**Отчёт по лабораторной работе №8** по курсу «Фундаментальная информатика»

Студент группы М8О-101БВ-24 Волоснев Федор Антонович, № 4 по списку

Контакты (e-mail): feodor.a.v.03@gmail.com

Работа выполнена: «7» ноября 2024 г.

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчёт сдан « » \_\_\_\_\_\_\_\_ 2024 г., итоговая оценка \_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Изучение этапов компиляции программы, написанной на языке С

1. **Цель работы:** Составление и отладка программы из нескольких файлов на языке С, с последующим созданием искусственных ошибок на каждом из этапов компиляции программы, чтобы показать что разные ошибки срабатывают в разные моменты компиляции.
2. **Задание: Написать программу из нескольких файлов и показать успешную компиляцию. Придумать по 2 ошибки для каждого из этапов и показать, что эти ошибки происходят на нужном этапе.**
3. **Оборудование** (лабораторное):

ЭВМ\_\_\_\_\_\_\_\_\_\_, процессор \_\_\_\_\_\_\_\_\_\_\_\_\_\_, имя узла сети \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с ОП \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_ Мб, НМД \_\_\_\_\_\_\_ Мб. Терминал \_\_\_\_\_\_\_\_\_\_\_\_ адрес \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Принтер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_\_\_\_\_ с ОП\_\_\_\_\_\_, НМД \_\_\_\_\_ Мб. Монитор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Программное обеспечение (лабораторное):**

Операционная система семейства \_\_\_\_\_\_\_\_\_\_\_, наименование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_

интерпретатор команд \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы \_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства \_\_\_\_\_\_\_, наименование \_\_\_\_\_\_\_\_\_\_\_\_ версия \_ \_\_\_\_\_

интерпретатор команд \_\_\_\_ \_\_\_\_\_\_\_\_ версия \_\_\_\_.

Система программирования \_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере:

**6. Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема,диаграмма,рисунок,таблица] или формальные спецификации с пред- и постусловиями)

Программа будет состоять из трех файлов. Первый – с функцией main, основной. Второй – файл, содержащий определение некоторых математических функций (сложение, вычитание, умножение, деление). Третий – заголовочный файл, в котором эти математических функции будут объявлены.

Демонстрируем успешную компиляции программы.

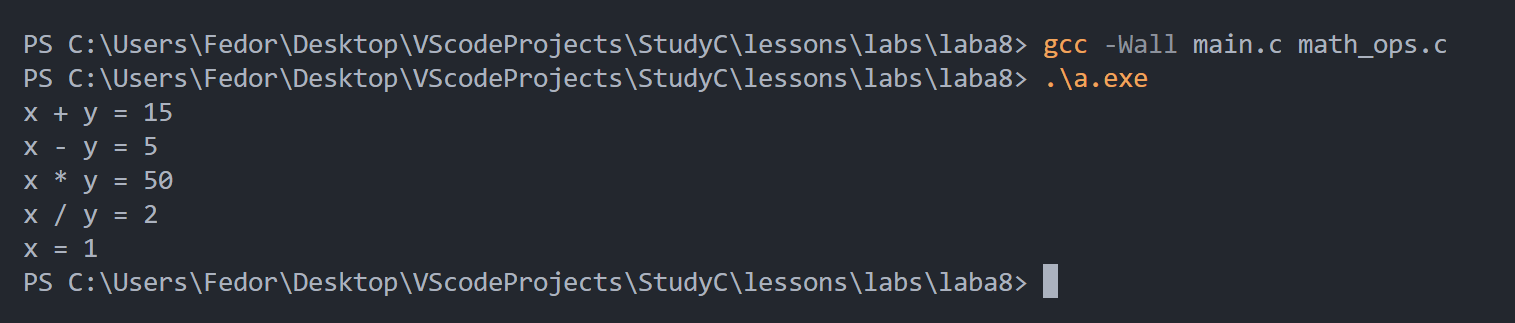
Придумаем по 2 ошибки для каждого из этапов компиляции. Будем комментировать строки, приводящие к ошибкам, чтобы:

* Указать этап компиляции, на котором происходит ошибка, созданная данной строкой.
* Для удобного применения этой ошибки нужно просто расскомментировать строку.

