|  |  |
| --- | --- |
|  | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова» (БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-02 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Факультет |  | О |  | | Естественнонаучный |
|  |  | шифр |  | | наименование |
| Кафедра |  | О7 |  | | Информационные системы и технологии |
|  |  | шифр |  | | наименование |
| Дисциплина |  | | | Разработка трансляторов искус. языков | |

КУРСОВАЯ РАБОТА

на тему

|  |
| --- |
|  |
|  |
|  |

|  |  |
| --- | --- |
| Выполнил студент группы | О712Б |
| Борисенко Д. Д. |  |
| Фамилия И.О. |  |
| **РУКОВОДИТЕЛЬ** | |
| Устиновский Г. С. | |
| Фамилия И.О. Подпись | |
| Оценка | |
| « » | 20 г. |

**СОДЕРЖАНИЕ**

[ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 3](#_Toc151933694)

[ВВЕДЕНИЕ 4](#_Toc151933695)

[1 Постановка задачи 5](#_Toc151933696)

[2 Синтаксис и особенности языка Structured Text 6](#_Toc151933697)

[2.1 Типы данных 6](#_Toc151933698)

[2.2 Конструкции языка 6](#_Toc151933699)

[3 Описание инструментов Flex и Bison. 8](#_Toc151933700)

[4 Проектирование транслятора 9](#_Toc151933701)

[5 Реализация транслятора 11](#_Toc151933702)

[6 Демонстрация работы 13](#_Toc151933703)

[ЗАКЛЮЧЕНИЕ 15](#_Toc151933704)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc151933705)

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Structured Text (ST) — язык программирования стандарта [IEC 61131-3](https://ru.wikipedia.org/wiki/IEC61131-3). Предназначен для [программирования промышленных контроллеров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80) и операторских станций. Широко используется в [SCADA](https://ru.wikipedia.org/wiki/SCADA)/[HMI](https://ru.wikipedia.org/wiki/HMI)/[SoftLogic](https://ru.wikipedia.org/w/index.php?title=SoftLogic&action=edit&redlink=1" \o "SoftLogic (страница отсутствует))-пакетах.

International Electrotechnical Commission (IEC)  — международная некоммерческая организация по [стандартизации](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82) в области [электрических](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE), [электронных](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D0%B0) и смежных [технологий](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F). Некоторые из стандартов МЭК разрабатываются совместно с Международной организацией по стандартизации ([ISO](https://ru.wikipedia.org/wiki/ISO)).

Abstract Syntax Tree (AST) —  [конечное](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B3%D1%80%D0%B0%D1%84) помеченное ориентированное [дерево](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_(%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%BE%D0%B2)), в котором внутренние вершины сопоставлены (помечены) с [операторами](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) языка программирования, а листья — с соответствующими [операндами](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D0%BD%D0%B4). Таким образом, листья являются пустыми операторами и представляют только переменные и константы.

Flex – генератор лексического анализатора.

Bison - генератор синтаксических анализаторов.

ВВЕДЕНИЕ

В современном мире существует множество искусственных языков программирования, каждый из которых обладает своим собственным синтаксисом и структурой. В данной работе рассматривается процесс разработки транслятора для перевода исходного кода с языка ST на язык программирования C с использованием инструментов Flex и Bison [1].

Целью данной работы является создание и анализ транслятора, способного преобразовывать исходный код на языке ST [2] в эквивалентный код на языке программирования С.

1 Постановка задачи

Для разработки транслятора из языка ST в язык программирования Си были выбраны инструменты Flex [3] и Bison. Flex используется для создания лексического анализатора, который разбирает входной текст на лексемы, тогда как Bison используется для создания синтаксического анализатора, который определяет структуру предложений в языке и строит синтаксическое дерево. Обзор данных инструментов и их возможностей представлен в данной работе с целью эффективного использования их функционала при разработке транслятора. Для достижения цели разработки ставятся следующие задачи:

1. Изучение синтаксиса и особенностей языка ST.
2. Проектирование и разработка лексического и синтаксического анализаторов с использованием Flex и Bison соответственно.
3. Реализация транслятора и проверка его работоспособности на различных примерах исходного кода.

Отчет о проделанной работе должен быть оформлен в соответствии с ГОСТ 7.32-2017 [4].

