Курсовая работа

РАЗРАБОТКА КОМПИЛЯТОРА ПОДМНОЖЕСТВА ПРОЦЕДУРНОГО ЯЗЫКА

ФОРМУЛЯР

643.02068048.00001.ФО

Аннотация

Документ представляет собой формуляр на разработанный в рамках курсовой работы компилятор процедурного языка.

Документ содержит:

* общие сведения;
* основные характеристики;
* комплектность.

Содержание

[1 Общие сведения 4](#_Toc135392859)

[2 Основные характеристики 5](#_Toc135392860)

[2.1 Перечень реализуемых функций 5](#_Toc135392861)

[2.2 Описание принципов функционирования АС, регламент и режимы функционирования 5](#_Toc135392862)

[3 Комплектность 7](#_Toc135392863)

[3.1 Перечень технических и программных средств, в том числе носителей данных 7](#_Toc135392864)

[3.2 Перечень эксплуатационных документов 7](#_Toc135392865)

1. Общие сведения

Название программы: Компилятор процедурного языка в ASM код.

Основная цель компилятора процедурного языка в код для ASM состоит в автоматизации процесса преобразования исходного кода на процедурном языке в низкоуровневый код.

Программа разработана на языке Python3, используя сторонние библиотеки LLVMLite, Antlr-4.

Заказчик - ВлГУ имени Александра Григорьевича и Николая Григорьевича Столетовых, кафедра ИЗИ.

Разработчик программы – ст.гр. ИСБ-120 Сухецкий Григорий.

Документация к программе включает в себя:

* Руководство программиста, содержащее информацию о назначении, характеристиках и обращении к программе.
* Программа и методика испытаний, описывающие процесс проверки функциональности и корректности работы программы.
* Руководство пользователя, предоставляющее инструкции по установке, использованию и настройке программы.
* Формуляр, содержащий общую информацию о программе.

1. Основные характеристики
   1. Перечень реализуемых функций

Программа реализует следующие функции:

* Лексический анализ исходного кода на процедурном языке.
* Синтаксический анализ и построение абстрактного синтаксического дерева (AST).
* Генерация низкоуровневого ASM кода, соответствующего исходному коду программы.
* Оптимизация кода для повышения производительности и эффективности выполнения программы.
  1. Описание принципов функционирования АС, регламент и режимы функционирования

Программа работает на основе следующих принципов функционирования:

Лексический анализ: Исходный код программы разбирается на лексемы (токены). Каждая лексема представляет собой элемент языка, такой как идентификатор, ключевое слово, оператор или значение.

Синтаксический анализ: Лексемы объединяются в грамматические конструкции согласно заданной грамматике языка. В результате строится абстрактное синтаксическое дерево (AST), которое представляет структуру исходного кода программы.

Семантический анализ: Производится проверка семантики языка, включающая проверку типов данных, области видимости переменных, правильности использования операторов и других аспектов языка. Выявленные ошибки сообщаются программисту.

Генерация кода: На основе AST и семантического анализа генерируется низкоуровневый код на языке ассемблера, соответствующий исходному коду программы.

Оптимизация кода: Во время генерации код подвергается оптимизации для повышения его производительности и эффективности выполнения программы. В результате получается оптимизированный код, который может быть исполнен на различных платформах.

Программа функционирует следующим образом:

* Разработчик размещает исходный код программы на процедурном языке в файле test.revi в папке с программой.
* Запускается файл main.py, который последовательно вызывает методы из других файлов для компиляции исходной программы.
* В процессе компиляции, программа проверяет синтаксис, семантику и генерирует код на языке ассемблера.
* Если в результате компиляции не было ошибок, сгенерированный код сохраняется в папку с программой в файл revi.s
* Разработчик может использовать сгенерированный код для дальнейшей обработки, оптимизации или компиляции в машинный код.

1. Комплектность
   1. Перечень технических и программных средств, в том числе носителей данных

Компилятор состоит из следующих подсистем:

1. Подсистема лексического и синтаксического анализа
2. Подсистема абстрактного синтаксического дерева
3. Подсистема таблицы символов
4. Подсистема генерации кода
   1. Перечень эксплуатационных документов

Перед эксплуатацией программисту необходимо ознакомиться со всеми необходимыми инструкциями пользования программой, которые содержатся в документе «Руководство программиста».

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Лист регистрации изменений*** | | | | | | | | | |
| *Изм.* | *Номера листов (страниц)* | | | | *Всего листов (страниц) в доку-менте* | *Номер доку-мента* | *Входящий номер сопроводи-тельного документа и дата* | *Подпись* | *Дата* |
| *изменен-ных* | *заменен-ных* | *новых* | *аннулиро-ванных* |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |