|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение  высшего образования  «Владимирский государственный университет  имени Александра Григорьевича и Николая Григорьевича Столетовых»  (ВлГУ)  разработка компилятора подмножества  процедурнОГО языка в АССЕМБЛЕР  Пояснительная записка  RU. 643.02068048.0001-01 81 01  На 11 листах   |  |  |  | | --- | --- | --- | | Руководитель |  | к.т.н. доцент кафедры ИЗИ Ю.М. Монахов | |  | | Исполнитель |  | студент гр. ИCБ-117 Романченко С.С. |   **Владимир 2020** | | | | | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  | | **АННОТАЦИЯ**  В данном документе приведена работа компилятора подмножества процедурного языка. Программа разрабатывалась на языке С++ в интегрированной среде разработки Microsoft Visual Studio 2019.  Основная функция компилятора – это проверка входного документа с цепочкой символов на соответствие входному языку и генерация выходной цепочки на языке машинных команд и ассемблере.  Основные части компилятора:  – лексический анализатор;  – синтаксический анализатор;  – оптимизатор;  – генератор кода. | | | | | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
|  |  |  |  |  | *643-02068048-00001-01* | | | | | |
|  |  |  |  |  |
| *Изм* | *Лист* | *№ докум* | *Подпись* | *Дата* |
| *Инв. № подл.* |  | *Разраб.* | |  |  |  | *«Разработка компилятора подмножества процедурного языка в ассемблер»* | *Лит* | | | *Лист* | *Листов* |
| *Проверил* | |  |  |  |  |  |  | *2* |  |
|  | |  |  |  | *ИСБ-117, Романченко С.С.* | | | | |
| *№ контр.* | |  |  |  |
| *Утв* | |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **СОДЕРЖАНИЕ**  1 РАЗРАБОТКА КОМПИЛЯТОРА 4  2 ЛЕКСИЧЕСКИЙ АНАЛИЗАТОР 5  3 СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР 7  4 ОПТИМИЗАТОР 8  5 ГЕНЕРАТОР ОБЪЕКТНОГО КОДА 9  6 ТЕСТИРОВАНИЕ 10 | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *3* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | **1 РАЗРАБОТКА КОМПИЛЯТОРА**   * 1. **Основные требования**   Разработка производилась в соответствии со следующими требованиями:  Требования к входному языку:   1. Должны присутствовать операторные скобки. 2. Должна игнорироваться индентация программы. 3. Должны поддерживаться комментарии любой длины. 4. Входная программа должна представлять собой единый модуль, но также должна быть поддержка вызова функций.   Операторы:   1. Оператор присваивания. 2. Арифметические и логические операторы. 3. Условный оператор (ЕСЛИ). 4. Операторы цикла. 5. Базовый вывод. 6. Типы (целочисленный 32 бита, с плавающей запятой 32 бита).   Требования к выходному языку:   1. В ассемблере для архитектуры х64. | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *4* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | **2 ЛЕКСИЧЕСКИЙ АНАЛИЗАТОР**  Лексический анализатор является первым этапом в разработке компилятора. Работа лексического анализатора основана на регулярных выражениях. На вход подаются строки исходной программы. В результате работы лексического анализатора строится дерево, где каждой лексеме присваивается свой уникальный ID.  На рисунке 1 представлен список лексем, которые могут встречаться в исходной программе.  Рисунок 1 – Список лексических выражений  На рисунке 2 изображено описание регулярных выражений.  На рисунке 3 показана таблица символов (в каком виде они хранятся в дереве). | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *5* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | Рисунок 2 – Описание регулярных выражений    Рисунок 3 – Таблица символов | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *6* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | **3 СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР**  На втором этапе разрабатывается синтаксический анализатор. На вход он получает строку из дерева. Парсер работает справа налево и преобразует данные в абстрактное синтаксическое дерево.  На рисунке 4 приведен код работы парсера. На рисунке 5 результат его работы.    Рисунок 4 – Парсер    Рисунок 5 – Абстрактное синтаксическое дерево | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *7* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | **4 ОПТИМИЗАТОР**  Оптимизатор удаляет лишние циклы и пустые функции. На рисунке 6 приведен пример работы оптимизатора.    Рисунок 6 – Работа оптимизатора | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *8* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | **5 ГЕНЕРАТОР ОБЪЕКТНОГО КОДА**  Генератор объектного кода делится на 3 части. Первая – это создание заголовков и выделение памяти под все переменные и функции. Вторая – инициализация переменных. Третья – создание кода посредством рекурсивного обхода абстрактного синтаксического дерева.    Рисунок 7 - Пример сгенерированного кода | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *9* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | **6 ТЕСТИРОВАНИЕ**      Рисунок 8 - Тестирование работы цикла for      Рисунок 9 - Тестирование работы цикла while | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *10* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |
|  | | Рисунок 10 - Тестирование работы вызова функций функции | | | | | | |
| *Подпись и дата* |  |
| *Инв. № дубл* |  |
| *Взаим. инв. №* |  |
| *Подпись и дата* |  |
| *Инв. № подл.* |  |
|  |  |  |  |  | *643-02068048-00001-01* | *Лист* |
|  |  |  |  |  | *11* |
| *Изм* | *Лист* | *№ докум* | *Подп.* | *Дата* |

Ссылка на листинг программы

<https://github.com/sofiaromanchenko/Compiler>