2 Синтаксис и особенности языка Structured Text

**ST** – это текстовый язык высокого уровня общего назначения, по синтаксису схожий с языком Pascal. Удобен для программ, включающих числовой анализ или сложные алгоритмы. Может использоваться в программах, в теле функции или функционального блока, а также для описания действия и перехода внутри элементов SFC. Согласно IEC 61131-3 ключевые слова должны быть введены в символах верхнего регистра. Пробелы и метки табуляции не влияют на синтаксис, они могут использоваться везде.

### **2.1 Типы данных**

Согласно стандарту IEC 61131-3, язык ST поддерживает весь необходимый набор типов, аналогичный классическим языкам программирования. Целочисленные типы: SINT (char), USINT (unsigned char), INT (short int), UINT (unsigned int), DINT (long), UDINT (unsigned long), LINT (64 бит целое), ULINT (64 бит целое без знака). Действительные типы: REAL (float), LREAL (double). Специальные типы BYTE, WORD, DWORD, LWORD представляют собой битовые строки длиной 8, 16, 32 и 64 бит соответственно. Битовых полей в ST нет. К битовым строкам можно непосредственно обращаться побитно. Логический тип BOOL может иметь значение TRUE или FALSE. Физически переменная типа BOOL может соответствовать одному биту. Строка STRING является именно строкой, а не массивом. Есть возможность сравнивать и копировать строки стандартными операторами.

### **2.2 Конструкции языка**

К конструкциям языка ST относятся:

1. арифметические операции;
2. логические (побитовые) операции;
3. операции сравнения;
4. операция присвоения;
5. конструкция IF – ELSEIF – ELSE;
6. цикл FOR;
7. цикл WHILE;
8. цикл REPAET UNTIL;
9. конструкция CASE.

При записи арифметических выражений допустимо использование скобок для указания порядка вычислений. При записи выражений допустимо использовать переменные (локальные и глобальные) и константы.

3 Описание инструментов Flex и Bison.

Flex является генератором лексических анализаторов, который используется для разбора входного текста или последовательности символов на лексемы или токены. Он основан на работе с регулярными выражениями, где задаются шаблоны для распознавания лексем. Flex позволяет создавать правила для обнаружения определенных последовательностей символов в тексте программы и определять соответствующие действия для этих правил. Он часто используется в паре с Bison для построения компиляторов и трансляторов.

Bison представляет собой генератор синтаксических анализаторов (парсеров), который работает на основе контекстно-свободной грамматики. Он используется для создания синтаксического анализатора, который проверяет последовательность лексем (или токенов), определенных лексическим анализатором, и строит синтаксическое дерево в соответствии с заданной грамматикой языка. Bison позволяет описывать правила грамматики для языка программирования и определять действия, выполняемые при распознавании этих правил.

Использование Flex и Bison вместе позволяет разработчикам создавать комплексные трансляторы [5] и компиляторы для различных языков программирования. Flex выполняет лексический анализ, разбивая текст на лексемы, а Bison осуществляет синтаксический анализ, устанавливая правильную последовательность лексем и формируя структуру согласно грамматике языка. Это сочетание инструментов обеспечивает возможность создания эффективных и мощных инструментов для анализа и обработки исходного кода на различных языках программирования.

4 Проектирование транслятора

Разработка транслятора включает ряд этапов, начиная с создания лексического анализатора для выделения лексем в коде ST до создания синтаксического анализатора для построения структуры программы и генерации кода на языке Си.

Лексический анализатор, реализованный с использованием инструмента Flex, определяет правила для распознавания ключевых слов, идентификаторов, операторов, чисел и строк в исходном коде. Это обеспечивает разбиение текста на лексемы, которые затем используются синтаксическим анализатором.

Синтаксический анализатор, созданный с помощью Bison, определяет грамматику языка ST и строит абстрактное синтаксическое дерево (AST). На этапе проектирования представлено примерное синтаксическое дерево, демонстрирующее структуру программы. Дерево представлено на рисунке 4.1.

