**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

БАКАЛАВРСКАЯ ПРОГРАММА **09.03.01/03 Вычислительные машины, комплексы,**

**системы и сети**

Компилятор для языка программирования на основе

обратной польской записи

Техническое задание на курсовую работу

по дисциплине Технология разработки программных систем

Листов 8

Студент \_гр. ИУ6-53Б**\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**В.К. Залыгин**\_\_\_\_**

(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель курсовой работы,

(Ст. преподаватель)  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_**Б.И. Бычков**\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2024

# 1 Введение

Настоящее техническое задание распространяется на разработку компилятора для стекового языка программирования (далее — исходный язык) с синтаксисом на основе обратной польской записи (postfix language compiler) [PLC]. Решение могут использовать разработчики, создающие программы на исходном языке с целью описания алгоритмов обработки данных над целыми числами, а также компиляции кода в объектные и исполняемые файлы под машины с операционной системой семейства Linux и архитектурой процессора x86-64, генерации ассемблерных листингов.

Стековые языки программирования используются в сферах, требующих высокой производительности и минимализма. Стековая архитектура, лежащая в основе данных языков, позволяет оперировать данными, хранящимися на стеке, и последовательно выполнять операции без необходимости использовать переменные. Вследствие этого код на стековых языках часто более лаконичен и прост для анализа. Также такой подход делает языки более гибкими, позволяя создавать и комбинировать функции как объекты первого класса.

По сравнению с аналогичными компиляторами (forth, joy, cat) преимуществом данной разработки является простота исходного языка и малое количество синтаксических конструкций, что позволяет программисту быстро освоить парадигму исходного языка и получить практические навыки построения программ.

# 2 Основания для разработки

Программа PLC разрабатывается по тематике кафедры.

# 3 Назначение разработки

Назначение PLC заключается в трансляции текста на исходном языке в текст на языке ассемблера и сборки программ в объектные и исполняемые файлы. Разработчики могут использовать компилятор для составления программ обработки данных над целыми числами.

# 4 Требования к программному изделию

## 4.1 Требования к функциональным характеристикам

## 4.1.1 Выполняемые функции

* генерация ассемблерного кода в синтаксисе Intel из текстов на исходном языке;
* создание объектных файлов из текстов на исходном языке;
* создание исполняемых файлов из текстов на исходном языке.

## Исходные данные

Исходные данные должны быть представлены текстом программы на исходном языке и флагами, указывающими на тип выходного файла (ассемблерный листинг, объектный файл, исполняемый файл) и его имя.

Исходный язык должен поддерживать следующие конструкции:

- сохранение целых 32-разрядных чисел со знаком (элементов стека) в стек операндов;

- арифметические операции с целыми числами (сложение, вычитание, умножение, деление целочисленное);

- операции работы со стеком (дублирование, удаление, перемещение элементов на вершину элементов стека);

- операция создания списка команд;

- операция применения списка команд;

- ветвление;

- операции сравнения чисел (равенство, неравенство, больше, меньше);

- операция преобразования числа в булево значение;

- операции над булевыми значениями (и, или, не);

- операция присваивания имени элементу;

- операция добавления элемента по имени на стек;

- операция вывода элемента в стандартный поток вывода;

- операция ввода элемента из стандартного потока ввода;

## 4.1.3 Результаты:

* в случае успешной операции — файл с ассемблерным листингом или объектный файл, или исполняемый файл;
* в случае неуспешной операции — ошибка с описанием проблемы (ошибка в синтаксисе текста программы на исходном языке, ошибка отсутствия необходимых зависимостей компилятора).

## 4.2 Требования к надежности

4.2.1 Предусмотреть контроль синтаксической корректности текста на исходном языке.

4.2.3 Предусмотреть контроль консистентности флагов, передаваемых при вызове программы-компилятора.

## 4.3 Условия эксплуатации

Условия эксплуатации в соответствии с СанПиН 2.2.2/2.4.1340-03.

## 4.4 Требования к составу и параметрам технических средств

4.4.1 Программное обеспечение должно функционировать на IBM-совместимых персональных компьютерах.

