Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 1

тема «Знакомство с языком программирования C»

по дисциплине «Языки программирования и методы трансляции»

Выполнил: студент группы ИСТ-22-1б Братчиков З.С.

Проверил: Батин С.Е.

Пермь, 2023

Содержимое

[Упражнение 1 3](#_Toc114935095)

[1.1. Постановка задачи 3](#_Toc114935096)

[1.2. Код программы 3](#_Toc114935097)

[1.3 Пример работы программы 3](#_Toc114935098)

[Упражнение 2 3](#_Toc114935099)

[2.1. Постановка задачи 3](#_Toc114935100)

[2.2. Код программы 4](#_Toc114935101)

[2.3. Примеры работы программы 4](#_Toc114935102)

[Упражнение 3 6](#_Toc114935103)

[3.1. Постановка задачи 6](#_Toc114935104)

[3.2 Код программы 6](#_Toc114935105)

[3.3. Пример работы программы 6](#_Toc114935106)

[Упражнение 4 6](#_Toc114935107)

[4.1. Постановка задачи 6](#_Toc114935108)

[4.2. Код программы 7](#_Toc114935109)

[4.3. Пример работы программы 7](#_Toc114935110)

[Упражнение 5 8](#_Toc114935111)

[5.1 Постановка задачи 8](#_Toc114935112)

[5.2 Код программы 8](#_Toc114935113)

[5.3 Пример работы программы 9](#_Toc114935114)

# Упражнение 1

## 1.1. Постановка задачи

Написать и запустить программу, выводящую строку “Hello, World!” в терминал на языке программирования C.

## 1.2. Код программы

#include <stdio.h>

void main (){

puts ("Hello, World");// Вывод строки в консоль

}

## 1.3 Пример работы программы

На рис. 1 продемонстрирован результат работы программы 1 упражнения.

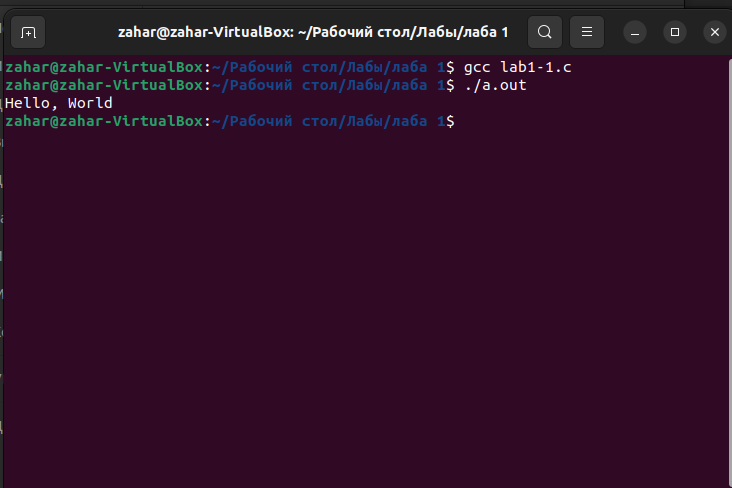


Рис.1. Результат программы задания 1

# Упражнение 2

## 2.1. Постановка задачи

Написать и получить результат работы программы, решающей линейное уравнение ax+b=c, коэффициенты a, b, с которого задаются при выполнении.

## 2.2. Код программы

#include <stdio.h>

void main(){

double a, b, c, x;

// Ввод коэффициентов a, b и c

printf(" Введите коэффициент a: ");

scanf("%lf", &a);

printf(" Введите коэффициент b: ");

scanf("%lf", &b);

printf(" Введите коэффициент c: ");

scanf("%lf", &c);

// Проверка на нулевой коэффициент a

if (a == 0) {

if (b == c) {

printf("Уравнение имеет бесконечно много решений.\n");

}else{

printf("Уравнение не имеет решений.\n");

}

} else {

// Решение уравнения

x = (c - b) / a;

printf("x = %.2f", x);

}

}

## 2.3. Примеры работы программы

На рис.2, 3, 4 показаны результаты работы программы упражнения 2.

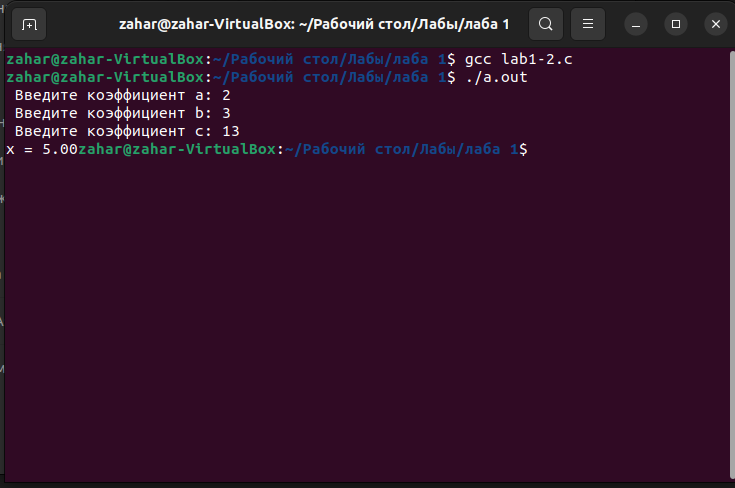
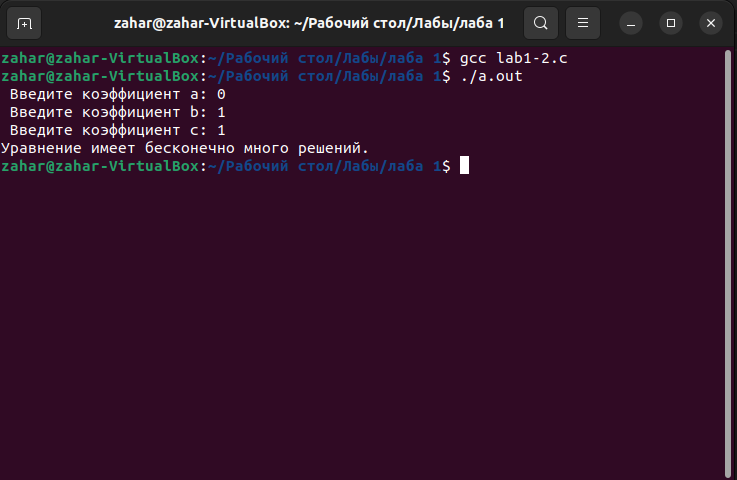
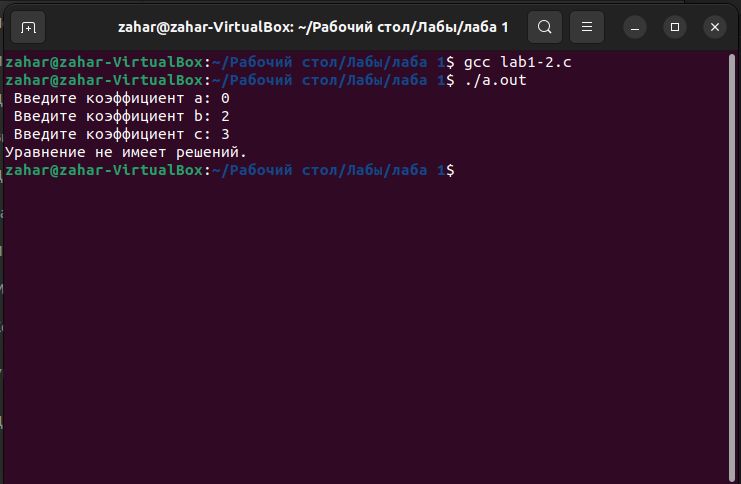


Рис.2. Пример 1 работы программы

Рис. 3. Пример 2 работы программы

Рис. 4. Пример 3 работы программы

# Упражнение 3

## 3.1. Постановка задачи

Вывести в терминал набор аргументов вызова исполняемого файла, переданных при вызове программы.

## 3.2 Код программы

#include <stdio.h>

void main(int argc, char \*argv[]) { // argc - счетчик аргументов командной строки, argv - массив указателей на строки (каждый элемент массива - аргумент командной строки).

printf("Количество аргументов: %d\n", argc);// Вывод количества аргументов, переданных командной строке

// Вывод аргументов командной строки в терминал построчно

for (int i = 0; i < argc; i++) {

printf("Аргумент %d: %s\n", i, argv[i]);

}

}

## 3.3. Пример работы программы

На рис.5 представлен результат работы программы упражнения 3.

