МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

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

Факультет автоматики и вычислительной техники

Кафедра автоматики



**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине: «Информатика»

«ИНТЕРФЕЙС С ОПЕРАЦИОННОЙ СИСТЕМОЙ.

ФАЙЛОВЫЕ ОПЕРАЦИИ»

Вариант №13

Выполнили:Проверил:

Студенты гр. АВТ-419 Ядрышников О.Д.

Енина Ксения Сергеевна

Мерзляков Кирилл Андреевич

«19» октября 2024 г. «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

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

(оценка, подпись)

Новосибирск

2024

**Цель работы:** Ознакомиться со стандартными функциями языка СИ, используемыми для организации доступа к файлам.

**Исходные данные и задание для варианта:**

4.1. Написать программу, обрабатывающую текстовый файл и записывающую обработанные данные в файл с таким же именем, но с другим типом (табл. 4.1)

4.2. Написать программу, выполняющую посимвольную обработку текстового файла (табл. 4.2.).

Ввод параметров организовать в командной строке запуска программы.

Таблица 4.1

| Вари  ант | Задание | Параметры командной строки |
| --- | --- | --- |
| 13 | Исключить строки, начинающиеся и заканчивающиеся заданным символом | 1. Имя входного файла  2. Заданный символ |

Таблица 4.2

| Вариант | Задание | Параметры командной строки |
| --- | --- | --- |
| 13 | Заменить заданную пару букв на символы #@ | 1. Имя входного файла  2. Заданная пара букв |

**Текст программы 4.1:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

/\*

Лабораторная работа №4 - Вариант 13

Работу выполнили: Мерзляков К., Енина К.

\*/

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

/\*

argv[0] - имя запускаемого файла

argv[1] - первый параметр (имя входного файла)

argv[2] - символ, строки, начинающиеся с которого, нужно исключить

\*/

// Проверяем количество аргументов на корректность

if (argc != 3) {

printf("Использование: ./main input.txt s\n");

exit(0);

}

// Открываем входной файл на чтение

FILE \*f\_in = fopen(argv[1], "rt");

// Проверяем, что файл успешно открылся

if (f\_in == NULL) {

printf("Ошибка: Не удалось открыть файл \"%s\" на чтение\n", argv[1]);

exit(1);

}

// Собираем имя выходного файла

// "input.txt" + ".out" -> "input.txt.out"

char f\_out\_name[strlen(argv[1]) + 5];

snprintf(f\_out\_name, sizeof(f\_out\_name), "%s.out", argv[1]);

// Открываем выходной файл на запись

FILE \*f\_out = fopen(f\_out\_name, "wt");

// Проверяем, что файл успешно открылся

if (f\_out == NULL) {

printf("Ошибка: Не удалось открыть файл \"%s\" на запись\n", f\_out\_name);

exit(1);

}

// Достаем из аргумента символ,

// строки, начинающиеся с которого, нужно исключить

char exclude\_char = argv[2][0];

// Создаем строку, в которую будем считывать каждую из строк файла

char line[128];

// Считываем строки, пока удается считать: fscanf() == 1

// когда не удастся: fscanf() == 0

while (fscanf(f\_in, "%127s\n", line) == 1) {

// если строка начинается или заканчивается на exclude\_char,

// то переходим к следующей (не записываем в выходной файл)

if (line[0] == exclude\_char || line[strlen(line) - 1] == exclude\_char) {

continue;

}

fprintf(f\_out, "%s\n", line);

}

// закрываем файлы

fclose(f\_in);

fclose(f\_out);

printf("Обработанные данные были записаны в файл \"%s\"\n", f\_out\_name);

return 0;

}

**Текст программы 4.2:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

/\*

Лабораторная работа №4 - Вариант 13

Работу выполнили: Мерзляков К., Енина К.

