**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**

**(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)**

**ОТЧЕТ**

**по курсовой работе**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Тема** | | Разработка транслятора с языка ST на язык С | | | | | | | | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
| **Обучающегося группы** | | | О712Б | | | |  | Костров Г.Ю. | | |
|  | | | группа | | | |  | Фамилия и инициалы | | |
| **Направление подготовки / специальность** | | | | 09.03.02 | | | | |  | Информационные системы и |
|  | | | | индекс | | | | |  | полное наименование направления подготовки / специальности |
| технологии | | | | | | | | | | |
| **Направленность**  **образовательной программы** | | | | | Технологии разработка информационных | | | | |
|  | | | | | профиль / специализация / магистерская программа | | | | |
| систем | | | | | | | | | | |
| **Дисциплина (модуль)** | | | | | Разработка транслятора искусственных языков | | | | | |
|  | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | | | | | | |  | | **Руководитель:** | | | |  | | | | | | | | | | | |
|  | | |  | | | | | | | |  | |  | | | | подпись | | | | | | | | | | | |
|  | | | |  |  | | | | | | |  | |  | | | |  |  | | | | | | | | |
|  | | | | |  | | | | | | |  | | ученая степень, ученое звание | | | | | | | Фамилия ИО | | | | | | | | |
|  |  |  | | | |  |  |  |  |  | |  | | **Оценка:** | |  | | | | | | | | | | | | | |
|  |  |  | | | |  |  |  |  |  | |  | | « » |  |  | | | | | | |  | 20 |  | г. |  | | |
|  | | |  | | | | | | | |  | | **Обучающийся:** | | | |  | | | | | | | | | | | |
|  | | |  | | | | | | | |  | |  | | | | подпись | | | | | | | | | | | |
|  | | | |  |  | | | | | | |  | |  | | | | | |  | |  | | | | | | | |
|  | | | | |  | | | | | | |  | |  | | | | | | | Фамилия ИО | | | | | | | | |
|  |  |  | | | |  |  |  |  |  | |  | | « » |  |  | | | | | | |  | 20 |  | г. |  | | |

**РЕФЕРАТ**

Отчет 17 стр., 4 рис., 5 источн.

FLEX, BISON, ST, C, ТРАНСЛЯТОР, Лексический Анализатор, Синтаксический Анализатор, Токен, Лексема.

В данной курсовой работе был разработан транслятор с языка программирования ST на язык программирования С. Для реализации этого транслятора были использованы синтаксический и лексический анализаторы bison и flex соответственно.

Целью данной курсовой работы является изучение процесса разработки транслятора, который выполняет перевод исходного кода с искусственного языка St на язык программирования C. Для достижения данной цели исследуется процесс работы лексического и синтаксического анализаторов flex и bison, которые помогут нам реализовать перевод исходного кода.

Объектом исследования является процесс разработки транслятора с использованием различных инструментов и технологий, их возможности и способы применения при создании переводчика искусственного языка St на язык программирования C.

Результаты данной работы могут быть полезны для разработчиков, изучающих процесс создания трансляторов и интерпретаторов, а также для студентов и исследователей, интересующихся компиляцией и языками программирования.

**СОДЕРЖАНИЕ**

[ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ 1](#_Toc155884379)

[ВВЕДЕНИЕ 1](#_Toc155884380)

[1 Постановка задачи 1](#_Toc155884381)

[2 Синтаксис и особенности языка Structured Text 1](#_Toc155884382)

[2.1 Типы данных 1](#_Toc155884383)

[2.2 Конструкции языка 1](#_Toc155884384)

[3 Описание инструментов Flex и Bison. 1](#_Toc155884385)

[4 Проектирование транслятора 1](#_Toc155884386)

[5 Реализация транслятора 1](#_Toc155884387)

[6 Демонстрация работы 1](#_Toc155884388)

[ЗАКЛЮЧЕНИЕ 1](#_Toc155884389)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 1](#_Toc155884390)

[ПРИЛОЖЕНИЕ А Текст программы 1](#_Toc155884391)

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.

Flex – генератор лексического анализатора.

Bison – генератор синтаксических анализаторов.

