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 Язов М.М.

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>

int main (void) {

printf("Hello World!\n");

return 0;

}

## 1.3 Пример работы программы

На рис. 1 продемонстрирован результат работы программы 1 упражнения.

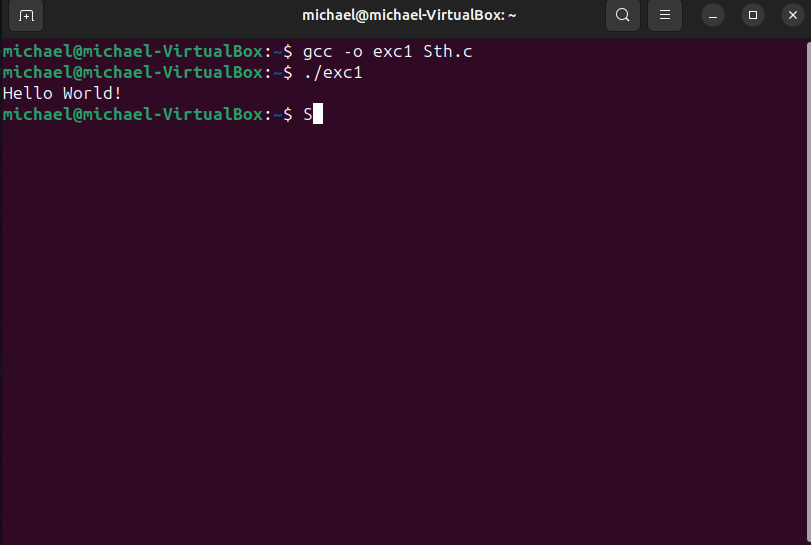


Рисунок 1. Результат программы 1 упражнения

# 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***

## 2.3. Примеры работы программы

#include <stdio.h>

int main() {

float a, b, c, x;

printf("Write a, b, c: ");

scanf("%f %f %f", &a, &b, &c);

x = (c - b)/a;

printf("Answer %f\n", x);

return 0;