\*/

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

/\*

argv[0] - имя запускаемого файла

argv[1] - первый параметр (имя входного файла)

argv[2] - символы, которые нужно заменить на "#@"

\*/

// Проверяем количество аргументов на корректность

if (argc != 3) {

printf("Использование: ./main input.txt zx\n");

exit(0);

}

// Открываем входной файл на чтение

FILE \*f\_in = fopen(argv[1], "rt");

// Проверяем, что файл успешно открылся

if (f\_in == NULL) {

fprintf(stderr, "Ошибка: Не удалось открыть файл \"%s\" на чтение\n", argv[1]);

exit(1);

}

// Собираем имя выходного файла

// "input.txt" + ".out" -> "input.txt.out"

char f\_out\_name[strlen(argv[1]) + 5];

snprintf(f\_out\_name, sizeof(f\_out\_name), "%s.out", argv[1]);

// Открываем выходной файл на запись

FILE \*f\_out = fopen(f\_out\_name, "wt");

// Проверяем, что файл успешно открылся

if (f\_out == NULL) {

fprintf(stderr, "Ошибка: Не удалось открыть файл \"%s\" на запись\n", f\_out\_name);

exit(1);

}

// Достаем из аргумента символы,

// которые нужно заменить

const char replace\_1 = argv[2][0];

const char replace\_2 = argv[2][1];

// Замена последовательности двух введенных выше символов на "#@"

int prev\_c = EOF;

char cur\_c;

while ((cur\_c = fgetc(f\_in)) != EOF) {

if (prev\_c == replace\_1 && cur\_c == replace\_2) {

fputc('#', f\_out);

fputc('@', f\_out);

prev\_c = EOF;

} else {

if (prev\_c != EOF) {

fputc(prev\_c, f\_out);

}

prev\_c = cur\_c;

}

}

if (prev\_c != EOF) {

fputc(prev\_c, f\_out);

}

// Закрываем файлы

fclose(f\_in);

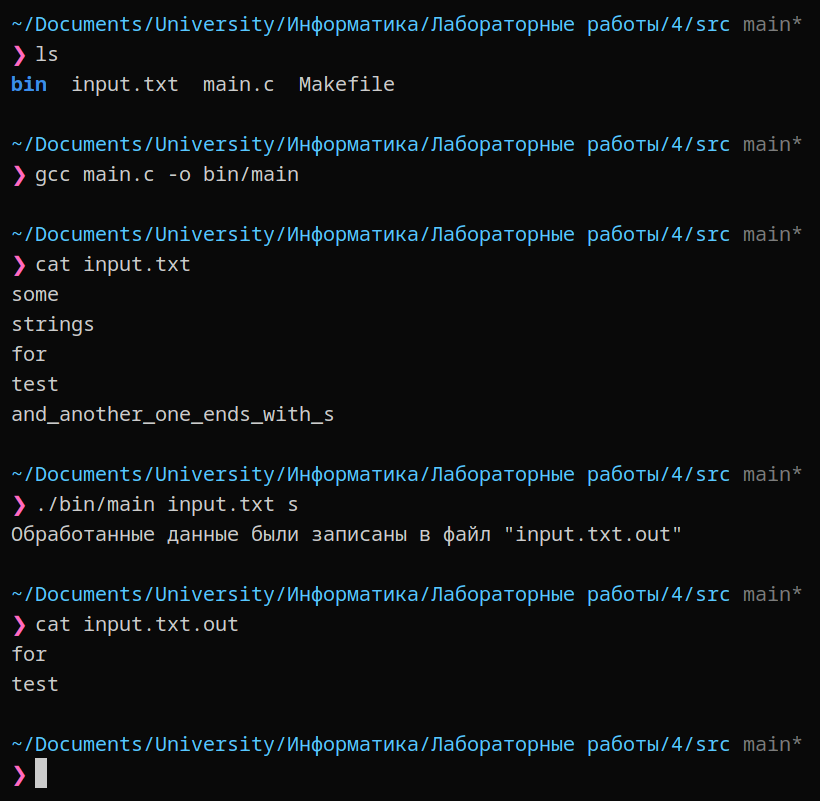
fclose(f\_out);

printf("Обработанные данные были записаны в файл \"%s\"\n", f\_out\_name);

return 0;

}

**Результаты (4.1):**



**Результаты (4.2):**



**Вывод:** Мы ознакомились со стандартными функциями языка СИ, используемыми для организации доступа к файлам и написали программу, в которой применили полученные навыки.