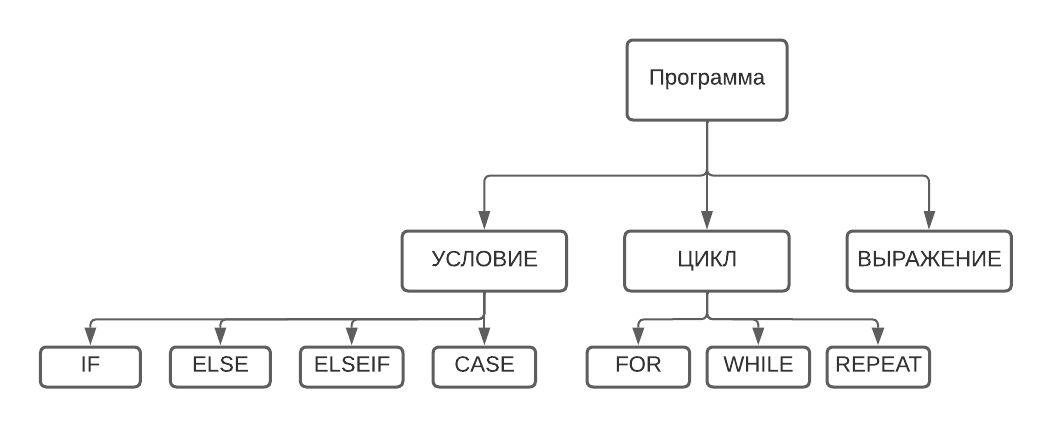


Рисунок 4.1 – Синтаксическое дерево на этапе проектирования

Тестирование транслятора представляет собой третий этап разработки. Оно включает в себя выполнение различных тестовых сценариев для проверки работоспособности и корректности работы программы на различных фрагментах кода.

Этапы разработки транслятора представляют собой комплексный процесс, включающий в себя анализ, проектирование и тестирование, направленный на создание надежного и эффективного инструмента для перевода кода из ST в язык программирования C.

5 Реализация транслятора

**Реализация транслятора начинается с установки лексем в L.l файле. В данном файле необходимо выписать все лексемы, но, чтобы лишний раз не переписывать обозначения, можно сразу вынести в шаблон эквивалентное обозначение для языка С. К примеру, чтобы не переопределять в Bison файле знак «:=», можно сразу обозначить его как «=». Во Flex файле это будет выглядеть так:**

":="        {strcpy(yylval.data, "=");  return ASSIGN;}

Так обозначаются почти все остальные лексемы, но есть исключения. Например, нетрудно заметить, что в языке программирования С отсутствует отдельное определение переменных в блоках VAR и END\_VAR, отчего эти две лексемы было решено просто проигнорировать.

Также для удобства читки лексем были проигнорированы табуляции, пробелы и переносы строк. Стоит упомянуть об использовании регулярных выражений для лексем, обозначающих знаки, числа, слова, так как намного удобнее работать с такими лексемами.

Далее реализуется синтаксический файл.

Для наглядности примерная реализация синтаксического дерева в программе изображена на рисунке 5.1.