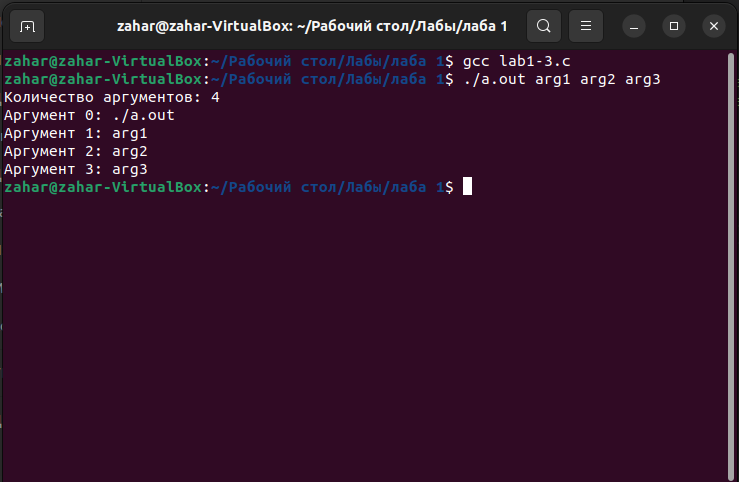


Рис. 5. Пример работы программы упражнения 3

# Упражнение 4

## 4.1. Постановка задачи

Вывести в терминал содержимое текстового файла, название которого передается при вызове исполняемого файла.

## 4.2. Код программы

#include <stdio.h>

#include <stdlib.h>// заголовочный файл который содержит в себе функции, занимающиеся выделением памяти, контролем процесса выполнения программы, преобразованием типов и другие(используется для NULL)

void main(int argc, char \*argv[]) {

// Открытие файла для чтения

FILE \*file = fopen(argv[1], "r");

if (file == NULL) {

perror("Ошибка при открытии файла");// Сообщение об ошибке при открытии файла(функция fopen возвращает указатель "NULL" при неудачной попытке открытия файла)

}

// Чтение файла и его вывод построчно

char line[1024]; // Создание массива (буффера) для хранения строки

while (fgets(line, sizeof(line), file)) {

printf("%s", line);

}

// Закрытие файла

fclose(file);

}

## 4.3. Пример работы программы

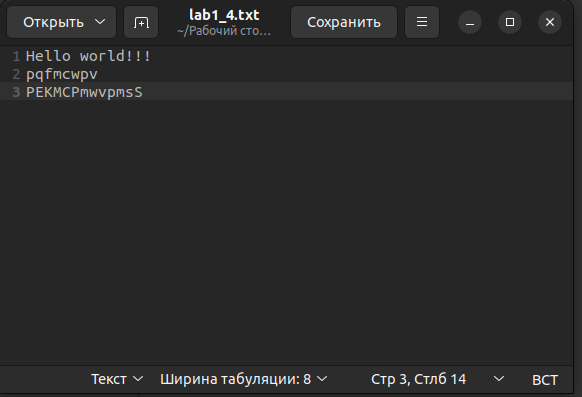


Рис. 6. Текстовый файл к упражнению 4

На рис.7 представлен результат работы программы упражнения 4.

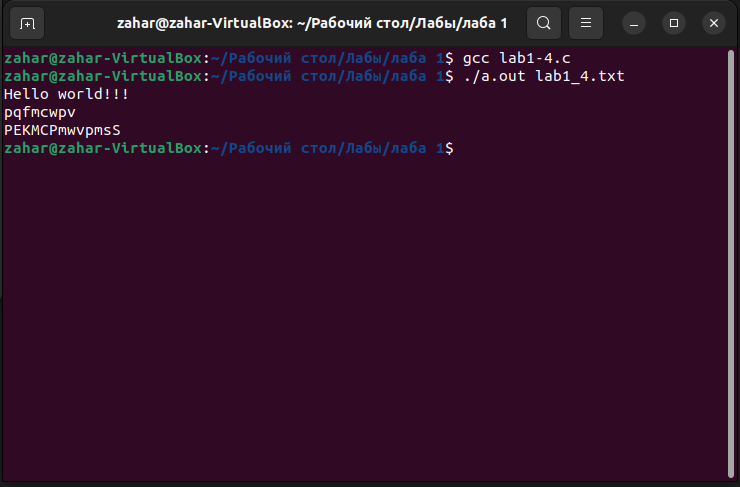


Рис. 7. Пример работы программы упражнения 4

# Упражнение 5

## 5.1 Постановка задачи

Вывести в начало текстового файла запрошенную при работе программы информацию и после дописать содержимое файла входных данных.

