



Итоговая работа по курсу «Программирование на языке C (базовый уровень)»

Иванов Юрий Владимирович

2024 год

Оглавление

Постановка задания	2
Описание программы	4
Исходный код программы	4
Описание файлов программы	4
Описание работы программы	4
Примеры написания командной строки для запуска:	6
Демонстрация работы программы	7
Сборка программы утилитой make	8

Постановка задания

Создание консольного приложения — Статистика температуры

Необходимо реализовать консольное приложение, которое осуществляет считывание текстового файла csv, состоящего из строк следующего формата:

YEAR;MONTH;DAY;HOUR;MINUTE;TEMPERATURE

dddd;mm;dd;hh;mm;temperature

dddd - год 4 цифры

mm - месяц 2 цифры

dd - день 2 цифры hh

- часы 2 цифры mm -

минуты 2 цифры

temperature - целое число от -99 до 99

В архиве файле хранится статистика собранная датчиком температуры за 1 календарный год. Предполагается, что датчик собирал информацию не чаще чем 1 раз в минуту и сохранял в заданном формате каждое значение в текстовый файл с новой строки.

В какой-то момент времени датчик мог не работать, тогда данные поэтому периоду могут отсутствовать. Пример входного файла:

YEAR	MONTH	DAY	HOUR	MINUTE	TEMPERATURE
2021	1	1	23	1	-5

2021	1	1	23	3	-6
2021	1	1	23	1	-7
2021	1	2	7	5	-10

Требования к обработке данных

Необходимо вывести статистику по каждому месяцу, с учетом исходных данных:

- среднемесячная температура
 - минимальная температура в текущем месяце
 - максимальная температура в текущем месяце
- Также необходимо вывести статистику за год:
- среднегодовая температура
 - минимальная температура
 - максимальная температура

Требования к аргументам командной строки

Приложение должно обрабатывать аргументы командной строки: минимальный набор поддерживаемых ключей:

- -h Описание функционала приложения. Список ключей, которые обрабатывает данное приложение и их назначение.
- -f <filename.csv> входной файл csv для обработки.
- -m <номер месяца> если задан данный ключ, то выводится только статистика за указанный месяц.
- если нет параметров, то выдается help

Требования к ошибкам в входных данных

- Приложение должно корректно работать на любых входных данных, если формат csv файла не соответствует заданному, то необходимо указать номер строки файла csv, в которой обнаружена ошибка и не учитывать данную строку.

В архиве с заданием [temperature_data_examples.zip](#) лежат два файла:

- temperature_big.csv — файл со статистикой за год
- temperature_small.csv — укороченный файл с ошибками для тестирования

Требования к сборке приложения

- Приложение должно собираться при помощи утилиты make.
- Все прототипы функций, используемые в приложении, должны быть вынесены в отдельный файл temp_functions.h

- Тексты функций — в файл temp_functions.c
- Для реализации приложения рекомендуется использовать массив из структурного типа данных для хранения показаний датчика.

Описание программы

Исходный код программы

https://github.com/KotovladeletsGT/Kursach_MFTI

Описание файлов программы

main.c – основной файл программы с точкой входа, функцией main. Содержит только логику запуска функций и обработку ключей, поступающих от командной строки;

temp_function.h – файл прототипов функций программы;

temp_function.c – файл описания функций программы;

banner_function.h – файл прототипов функций печати;

banner_function.c – файл описания функций печати;

Makefile – файл инструкция для утилиты сборки mingw32-make;

report.exe – собранный исполняемый файл программы.

Описание работы программы

Программа является консольным приложением и рекомендуется запускать её из командной строки.

При запуске без указания ключей, программа предложит ввести ключ «-h» для получения инструкций.

Программа допускает применение следующих ключей:

«-h» - получение информации о возможных ключах запуска с кратким описанием их назначения и о возможностях программы;

«-f file_name.csv» - указание файла для обработки, где file_name – имя файла. Если в этом режиме не добавлена опция «-m», то выведется полная статистика за год;

«-m xx» - указание месяца для получения статистики по конкретному месяцу, где xx – месяц (1 - январь, 12 - декабрь);

report – программа предложит ввести «-h»

report -h – программа выведет справку

report -f temperature_big.csv -m 3 – программа обработает файл и выведет статистику за третий месяц – март

report -f temperature_big.csv – программа обработает файл и сперва выведет статистику за весь год суммарно – а затем, по каждому месяцу с января по декабрь

Демонстрация работы программы

Прикрепите в окошке ниже снимки работы программы. Подпишите каждый снимок.

Образцы снимков:

