# РАЗРАБОТКА КОМПЛЕКСА ДЛЯ ПРОГРАММИРОВАНИЯ РАДИОЭЛЕКТРОННЫХ СРЕДСТВ

*Ляховский А.А., студент гр.940401*

*Белорусский государственный университет информатики и радиоэлектроники*
*г. Минск, Республика Беларусь*

*Листопад Н.И. – доктор техн. наук, профессор*

**Аннотация.** В статье описан процесс и выводы в ходе разработки комплекса для программирования радиоэлектронных средств. Сравнивается производительность результата работы комплекса с аналогичными разработками.

**Ключевые слова.** Компилятор, интерпретатор, архитектура процессоров, оптимизация, язык программирования, формальные грамматики.

Компиляторы играют важную роль в разработке и программировании различных радиоэлектронных средств. Это программное обеспечение позволяет разработчикам создавать эффективный и оптимизированный код для микроконтроллеров, что, в свою очередь, улучшает производительность устройств и уменьшает затраты на их разработку. Одним из главных свойств современных компиляторов является их способность оптимизировать генерируемый исполняемый код. Это особенно важно для микроконтроллеров, которые имеют ограниченные ресурсы, такие как память и процессорная мощность.

Цели работы:

- Разработать масштабируемый код с целью интеграции других архитектур;
- Разработать удобочитаемый синтаксис для языка программирования;
- Разработка алгоритмов оптимизаций для генерации оптимальных инструкций при генерации кода с целью улучшения производительности и уменьшении веса исполняемых файлов.

Задачей исследования является разработка программного комплекса, состоящего из компилятора некоторого собственного языка программирования и собственной среды разработки.

Компилятор в свою очередь состоит из отдельных слабосвязанных модулей, а именно: лексический анализатор, препроцессор, синтаксический анализатор и генератор машинных инструкций.

В процессе работы были использованы следующие средства разработки:

- Microsoft Visual Studio 2019;
- Язык программирования C#;
- Компилятор ассемблера FASM;
- Графическая библиотека WPF;
- Пакет для библиотеки WPF AvalonEdit;

При создании синтаксиса языка учитывалась частота использования различных служебных символов и возможность минимизации использования некоторых традиционных способов разделения блоков кода и подвыражений с целью улучшить читаемость кода для разработчика и увеличение скорости написания выражений.

Создание рабочего компилятора требует частой отладки исполняемых файлов, получаемых в результате его работы. Для удовлетворения такого требования был выбран набор инструкций генерируемых исполняемых файлов X86, являющийся идентичным для персонального компьютера, на котором велась разработка. Однако учитывая важность поддержки других наборов инструкций, для организации архитектуры исходного кода компилятора было принято решение, позволяющее легко расширить комплекс для генерации исполняемых файлов с другим набором инструкций.

Для взаимодействия с внутренними данными процесса сгенерированной программы в режиме реального времени исполнения реализован механизм взаимодействия исполняемых файлов с операционной системой через возможность статического подключения dll файлов.

Важной частью современных компиляторов являются оптимизации на этапе генерации машинных инструкций и анализа синтаксиса входного кода. К примеру, не имеет смысла генерировать инструкции в прямом соответствии с поступающим на вход выражением, оперирующим лишь константами, вместо этого можно вычислить его заранее. Таким образом появляется возможность экономить как на размере исполняемых файлов, так и повысить их производительность.

Решение задачи по реализации оптимизаций при генерации машинных инструкций заключалось в использовании оптимизирующих инструкций используемой целевой архитектуры, а также углубленный семантический анализ внутренних объектных моделей, полученных как на этапе синтаксического анализа, так и на этапе генерации кода.

Встроенная среда разработки позволяет автоматизировать работу с исходными файлами, подаваемыми на вход компилятору, а также повышает читаемость кода на разрабатываемом языке программирования. При создании среды были выбраны графические библиотеки, совместимые с языком программирования, на которым велась разработка комплекса. Пример использования показан на рисунке 1.