## 5.2 Код программы

#include <stdio.h>

void main(int argc, char \*argv[]){

FILE \*file = fopen(argv[1], "r");// открытие файла для чтения

if (file == NULL) {

perror("Ошибка при открытии файла");// Сообщение об ошибке при открытии файла(функция fopen возвращает указатель "NULL" при неудачной попытке открытия файла)

}

// Определение количества строк и их длины

int cntStr = 0;// Счетчик количества строк

int lenStr = 0;// Переменная, хранящая максимальную длину строки, содержащейся в файле, для создания массива для хранения этих строк

while (!feof(file)){// Перебор строк в файле до конца файла

if (fgetc(file) == '\n'){// Проверка на символ, указывающий на конец строки(\n)

cntStr++;

}else{

lenStr++;

}

}

// Запись содержимого файла в массив

int j = 0;

char Origintext[cntStr][lenStr];// Создание массива для хранения строк

fseek(file, 0, SEEK\_SET);// Возврат в начало файла

while(!feof(file)){

fgets(Origintext[j], lenStr-1, file);

j++;

}

fclose(file);// Закрытие файла

file = fopen(argv[1], "w");// Открытие файла для записи

printf("Введите какой-нибудь текст\n");

char txtNew[256];// Массив, хранящий введенный с клавиатуры текст

fgets(txtNew, 256, stdin);// Заполнение массива

fputs(txtNew, file);// Добавление данных из массива в файл

//Запись в файл исходных строк после строк, введенных пользователем

for (int i = 0; i< cntStr; i++){

fputs(Origintext[i], file);

}

fclose(file);

}

## 5.3 Пример работы программы

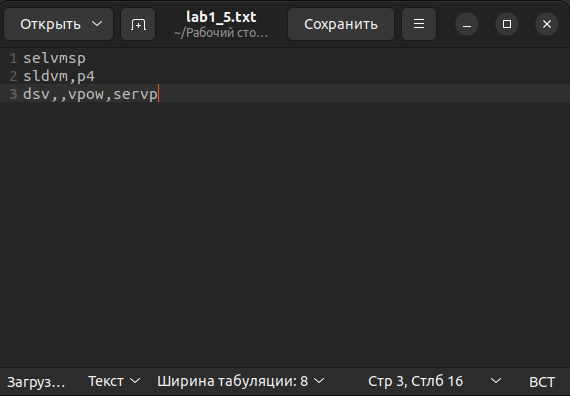


Рис. 8. Текст к упражнению 5

На рис. 9 изображена работа программы для упражнения 5.

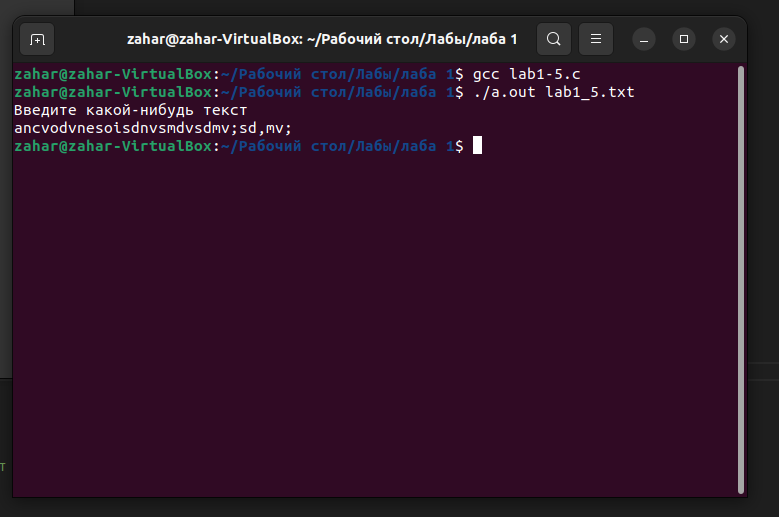


Рис. 9. Пример работы программы упражнения 5

На рис.10 показан текстовый файл после выполнения программы.

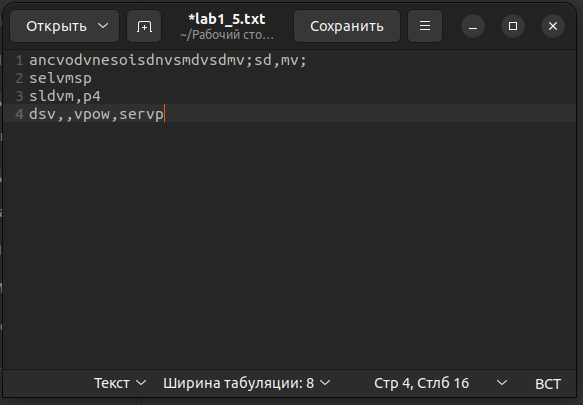


Рис. 10. Текстовый файл после выполнения кода