}

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

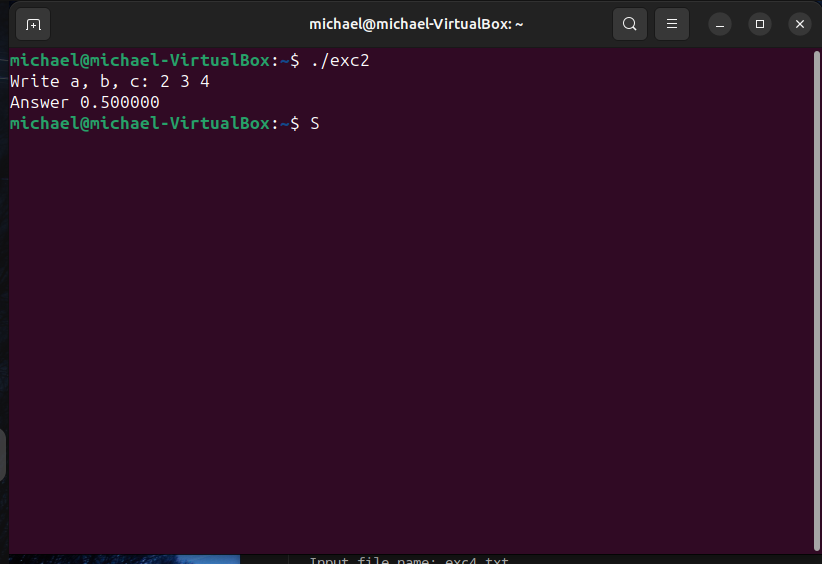


Рисунок 2. Пример 1 работы программы

Рисунок 3. Пример 2 работы программы

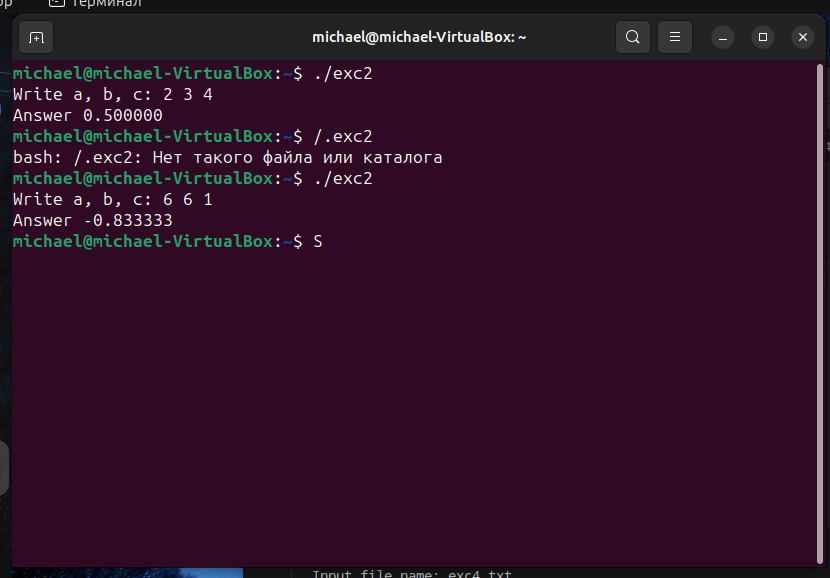
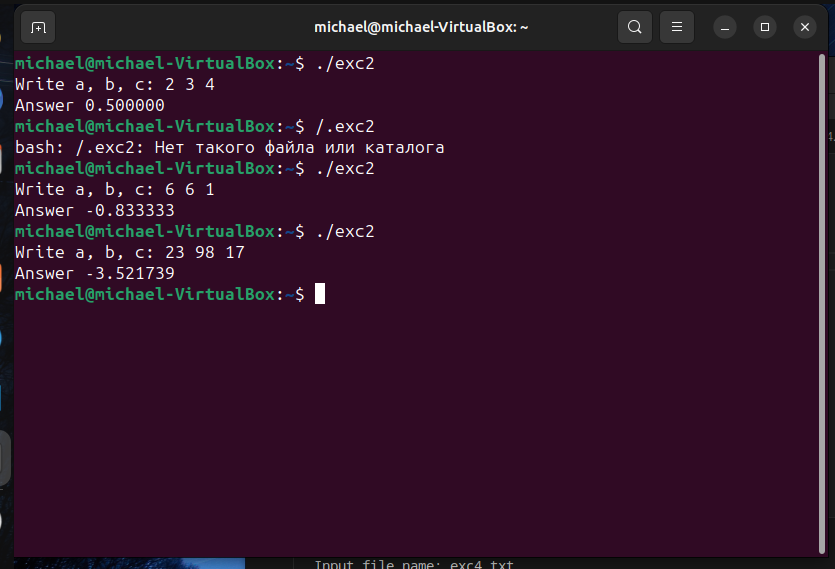


Рисунок 4. Пример 3 работы программы



# 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>

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

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

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

}

return 0;

}

## 3.3. Пример работы программы

На рис.5 представлен результат работы программы упражнения 3.

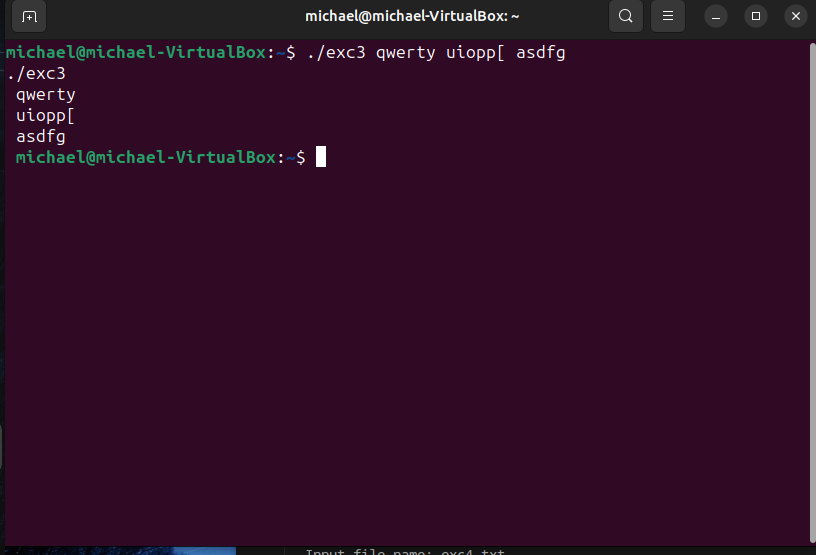


Рисунок 5. Пример работы программы упражнения 3

# 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 <string.h>

#include <stdlib.h>

int main() {

char buffer[100];

printf("Input file name: ");

fgets(buffer, sizeof(buffer), stdin);

size\_t length = strlen(buffer);

if (length > 0 && buffer[length - 1] == '\n') {

buffer[length - 1] = '\0';