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.

Pascal – язык программирования.

Токен – лексема с атрибутом.

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)).

C – язык программирования высокого уровня.

ВВЕДЕНИЕ

В современном мире разработка трансляторов играет важную роль в переводе программного кода с одного языка программирования на другой. В данной курсовой работе рассматривается процесс разработки транслятора с искусственного языка ST на язык программирования С с использованием лексического и синтаксического анализаторов Flex и Bison соответственно. Лексический анализатор (Flex) используется для разбора входного текста и выделения лексем, а синтаксический анализатор (Bison) – для построения синтаксического дерева и генерации кода на языке С. В работе рассматриваются основные этапы создания транслятора, методы его реализации, а также анализируются преимущества и ограничения данного подхода. Полученный результат представляет значимый вклад в область компиляции и может послужить основой для дальнейших исследований и разработок в этом направлении.

Данная работа посвящена разработке транслятора, который может переводить исходный код с языка программирования ST на язык программирования C. Для этой цели используются инструменты Flex и Bison. Основной целью работы является создание и анализ работы транслятора, который способен преобразовывать исходный код на языке ST в эквивалентный код на языке программирования C.

1 Постановка задачи

Для разработки транслятора из языка ST в язык программирования Си были выбраны инструменты Flex [3] и Bison [1]. 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. Проектирование и разработка лексического и синтаксического анализаторов с использованием Flex и Bison соответственно.
2. 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 основным конструкциям языка ST относятся:

* арифметические операции;
* логические операции;
* операции сравнения и присваивания;
* конструкция IF – ELSEIF – ELSE;
* цикл FOR;
* цикл WHILE;

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).

5 Реализация транслятора

**Для начала работы с транслятором необходимо установить лексемы в файле l.l. В этом файле следует перечислить все лексемы, однако для удобства можно сразу указать эквивалентные обозначения на языке C, чтобы избежать повторного перечисления обозначений. Например, вместо того, чтобы заново определять лексему "INT" в файле Bison, можно сразу указать ее как "int". Структура Flex файла с такими указаниями показана на рисунке 5.1.**



Рисунок 5.1 – Пример строки в файле l.l

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.

Далее реализуется синтаксический файл.

Для наглядности примерная реализация синтаксического дерева в программе изображена на рисунке 5.2.