Чтобы доказать, что ошибка срабатывает на нужной нам этапе, будем выводить результат предыдущего этапа, где ошибок нет, и результат нашего этапе с ошибкой (если этап первый, то выводить предыдущий не нужно).

Для этого используем такие флаги при запуске gcc:

* gcc -E имя\_файла.с – вывести результат работы препроцессора и остановится.
* gcc -S имя\_файла.c – вывести результат работы после компиляции и остановится
* gcc -c имя\_файла.с – вывести результат работы после ассемблирование и остановится

1. **Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе) и тесты либо соображения по тестированию].
2. Создадим заголовочный файл math\_ops.h, где объявим функции простейших математических операций (add, subtract, multiply, divide)
3. Создадим файл math\_ops.c, где определим функции, описанные выше.
4. Создадим главный файл main.c, куда подключим math\_ops.h и будем использовать результат работы функций математических операций для вывода. x = 10, y = 5. Успешно скомпилируем.  
   
5. Напишем код, приводящий к 2-м разным ошибкам на нужном нам этапе (всего 4 этапе – 8 ошибок):
   1. Препроцессор:
      1. Циклическое подключение двух файлов. В файле main.c напишем #include math\_ops.h, чтобы подключить функции, объявленные в этом заголовочном файл  
         В math\_ops.h напишем #include main.c, тем самым зациклив подключение двуx файлов. Запустим gcc -E main.c, чтобы показать, что на этапе препроцессора возникает ошибка.  
           
         Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

         Автоматически созданное описание
      2. Напишем макрос с помощью директивы #define, указав некорректно его название. Запустим gcc -E main.c, чтобы показать, что на этапе препроцессора возникает ошибка.  
           
         Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

         Автоматически созданное описание
   2. Компиляция
      1. Добавим строчку код: printf("all clear\n" с пропущенной скобкой. Убедимся, что после команды gcc -E main.c ошибок не возникнет (этап препроцессора пройден без ошибок), однако запустим gcc -S main.c возникнет ошибка.  
           
         Изображение выглядит как текст, снимок экрана, Шрифт

         Автоматически созданное описание
      2. В файле math\_ops.h объявим глобальную переменную типа int:   
         extern int global\_variable;  
         В файле math\_ops.c опять объявим переменную global\_variable, но уже типа float. Убедимся что ошибка возникнет именно на этапе компиляции:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

* 1. Ассемблирование
     1. В файле main.c напишем ассемблерную вставку с несуществующей инструкцией. Вызовем gcc -S main.c, убедимся что ошибок нет, вызовем gcc -c main.c и увидим ошибку.  
          
        Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

        Автоматически созданное описание
     2. В файле main.c напишем ассемблерную вставку с инструкцией mov, при этом укажем несуществующий регистр %%eax. Вызовем gcc -S main.c, убедимся что ошибок нет, вызовем gcc -c main.c и увидим ошибку.  
          
        Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

        Автоматически созданное описание
  2. Линковка
     1. В файле math\_ops.h объявим глобальную функцию global\_func(), но не определим её. В файле math\_ops.c вызовем global\_func(). Убедимся в отсутствии ошибок на этапе ассемблирования, но увидим ошибку на этапе линковки двух объектных файлов.  
        Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

        Автоматически созданное описание
     2. В файле math\_ops.c определим функцию main(int a, int b). На этапе ассемблирования ошибки не произойдет, так две разные функции, но с одинаковым названием main находятся в разных файлах, однако на этапе линковки произойдет ошибка.  
        Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

        Автоматически созданное описание

*Пункты 1-7 отчёта составляются сторого до начала лабораторной работы.*

*Допущен к выполнению работы. \_\_\_\_\_\_\_\_\_\_\_* **Подпись преподавателя**

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).
2. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные события(ошибки в сценарии и программе,нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Выводы**

Изучив этапы компиляции я узнал, что можно писать ассемблерные вставки в код на СИ, а также скомпилировать файл .с в файл .asm, где можно посмотреть как выглядит мой код на ассемблере. Также если программа состоит из нескольких файлов, то можно отдельно скомпилировать каждый из них до объектного файла .o и уже вручную выполнить этап линковки. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Недочёты при выполнении задания могут быть устранены следующим образом:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_