}

FILE \*file = fopen(buffer, "r");

if (file == NULL) {

perror("Openning file error ");

return 1;

}

fseek(file, 0, SEEK\_END); // Перемещение в конец файла

long file\_size = ftell(file); // Получение текущей позиции

rewind(file); // Возвращаемся в начало файла

char \*buffer1 = (char \*)malloc(file\_size + 1); // Выделение памяти под хранение инфы из файла

if (buffer1 == NULL) {

perror("Memory error");

free(buffer1);

return 1;

}

size\_t result = fread(buffer1, 1, file\_size, file);

if (result != file\_size) {

perror("Reading file error");

free(buffer1);

return 1;

}

buffer1[file\_size] = '\0';

fclose(file);

printf("Данные считанные с файла: \n%s\n", buffer1);

free(buffer1);

return 0;

}

## 4.3. Пример работы программы

Содержимое текстового файла exc4.txt (рис.6):

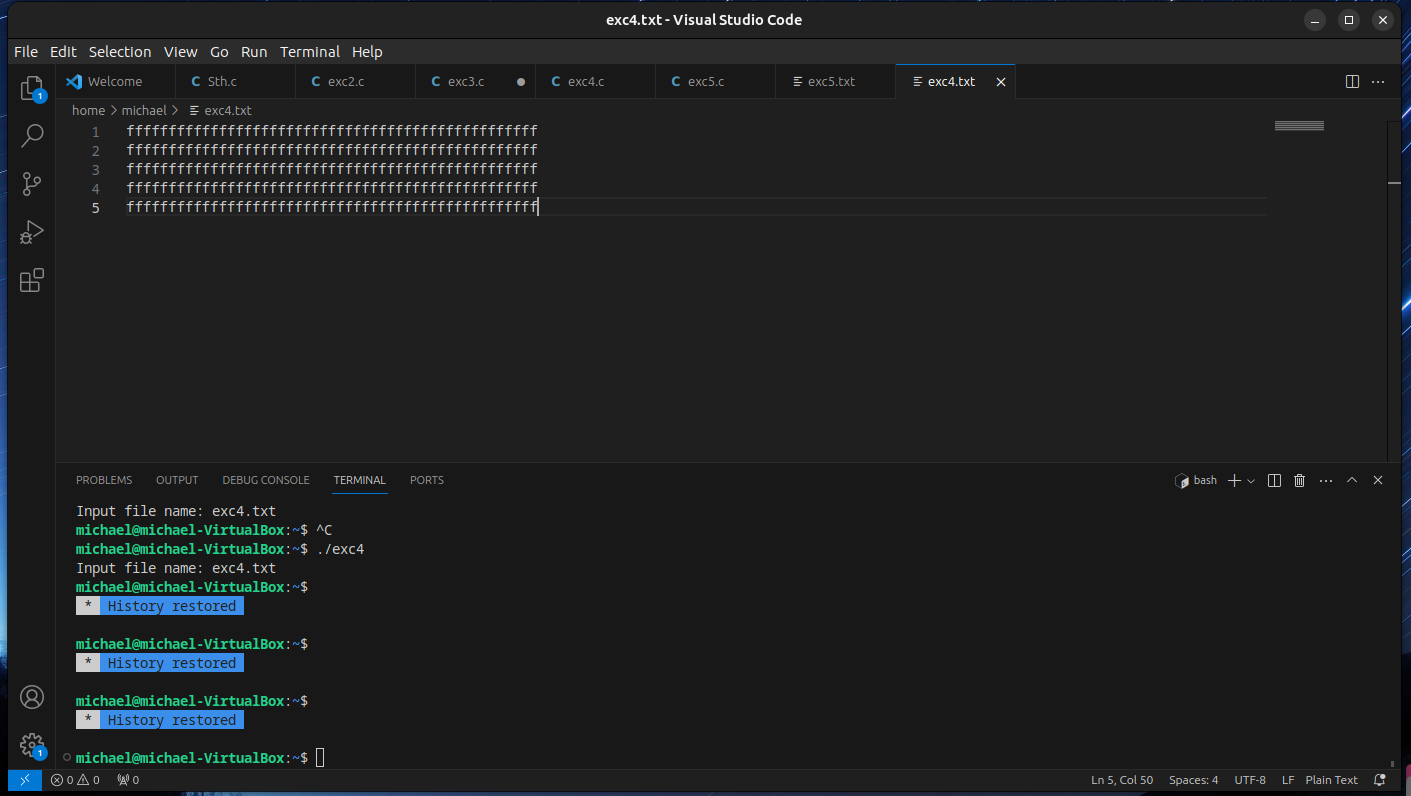


Рисунок 6. Текстовый файл к упражнению 4

На рис.7 представлен результат работы программы упражнения 4.

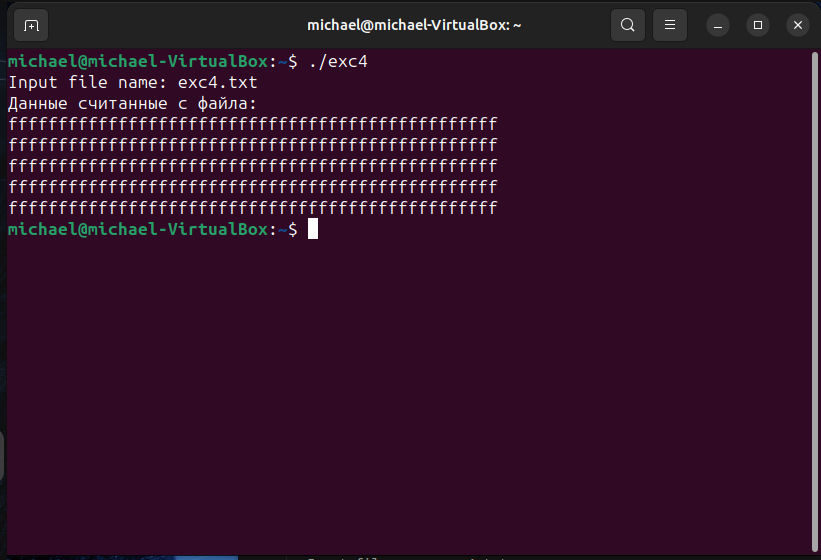


Рисунок 7. Пример работы программы упражнения 4

# 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>

#include <stdlib.h>

int main() {

FILE \*file = fopen("exc5.txt", "r");

if (file == NULL) {

perror("File error opening");

return 1;

}

fseek(file, 0, SEEK\_END); // Переходим в конец файла

long file\_size = ftell(file); // Получение текущей позиции

rewind(file); // Возвращение в начало файла

char \*buffer = (char \*)malloc(file\_size + 1); // Выделение памяти под инфу из файла

if (buffer == NULL) {

perror("Memory error");

return 1;

}

size\_t result = fread(buffer, 1, file\_size, file); // Считываем инфу из файла в буффер

if (result != file\_size) {

perror("File reading error");

free(buffer);

return 1;

}

fclose(file);

char buffer2[256];

printf("Введите строку: "); // Просим пользователя ввести строку

fgets(buffer2, sizeof(buffer2), stdin); // Считываем строку во второй буффер

file = fopen("exc5.txt", "w"); // Открываем файл для записи

fprintf(file, "%s", buffer2); // Заисываем вверх файла то что ввел пользователь

fprintf(file, "%s", buffer); // Записывем исходную информацию ниже того, что ввел пользователь

printf("Запись в файл успешно выполнена!\n");

fclose(file);

free(buffer);

return 0;

}

## 5.3 Пример работы программы

Содержимое текстового файла exc5.txt(рис.8):