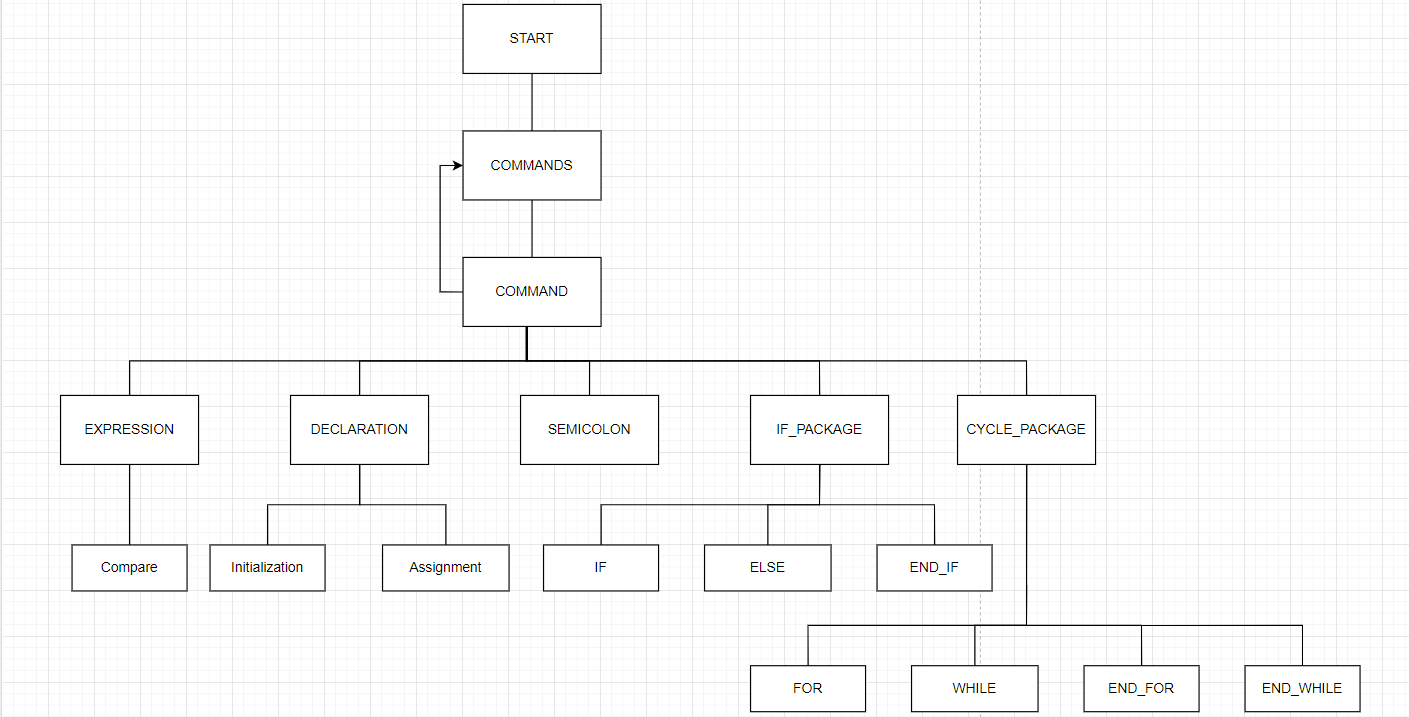


Рисунок 5.2 – Реализация синтаксического дерева в программе

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.

6 Демонстрация работы

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

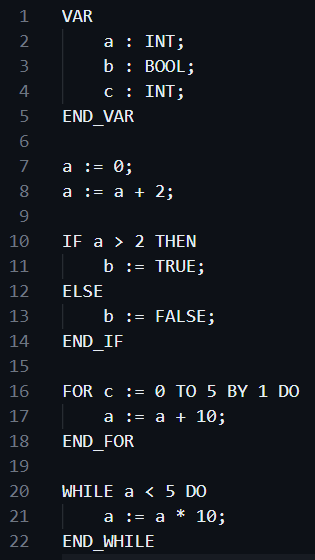


Рисунок 6.1 – Файл code.txt

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

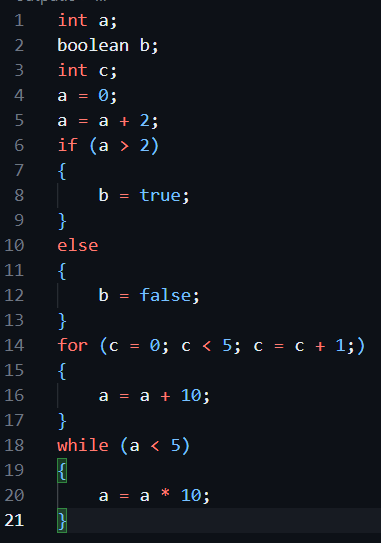


Рисунок 6.2 – Файл output.c

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 С 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 процессе данного проекта была успешно создана рабочая версия транслятора, способная переводить исходный код на языке программирования ST в эквивалентный код на языке С. Для этого был разработан лексический анализатор с использованием Flex для разбора входного текста на токены, а также синтаксический анализатор с помощью Bison для построения синтаксического дерева и генерации кода на С.

Создание транслятора из ST в язык программирования С имеет важное значение для обеспечения возможности перехода между различными языками программирования. Полученный транслятор представляет собой основу для дальнейших исследований и разработок в области компиляции, и может быть улучшен и расширен для более широкого использования в будущем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Bison. – URL: https://www.gnu.org/software/bison/manual/bison.html – дата обращения 25.10.2023.
2. Сведения о языке ST. – URL: <https://sm1820.github.io/beremiz/iec_guide/st_guide.html> – дата обращения 25.10.2023.
3. Документация Flex. – 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. Руководство по разработке на VisualStudio. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения 23.11.2023).

ПРИЛОЖЕНИЕ  
Текст программы

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