**Осипенко Данил Владимирович**, студент кафедры вычислительной техники и электроники Алтайского государственного университета

Научный руководитель – **Уланов Петр Николаевич**, старший преподаватель кафедры вычислительной техники и электроники Алтайского государственного университета

ЯЗЫК ПРОГРАММИРОВАНИЯ

В рамках научно-исследовательской работы разработан простой транслятор для самостоятельно спроектированного языка программирования высокого уровня в байт-код собственной виртуальной машины. Формат байт-кода также спроектирован самостоятельно. Транслятор состоит из лексического, синтаксического анализаторов и кодогенератора. Байт-кодовая виртуальная машина является стековой и поддерживает базовые типы данных (целочисленные, с плавающей точкой, строки, булевы значения).

**Ключевые слова:** Go, интерпретатор, транслятор, кодогенератор, виртуальная машина, программное обеспечение, язык программирования, лексический анализ, синтаксический анализ.

На сегодняшний день любая программа написана с помощью какого-либо языка программирования (далее ЯП), от самого низкоуровневого, например ассемблер, до высокоуровневых представителей, таких как: Python, Ruby, Go, Rust, C/C++ и т.д.. Сам по себе ЯП представляет собой: спецификацию; Программу компилятор или интерпретатор.

Компилятор (или интерпретатор) является главным элементом любого ЯП, определяющим основные показатели языка, такие как быстродействие, поддерживаемые платформы и функциональные возможности. Вследствие этого направление можно назвать одним из основополагающих и до сих пор актуальных [1, с. 1].



Рис. 1 Поверхностная структура типичного компилятора.

Язык, для которого был разработан компилятор, содержит в себе всего 2 примитивных типа данных (целочисленные и числа с плавающей точкой), возможность создавать функции, переменные, константы. Структура созданного компилятора включает в себя:

* Лексический анализ (Lexer) – разбор входного текста программы на токены.
* Синтаксический анализ (Parser) – группировка поступающих на вход токенов.
* Генератор – генерация байт кода для для стековой виртуальной машины.

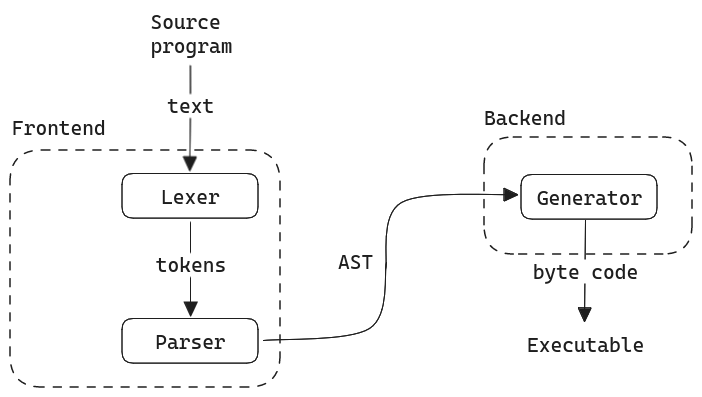


Рис. 2 Структура созданного компилятора.

Разработка компилятора проведена на языке Go (Golang). Лексический и синтаксический анализаторы созданы с помощью сторонней библиотеки participle [2]. Генератор реализован с помощью алгоритма прохода дерева (AST).

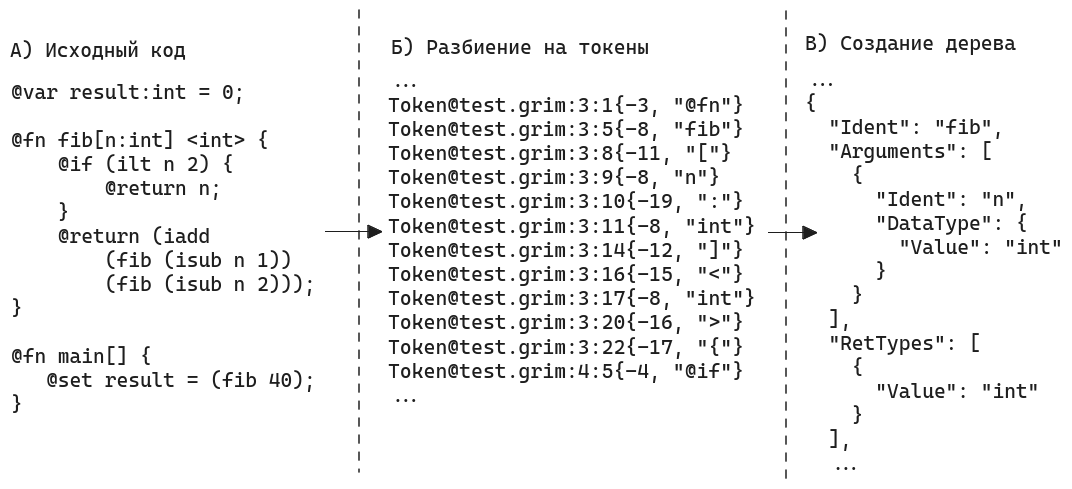


Рис. 3 Результат работы лексера (Б) и парсера (В).