4.4.2 Минимальная конфигурация технических средств, на которых развернут компилятор:

4.4.2.1 Архитектура процессора x86-64

4.4.2.2 Количество ядер процессора 1 Шт.

## 4.4.2.3 Объем ОЗУ 1 Гб.

4.4.3 Требования к конфигурации технических средств, на которых исполняется скомпилированная программа: процессор должен поддерживать набор команд x86-64.

## 4.5 Требования к информационной и программной совместимости

4.5.1 Программное обеспечение должно работать под управлением операционных систем семейства GNU/Linux.

4.5.2 Программное обеспечение должно иметь интерфейс командной строки.

# 5 Требования к программной документации

5.1 Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

5.2 В состав сопровождающей документации должны входить:

5.2.1 Расчетно-пояснительная записка на 25-30 листах формата А4 (без приложений 5.2.2 и 5.2.3).

5.2.2 Техническое задание (Приложение А).

5.2.3 Руководство программиста (Приложение Б).

5.3 Графическая часть должна быть включена в расчетно-пояснительную записку в качестве иллюстраций:

5.3.1 Схема структурная программного обеспечения.

5.3.2 Схемы алгоритмов.

5.3.3 Функциональная диаграмма программного обеспечения.

5.3.4 Синтаксическая диаграмма грамматики исходного языка.

5.3.5 Диаграмма пакетов программного продукта.

5.3.6 Синтаксическая диаграмма консольного интерфейса приложения в виде РБНФ.

5.3.7 Таблицы тестов.

# 6 Стадии и этапы разработки

| **Этап** | **Содержание этапа** | **Сроки и**  **объем** | **Представляемые результаты** | |
| --- | --- | --- | --- | --- |
| **Спецификации и программный продукт** | **Документы** |
|  | Выбор темы, составление задания, решение организационных вопросов | 1..2 недели  (10 %) | - | **Заполненный бланк задания на курсовую работу – вывешивается на сайт кафедры для получения утверждающей подписи заведующего кафедрой** |
|  | Анализ предметной области, разработка ТЗ. Исследование методов решения, выбор основных проектных решений | 3..4 недели | Результаты декомпозиции предметной области.  Эскизный проект: интерфейс, схемы, возможно, часть программы (выбранные готовые решения). | Фрагмент расчетно-пояснительной записки с обоснованием выбора средств и подходов к разработке |
|  | **Сдача ТЗ** | **4 неделя**  **(25 %)** | **-** | **Техническое задание – утверждается руководителем** |
|  | Проектирование и реализация основных компонентов – ядра программы | 5..7 недели | Технический проект основной части: структура программы, алгоритмы программ.  Программный продукт, реализующий основные функции (демонстрируется руководителю) | Фрагмент расчетно-пояснительной записки с обоснованием разработанных спецификаций  Тексты части программного продукта, реализующего основные функции. |
|  | **Сдача прототипа программного продукта** | **7 неделя**  **(50 %)** | **Прототип программного продукта – демонстрируется руководителю** |  |
|  | Разработка компонентов, обеспечивающих функциональную полноту | 8..10 | Рабочий проект программы.  Готовая программа | Черновик расчетно-пояснительной записки.  Тексты программного продукта. |
|  | **Сдача программного продукта** | **11 неделя**  **(75 %)** | **Готовая программа – оценивается руководителем в баллах** | **-** |
|  | Тестирование программы и подготовка документации | 12..14 | Тесты и результаты тестирования. | РПЗ и Руководство пользователя. |
| **9.** | **Оформление и сдача документации** | **14 неделя**  **(90 %)** | **–** | **Расчетно-пояснительная записка и Руководство пользователя – проверяются и подписываются руководителем** |
| 10. | Защита курсовой работы | 15..16 недели  (100%) | – | Доклад (3-5 минут). Защита курсовой работы.  Подписанная документация – вывешивается на сайт кафедры |

# 7 Порядок контроля и приемки

## 7.1 Порядок контроля

Контроль выполнения осуществляется руководителем еженедельно.

## 7.2 Порядок защиты

Защита осуществляется комиссии преподавателей кафедры.

## 7.3 Срок защиты

Срок защиты: 15-16 недели.

# 8 Примечание

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя