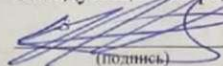


БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
Факультет информационных технологий и робототехники  
Кафедра «Программное обеспечение информационных систем и технологий»

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой

  
(подпись)

Ю.В. Полозков  
(инициалы и фамилия)

« 05 » 06 2020 г.

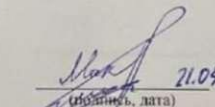
**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
ДИПЛОМНОГО ПРОЕКТА**

**«Прототип компилятора языка С»**

Специальность 1-40 01 01 «Программное обеспечение информационных технологий»

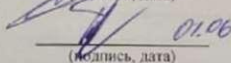
Специализация 1-40 01 01 05 «Управление качеством и тестирование программного обеспечения»

Обучающийся  
группы 10701216  
(номер)

  
(подпись, дата)

М.Г. Жибурт

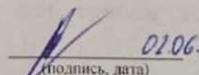
Руководитель

  
(подпись, дата)

А.А. Прихожий

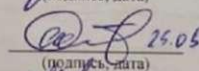
Консультанты:

по компьютерному проектированию

  
(подпись, дата)

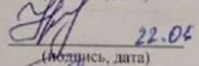
А.А. Прихожий

по разделу «Охрана труда»

  
(подпись, дата)

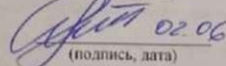
А.М. Лазаренков

по разделу «Экономика»

  
(подпись, дата)

И.В. Насонова

Ответственный за нормоконтроль

  
(подпись, дата)

Н.В. Романюк

Объем проекта:

расчетно-пояснительная записка – 70 страниц;

графическая часть – 2 листов;

магнитные (цифровые) носители – 1 единиц.

Минск 2020

## РЕФЕРАТ

### КОМПИЛЯТОР, ЯЗЫК С, ЯЗЫК АССЕМБЛЕРА, ПРОМЕЖУТОЧНОЕ ПРЕДСТАВЛЕНИЕ

Объектом разработки является «Прототип компилятора языка С».

Цель проекта заключается в создании компилятора подмножества языка С в язык Ассемблера, позволяющего демонстрировать результат работы каждой ступени компиляции.

В процессе работы выполнены следующие разработки: лексический анализатор, синтаксический анализатор, промежуточное представление, кодогенератор, логическая модель приложения, физическая модель приложения и руководство пользователя;

Результатом дипломного проектирования является разработанная программа, которая позволяет.

- Транслировать программу написанную на подмножестве языка С (нет поддержки большинства базовых типов, составных типов, указателей, массивов и макросов) в язык Ассемблер;

- Наблюдать результат работы компилятора на каждой стадии;

Студент-дипломник подтверждает, что приведенный в дипломном проекте расчетно-аналитический материал объективно отражает состояние разрабатываемого объекта, все заимствованные из литературных и других источников теоретические и методологические положения и концепции сопровождаются ссылками на их авторов.

Дипломный проект: 70 с., 55 рис., 53 источников, 2 прил.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Компилятор [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/свободный>
2. Транслятор [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/Транслятор>, свободный
3. GCC [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/GNU\\_Compiler\\_Collection](https://ru.wikipedia.org/wiki/GNU_Compiler_Collection), свободный
4. Абстрактное синтаксическое дерево [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Абстрактное\\_синтаксическое\\_дерево](https://ru.wikipedia.org/wiki/Абстрактное_синтаксическое_дерево), свободный
5. Clang [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/Clang>, свободный
6. LLVM [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/LLVM>, свободный
7. Compiler [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/Compiler>, свободный
8. Стадии компиляции картинка [Электронный ресурс] - Режим доступа: [https://upload.wikimedia.org/wikipedia/commons/thumb/c/cc/Compiler\\_design.svg/825px-Compilerdesign.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/c/cc/Compiler_design.svg/825px-Compilerdesign.svg.png), свободный
9. Лексический анализ [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Lexical\\_analysis](https://ru.wikipedia.org/wiki/Lexical_analysis), свободный
10. Парсинг [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/Parsing>, свободный
11. Очередность операторов в языке C [Электронный ресурс] - Режим доступа: [https://en.cppreference.com/w/c/language/operator\\_precedence](https://en.cppreference.com/w/c/language/operator_precedence), свободный
12. Контекстно-свободная грамматика [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Контекстно-свободная\\_грамматика](https://ru.wikipedia.org/wiki/Контекстно-свободная_грамматика), свободный
13. Форма Бэкуса-Наура [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Backus%E2%80%93Naur\\_form](https://en.wikipedia.org/wiki/Backus%E2%80%93Naur_form), свободный
14. Gnu Bison [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/GNU\\_Bison](https://en.wikipedia.org/wiki/GNU_Bison), свободный
15. Yacc [Электронный ресурс] - Режим доступа: <https://en.wikipedia.org/wiki/yacc>, свободный
16. ANTLR [Электронный ресурс] - Режим доступа: <https://en.wikipedia.org/wiki/ANTLR>, свободный
17. Расширенная форма Бэкуса-Наура [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Расширенная\\_форма\\_Бэкуса\\_—\\_Наура](https://ru.wikipedia.org/wiki/Расширенная_форма_Бэкуса_—_Наура), свободный

18. Промежуточное представление [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Intermediate\\_representation](https://en.wikipedia.org/wiki/Intermediate_representation), свободный
19. Rust-analyzer parsing [Электронный ресурс] - Режим доступа: <https://dev.to/bnjji/what-i-learned-contributing-to-rust-analyzer-4c7e>, свободный
20. Lexer [Электронный ресурс] - Режим доступа: <https://jack-vanlightly.com/blog/2016/2/3/creating-a-simple-tokenizer-lexer-in-c>, свободный
21. Сравнение оптимизации на IR и AST [Электронный ресурс] - Режим доступа: <https://stackoverflow.com/questions/22894520/compiler-code-optimization-ast-vs-ir>, свободный
22. Описание парсинга сверху вниз [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Top-down\\_parsing](https://en.wikipedia.org/wiki/Top-down_parsing), свободный
23. Описание парсинга снизу вверх [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Bottom-up\\_parsing](https://en.wikipedia.org/wiki/Bottom-up_parsing), свободный
24. LL Parser [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/LL\\_parser](https://ru.wikipedia.org/wiki/LL_parser), свободный
25. Парсинг методом рекурсивного спуска [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Recursive\\_descent\\_parser](https://ru.wikipedia.org/wiki/Recursive_descent_parser), свободный
26. C standard [Электронный ресурс] - Режим доступа: <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1548.pdf>, свободный
27. Обзор промежуточных представлений [Электронный ресурс] - Режим доступа: <https://cs.lmu.edu/~ray/notes/ir/>, свободный
28. SSA [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/SSA>, свободный
29. LLVM IR [Электронный ресурс] - Режим доступа: <https://llvm.org/docs/LangRef.html>, свободный
30. Трех-адресный код [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Three-address\\_code](https://en.wikipedia.org/wiki/Three-address_code), свободный
31. 13 презентация cs143 [Электронный ресурс] - Режим доступа: <https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/lectures/13/Slides13.pdf>, свободный
32. Кодогенерация [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/Кодогенерация>, свободный
33. Chaitins algorithm [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Chaitin%27s\\_algorithm](https://ru.wikipedia.org/wiki/Chaitin%27s_algorithm), свободный

- 34.17 презентация cs143 [Электронный ресурс] - Режим доступа: <https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/lectures/17/Slides17.pdf>, свободный
35. Оценка времени жизни переменных [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Live\\_variable\\_analysis](https://en.wikipedia.org/wiki/Live_variable_analysis), свободный
36. Сравнение промежуточного представления и ассемблера [Электронный ресурс] - Режим доступа: <https://idea.popcount.org/2013-07-24-ir-is-better-than-assembly>, свободный
37. Оптимизирующий компилятор [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Оптимизирующий\\_компилятор](https://ru.wikipedia.org/wiki/Оптимизирующий_компилятор), свободный
38. Сворачивание констант [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Constant\\_folding](https://ru.wikipedia.org/wiki/Constant_folding), свободный
39. Удаление мертвого кода [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/Dead\\_code\\_elimination](https://ru.wikipedia.org/wiki/Dead_code_elimination), свободный
40. Тестовый набор для не сложного компилятора [Электронный ресурс] - Режим доступа: [https://ru.wikipedia.org/wiki/https://github.com/nlsandler/write\\_a\\_c\\_compiler](https://ru.wikipedia.org/wiki/https://github.com/nlsandler/write_a_c_compiler), свободный
41. X64 ассемблер инструкции [Электронный ресурс] - Режим доступа: <https://www.felixcloutier.com/x86/>, свободный
42. X64 конвенции о вызовах функции [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/X86\\_calling\\_conventions](https://en.wikipedia.org/wiki/X86_calling_conventions), свободный
43. Исходный код компилятора который использовал реалокацию [Электронный ресурс] - Режим доступа: <https://github.com/msiemens/RusTiny>, свободный
44. Цикл статей как написать компилятор [Электронный ресурс] - Режим доступа: <https://norasandler.com/2017/11/29/Write-a-Compiler.html>, свободный
45. Лекции Стэнфордского университета cs143(2012) [Электронный ресурс] - Режим доступа: <https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/>, свободный
46. Логическое моделирование [Электронный ресурс] - Режим доступа: <https://www.umsl.edu/~sauterv/analysis/Fall2010Papers/varuni/>, свободный
47. Индекс ТИОБЕ [Электронный ресурс] - Режим доступа: <https://www.tiobe.com/tiobe-index/>, свободный
48. Пример логического моделирования [Электронный ресурс] - Режим доступа: <http://analyst.by/diagrams/logicheskaya-model-predmetnoy-oblasti>, свободный
49. Сколько стоит разработать компилятор [Электронный ресурс] - Режим доступа: <https://www.embecosm.com/2018/02/26/how-much-does-a-compiler-cost/>, свободный
50. Специфические процессоры для приложения [Электронный ресурс] - Режим доступа: [https://en.wikipedia.org/wiki/Application-specific\\_instruction\\_set\\_processor](https://en.wikipedia.org/wiki/Application-specific_instruction_set_processor), свободный

Карпов В.Э. Классическая теория компилятора. Учебное пособие. 2-е изд., испр. И дополн., Москва, 2018. - 92 с.