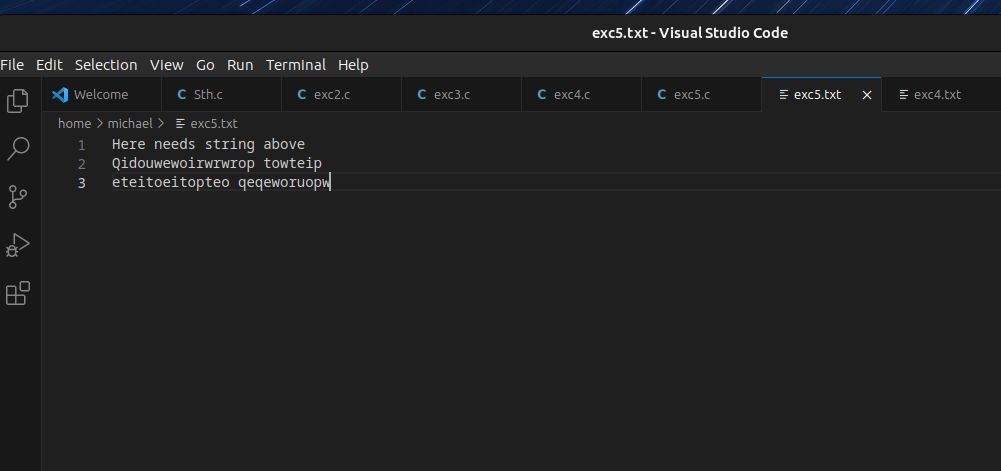


Рисунок 8. Текст к упражнению 5

На рис. 9 изображена работа программы для упражнения 5.

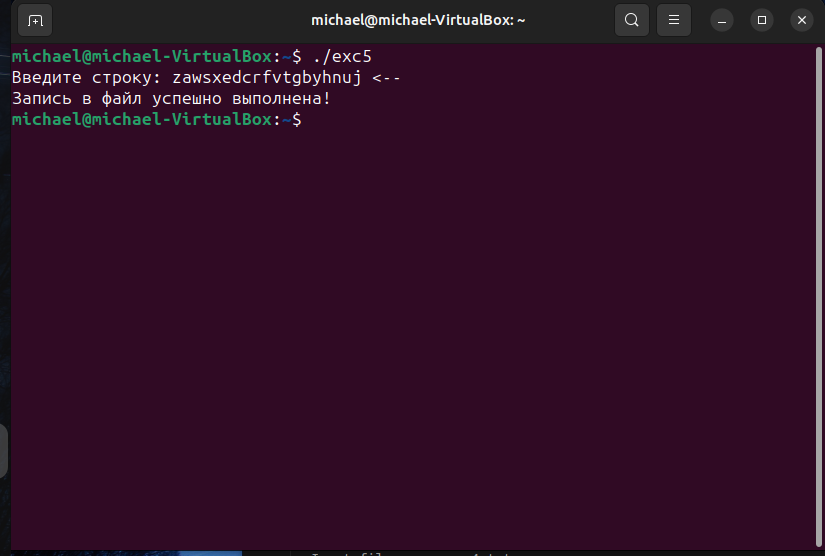


Рисунок 9. Пример работы программы упражнения 5

На рис.10 показан текстовый файл после выполнения программы.

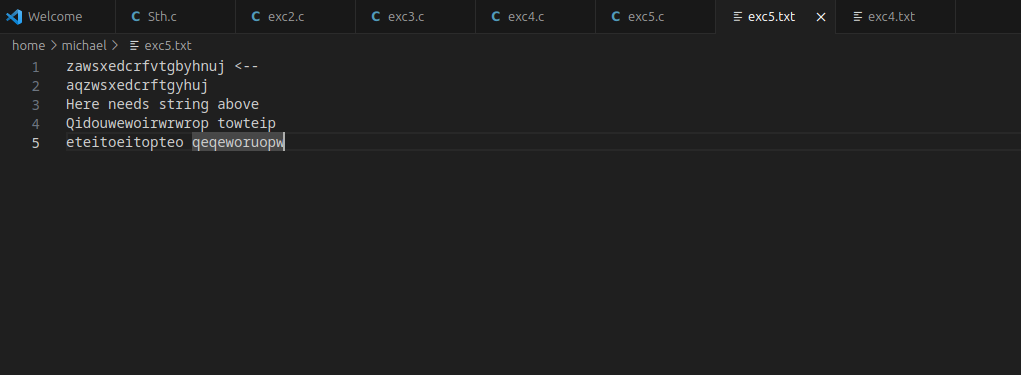


Рисунок 10. Текстовый файл после выполнения кода