Обычно для реализации виртуальных машин принято использовать низкоуровневые языки для системного программирования, например С/С++[3]. Но для простоты работы и возможности осуществления более высокой интеграции с ранее написанным компилятором, ВМ была также разработана на языке Go.

Для реализации выбрана стековая ВМ, которая выполнена в виде байт кодового интерпретатора (инструкции и их аргументы закодированы в списки байтов) [3; 4].

Она поддерживает 4 базовых типа данных: целые числа, числа с плавающей точкой, строки и булевы значения. Имеет два стека: для хранения и передачи значений; для хранения фрейма вызова функции. Фрейм, в свою очередь, имеет список локальных регистров.

По итогу разработаны компилятор и байт кодовая виртуальная машина для простого языка программирования. Получены навыки разработки и проектирования комплексных и многомодульных систем, использования систем контроля версий, знания об устройстве и функционировании языков программирования.

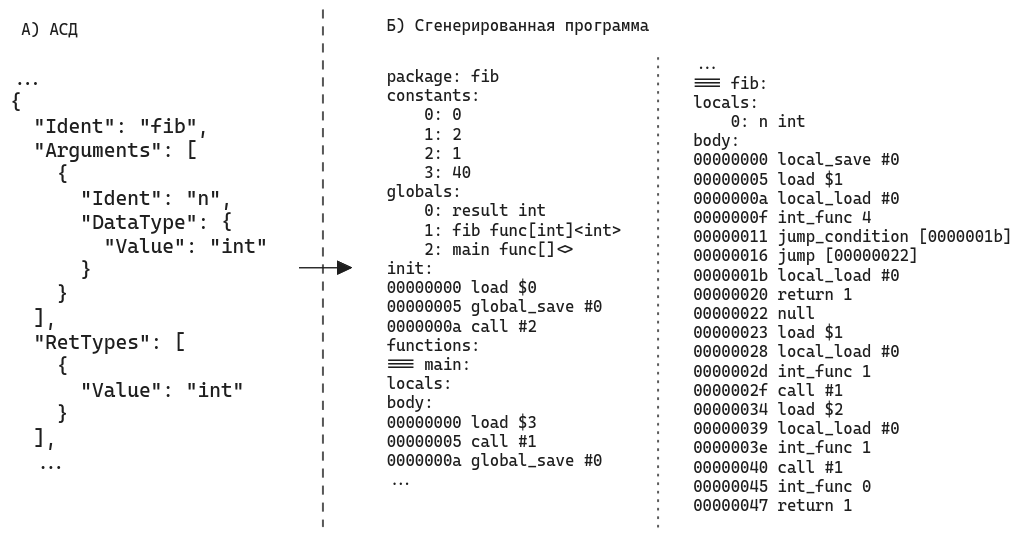


Рис. 4 Пример генерации из Абстрактного Синтаксического Дерева (А) в байт код (Б).



Рис. 5 Пример выполнения программы.

**Библиографический список**

1. Engineering of compiler: 2nd ed. /Keith D. Cooper, Linda Torczon. – Burlington: Publisher Morgan Kaufmann, 2011.-824 p.
2. A parser library for Go - GitHub [Электронный ресурс]. – Режим доступа: https://github.com/ alecthomas/participle, свободный. – Загл. с экрана. - Яз. англ.
3. A Bytecode Virtual Machine – Crafting Interpreters [Электронный ресурс] . – Режим доступа: https://craftinginterpreters.com/a-bytecode-virtual-machine.html, свободный. – Загл. с экрана. – Яз. англ.
4. Два мира виртуальных машин [Электронный ресурс]. – Режим доступа: https://habr.com/en/companies/intel/articles/254793/, свободный. – Загл. с экрана. - Яз. рус.

Сведения об авторах:

**Осипенко Данил Владимирович**, студент кафедры вычислительной техники и электроники Алтайского государственного университета, [unkn.spectrum@gmail.com](mailto:unkn.spectrum@gmail.com), +7(983)104-36-55;

**Уланов Петр Николаевич**, старший преподаватель кафедры вычислительной техники и электроники Алтайского государственного университета, ulanovpn@gmail.com, +7(905)083-73-75.