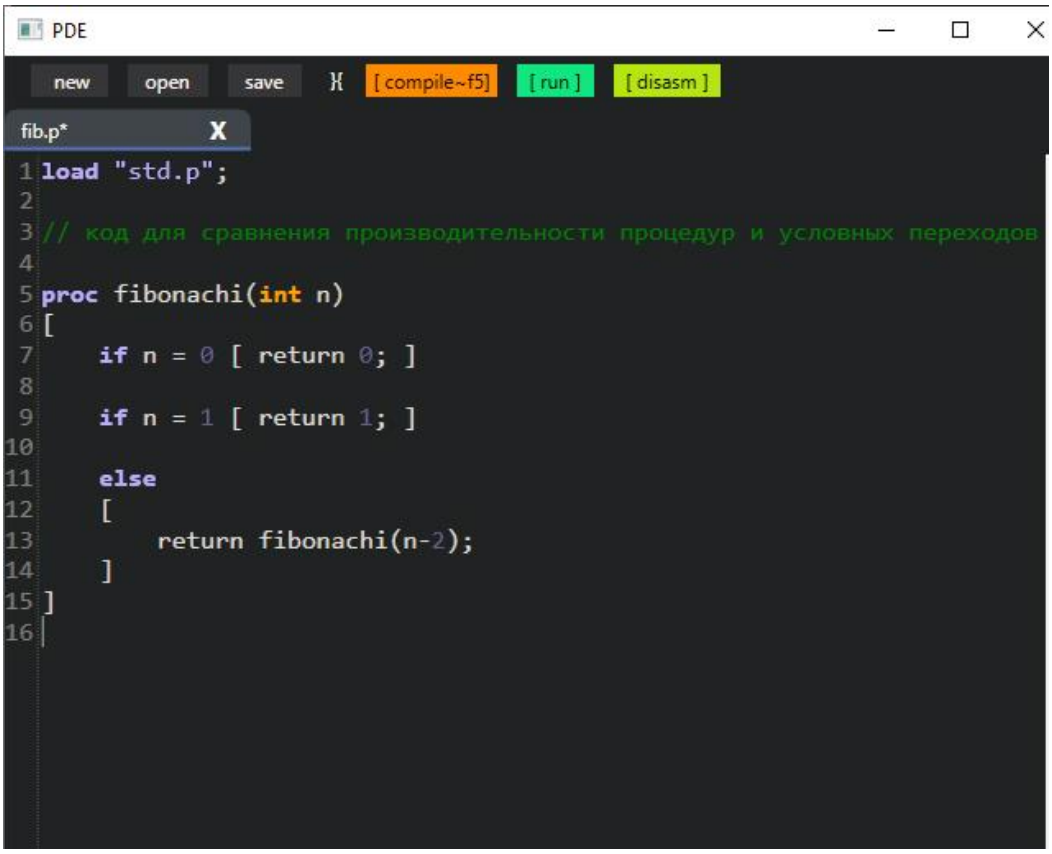


Рисунок 1 – Демонстрация среды разработки с примером исходного кода на разрабатываемом языке программирования

Для измерения производительности исполняемых файлов, полученных в результате работы компилятора с реализованными алгоритмами оптимизации, был взят алгоритм подсчёта чисел Фибоначчи, запущенный с использованием текущего программного комплекса и с использованием компилятора tcc для языка программирования Си. В обеих случаях использовалась рекурсия и условные переходы. Результаты замеров продемонстрированы в таблице 1.

Таблица 1 – Сравнение производительности компилятором tcc.

| №п | Время исполнения исполняемого файла с использованием текущего программного комплекса | Время исполнения исполняемого файла при использовании tcc |
|---|---|---|
| 1 | 973 мс | 1231 мс |
| 2 | 963 мс | 1103 мс |
| 3 | 903 мс | 993 мс |

Как результат был создан программный комплекс для программирования радиоэлектронных средств, был произведён анализ ~~замер~~ производительности сгенерированных исполняемых файлов.

***Список использованных источников:***

*1. "Язык ассемблера для процессоров Intel" Ирвин И.Н. – 211 с.*
*2. Compilers: Principles, Techniques, and Tools / Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman / Pearson Education, Inc – 396 с.*