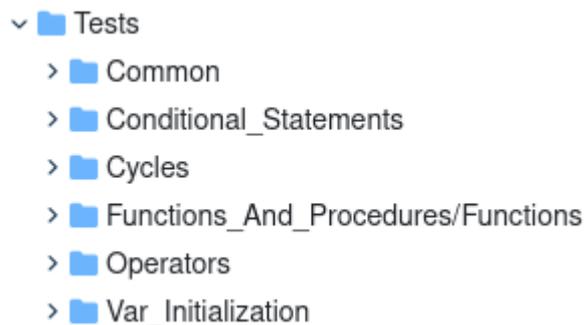


Рис. 1. Группы тестов

Группа состоит из секций, представляющих отдельные инструкции. Для создания тестовой секции, например, для тестирования трансляции функций, необходимо 2 файла - файл с расширением .kum на исходном языке программирования, а также файл с расширением .exp - файл с предполагаемым результатом работы транслятора, содержащий программу на языке C++. Пример секции тестов приведен на рис 2.

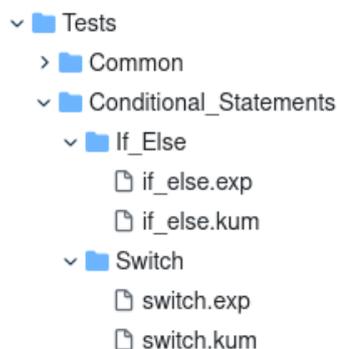


Рис. 2. Пример секции тестов

Под тестом транслятора понимается два файла, содержащих несколько инструкций на исходном и выходном языках программирования. На данный момент разработано более 200 тестов.

Программа-тестер

Для автоматизации тестирования разработано консольное приложение-тестер транслятора. В качестве языка программирования выбран Python 3. Для работы с программой разработан ряд аргументов командной строки, а также справка. Список аргументов командной строки приведен ниже:

– [-h] [--help] - show help.

- [-tr] [--translator] ["the path to pre-built kumir2 to arduino translator instance"] - show app what translator instance to use.
- [-t] [--path-to-tests] ["the path to tests folder"] = show app the folder with test files.
- [-o] [--output] ["the path to log file"] - show app where to store test logs.
- [-d] [--duplicate] - duplicate output to console.
- [-ss] [--skip-successfull] - skip open log info about successfully completed tests.
- [-sf] [--skip-failed] - skip open log info about failed tests.
- [-swe] [--skip-without-expectation] - skip open log info about tests for which the file with expectations was not found.
- [-sce] [--skip-with-compiler-error] - skip open log info about tests ended with compiler error.
- [-b] [--brief] - skip open log info about all tests.

Решено разработать программу гибкой и информативной. Для работы необходимы два аргумента – translator (путь к транслятору), а также – path-to-tests (путь к папке с тестами).

По умолчанию результат тестирования записывается в файл, так же есть возможность продублировать вывод в консоль. Существует четыре состояния, отражающие результат работы компилятора:

1. успех;
2. провал;
3. ошибка компиляции;
4. не хватает файла с ожиданиями.

Состояния “ошибка компиляции” и “не хватает файла с ожиданиями” могут быть получены по причине некорректного содержания тестовых файлов - либо написанная на языке программирования КуМир программа имеет ошибки и не может быть скомпилирована, либо не найден файл с расширением .exp, содержащий ожидания работы компилятора. В других случаях текст сообщения с результатами тестирования содержит предложение вызвать команду vimdiff с путями к файлам и прореферировать результат. Имеется возможность проанализировать как три файла сразу (исходный код, ожидаемый результат, результат работы компилятора), так и только ожидания и результат работы. Пример сообщения о результатах работы приведен на рис 3.

```
Test №0. Test /home/anton/Sources/kumir2/kumir_tests/Sources/2.
Test is not completed(
Sources for the test: /home/anton/Sources/kumir2/kumir_tests/Sources/2.kum
Test expectations: /home/anton/Sources/kumir2/kumir_tests/Expectations/2.c
Test results: /home/anton/Sources/kumir2/kumir_tests/Results/2.kumir.c.
To get more detail info about comparison results, print:
vimdiff /home/anton/Sources/kumir2/kumir_tests/Sources/2.kum /home/anton/Sources/kumir2/
kumir_tests/Expectations/2.c /home/anton/Sources/kumir2/kumir_tests/Results/2.kumir.c
or vimdiff /home/anton/Sources/kumir2/kumir_tests/Expectations/2.c /home/anton/Sources/kumir2/
kumir_tests/Results/2.kumir.c
```

Рис 3. Пример сообщения с возможностью подробно проанализировать результаты тестирования

В начале сообщения с результатами тестирования приведена краткая статистика - сколько тестов завершилось и с каким состоянием. Объем вывода можно фильтровать и выборочно отключать вывод подробной информации о сообщениях в состояниях 1-4, либо полностью отключить подробный вывод информации, добавив флаг -b. При дублировании вывода в консоль, в зависимости от состояния, сообщение будет выделено цветом. Пример краткого вывода по результатам тестирования приведен на рис. 4.

```

There were found: 28 tests.
Completed: 3
Failed: 23
With compiler error happend: 1
Missed expectation file: 1

```

Рис 4. Пример краткого вывода по окончании тестирования

Заключение

В ходе тестирования разработанного транслятора с языка программирования КуМир в язык программирования С++ был выявлен ряд ошибок в работе. На текущий момент корректно транслируется часть инструкций по объявлению и инициализации переменных. Трансляция условных выражений, циклов и функций содержит ошибки и недостатки, не позволяющие считать разработанный транслятор безопасным.

Автоматизация процесса тестирования позволяет быстро расширять список тестов и увеличивать процент покрытия, влияющий на надежность и безотказность работы транслятора. Создание тестовых случаев позитивно сказывается на процессе разработки, ведь тестовые сценарии являются документацией.

Автоматизация запуска программы-тестера при помощи Github Actions позволит быстрее отсеивать некачественный код, влияющий на работоспособность транслятора.

В дальнейшем разработанная система тестов может быть доработана при помощи методов Fuzzing-тестирования и автоматизированной генерации тестовых файлов.

Исходный код разработанной программы тестера и транслятора можно найти в открытом репозитории [7].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. <https://foxford.ru/wiki/informatika/sredaprogrammirovaniya-kumir>.
2. <http://arduino.ru/>.
3. Максименков Д.А., Рогов Р.Ю. Применение метода инструментирования тестовых программ при отладке оптимизирующих компиляторов // Вопросы радиоэлектроники. – Москва, 2010, т. 3, №3, стр. 50-61.
4. <https://www.ibm.com/products/rational-functional-tester> IBM.
5. <https://www.ibm.com/products/rational-test-workbench>.
6. <https://marketplace.appdirect.com/en-US/apps/59147/testcomplete-platform-by-smartbear/features>.
7. https://github.com/CaMoCBaJL/kumir2/tree/translator_tests.

REFERENCES

1. <https://foxford.ru/wiki/informatika/sredaprogrammirovaniya-kumir>.
2. <http://arduino.ru/>.
3. Maksimenkov D.A., Rogov R.Yu. Application of the test instrumentation method programs when debugging optimizing compilers. Questions radioelectronics, 2010, t.3, no. 3, pp. 50-61.
4. <https://www.ibm.com/products/rational-functional-tester> IBM.
5. <https://www.ibm.com/products/rational-test-workbench>.
6. <https://marketplace.appdirect.com/en-US/apps/59147/testcomplete-platform-by-smartbear/features>.
7. https://github.com/CaMoCBaJL/kumir2/tree/translator_tests

Информация об авторах

Антон Алексеевич Пронин – студент магистратуры кафедры «Информатика и программирование», ФГБОУ ВО "СГУ имени Н.Г. Чернышевского", г. Саратов, e-mail: gorka19800@gmail.com

Елена Евгеньевна Лапшева – старший преподаватель кафедры «Информатика и программирование», ФГБОУ ВО "СГУ имени Н.Г. Чернышевского", г. Саратов, e-mail: lapsheva@gmail.com

Authors

Anton Alexeevich Pronin – 1 st year Master's degree student, Faculty of Computer Science & IT, Saratov State University, Saratov, e-mail: gorka19800@gmail.com

Elena Evgenievna Lapsheva, Senior Lecture, Department of Programming and informatics, Saratov State University, Saratov, e-mail: lapsheva@gmail.com

Для цитирования

Пронин А.А., Лапшева Е.Е. Тестирование транслятора из языка программирования кумир в язык программирования C++ // «Информационные технологии и математическое моделирование в управлении сложными системами»: электрон. науч. журн. – 2023. – №2(18). – С.1-6 – DOI: 10.26731/2658-3704.2023.2(18).1-6 – Режим доступа: <http://ismm-irgups.ru/toma/218-2023>, свободный. – Загл. с экрана. – Яз. рус., англ. (дата обращения: 17.06.2023)

For citations

Pronin A.A., Lapsheva E.E. Testing the translator from the kumir programming language to the C++ programming language // *Informacionnyye tehnologii i matematicheskoe modelirovanie v upravlenii slozhnymi sistemami: elektronnyj nauchnyj zhurnal* [Information technology and mathematical modeling in the management of complex systems: electronic scientific journal], 2023. No. 2(18). P. 1-6. DOI: 10.26731/2658-3704.2023.2(18).1-6 [Accessed 17/06/23]