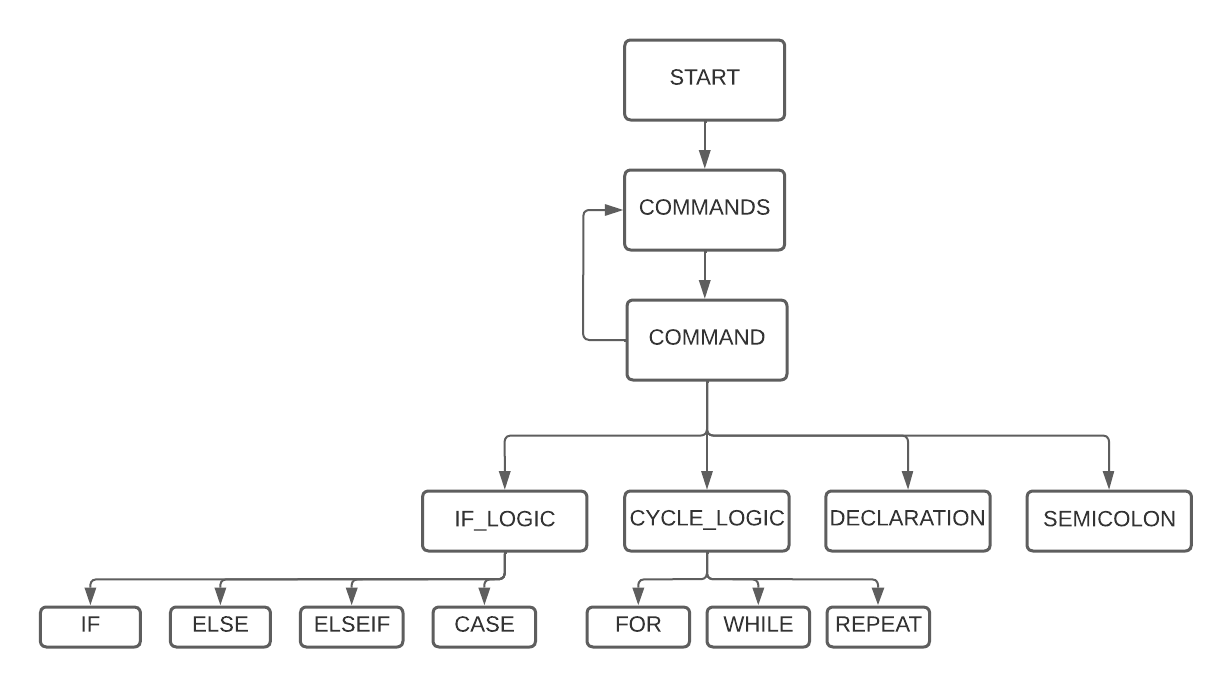


Рисунок 5.1 – Реализация синтаксического дерева в программе

Начало реализации начинается с определения START, которое и будет основным записывающим устройством в файл. С него уже идёт связь в COMMANDS, которое отвечает за запись всех команд, далее COMMAND (по факту одна команда). Несложно увидеть, что есть рекурсивная связь между COMMANDS и COMMAND, что даёт в программе возможность записи множества команд.

Далее уже идёт реализация, как и была на момент проектирования, но с учётом того, что в отдельную связь была взята точка с запятой, так как она встречается в идентичных местах, как и в языке С.

6 Демонстрация работы

Для демонстрации работы программы был сделан специальный файл ST.txt, который изображён на рисунке 6.1.

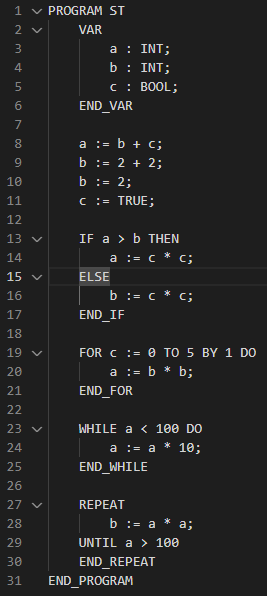


Рисунок 6.1 – Файл ST.txt

После выполнения программы выводится вся информация в файл output.txt, который изображён на рисунке 6.2.

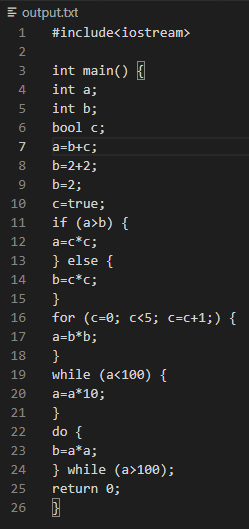


Рисунок 6.2 – Файл output.txt

По двум рисункам видно, что программа смогла транслироваться так, как это предполагалось в файле ST.txt.

ЗАКЛЮЧЕНИЕ

Разработка транслятора из языка программирования ST в язык Си представляет собой сложную и многоступенчатую задачу, которая требует глубокого понимания синтаксиса обоих языков, а также эффективного использования инструментов лексического и синтаксического анализа, таких как Flex и Bison.

В ходе данного проекта была реализована работоспособная версия транслятора, способная переводить исходный код на ST в эквивалентный код на языке программирования Си. Процесс разработки включал в себя создание лексического анализатора с помощью Flex для разбора входного текста на токены, а также синтаксического анализатора с использованием Bison для построения синтаксического дерева и генерации кода на Си.

В заключение, создание транслятора из ST в язык программирования Си представляет собой важный шаг в обеспечении возможности перехода между различными языками программирования. Реализованный транслятор является основой для дальнейших исследований и разработок в области инструментов компиляции, и может быть расширен и улучшен для более широкого применения в будущем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Bison [Электронный ресурс] URL: https://www.gnu.org/software/bison/manual/bison.html – дата обращения 25.10.2023.
2. Сведения о языке ST [Электронный ресурс] URL: <https://sm1820.github.io/beremiz/iec_guide/st_guide.html> – дата обращения 25.10.2023.
3. Документация Flex [Электронный ресурс] URL: https://westes.github.io/flex/manual/ – дата обращения 25.10.2023.
4. ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – Введ. 2018-07-01. М.: ФГУП СТАНДАРТИНФОРМ, 2018.
5. Peter Calingaert. Assemblers, compilers, and program translation (Computer software engineering series) Potomac, Maryland, USA – 276 – 303 c.

ПРИЛОЖЕНИЕ А

Исходные тексты программы располагаются в репозитории на GitHub по ссылке: https://github.com/Daniilwf/translator