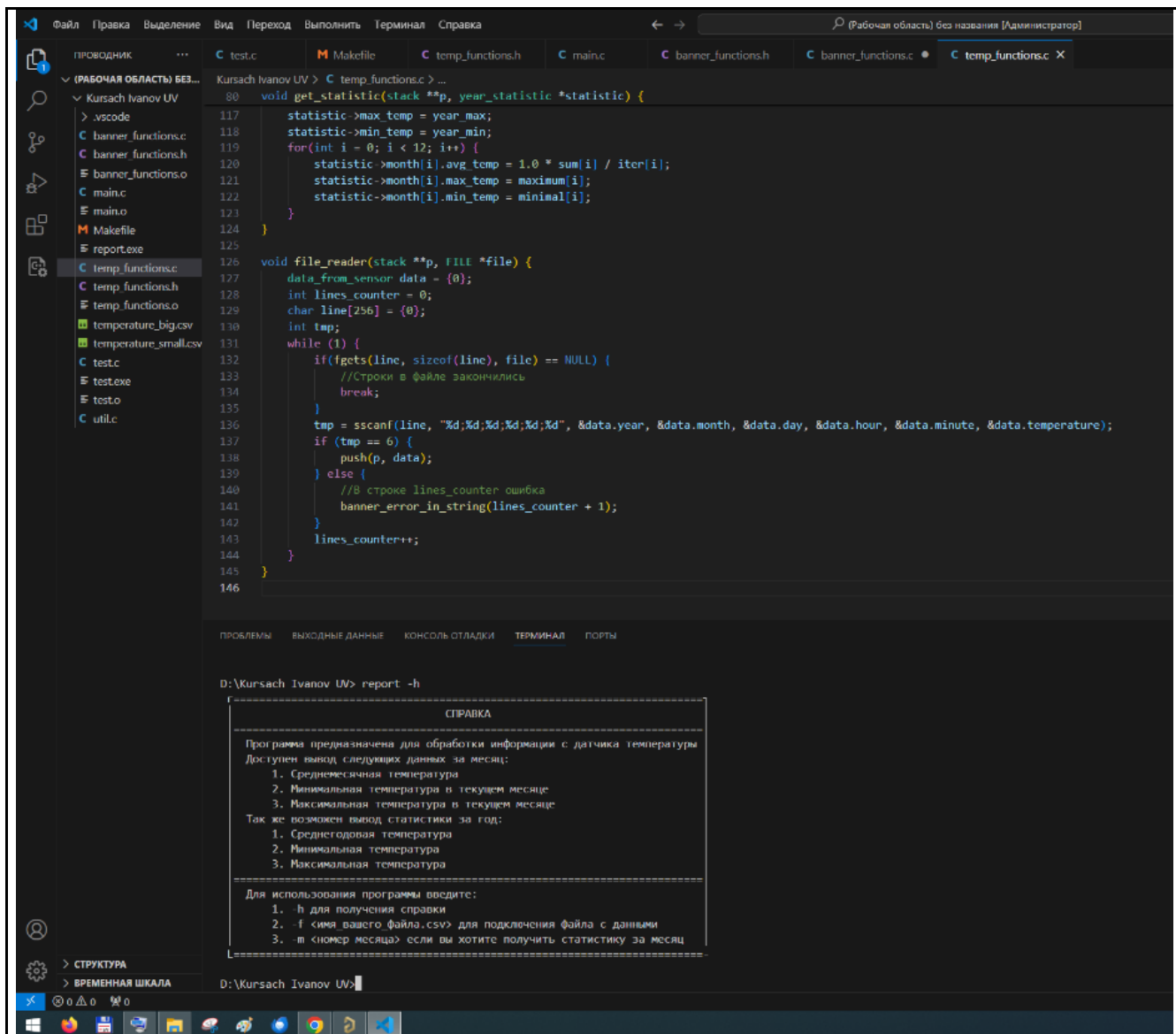


Рисунок 1. Запуск программы с вводом ключа -h. Программа вывела справку о себе

Рисунок 2. Запуск программы с ключём -f temperature_big.csv -m 3
Программа вывела статистику за март

The screenshot shows the Visual Studio Code interface. The left sidebar displays the file explorer with a project structure including files like `main.c`, `banner_functions.c`, and `temperature_big.csv`. The main editor window shows the source code of `main.c`, which includes logic for parsing command-line arguments and reading data from a CSV file. The bottom panel shows the terminal window with the command `D:\Kursach Ivanov UV> report -f temperature_big.csv -m 3` and its output, which displays statistics for the month of March.

```
int main(int argc, char** argv) {
    .help_flag = 0,
    .month = 0,
    .month_flag = 0,
    .error = 0
};

setlocale(LC_ALL, "rus");

get_flags(&flag, argc, argv);

if(flag.error) {
    return 0;
}

if((flag.file_flag || flag.help_flag || flag.month_flag) == 0) {
    banner_get_starting();
    return 0;
}

if (flag.help_flag) {
    banner_help();
    return 0;
}

if (flag.file_flag) {
    FILE *my_file = fopen(flag.file_path, "r");
    stack *all_data = NULL;
    year_statistic statistic = {0};
    if (my_file == NULL){
        banner_error_reading();
        fclose(my_file);
        return 0;
    } else {
        file_reader(&all_data, my_file);
        get_statistic(&all_data, &statistic);
    }

    if (flag.month_flag) {
        banner_month_statistic(&statistic, flag.month);
    } else {

```

Д:\Kursach Ivanov UV> report -f temperature_big.csv -m 3

```
=====
В месяце March термометр зафиксировал:
1. Среднемесячную температуру 15,500
2. Максимальную температуру 30
3. Минимальную температуру 1
=====
Д:\Kursach Ivanov UV>
```

Рисунок 3. Запуск программы с ключём -f temperature_big.csv

Программа вывела статистику за год



```
1 #include "temp_functions.h"
2 #include "banner_functions.h"
3
4
5
6 //После получения флагов решаем что делать:
7 //Если одновременно есть ошибка и пользователь ничего не ввёл - выводим справку
8 int main(int argc, char** argv) {
9
10     flags flag = {
11         .file_path = NULL,
12         .file_flag = 0,
13         .help_flag = 0,
14         .month = 0,
15         .month_flag = 0,
16         .error = 0
17     };
18
19     setlocale(LC_ALL, "Rus");
20
21     get_flags(&flag, argc, argv);
```

```
D:\Kursach Ivanov UV> report -f temperature_big.csv
-----
За текущий календарный год температура составила:
1. В среднем          15,500
2. Максимально достигла 30
3. Минимум был        1
-----
Ниже представлена собранная статистика за каждый месяц
-----
В месяце  Январь термометр зафиксировал:
1. Среднемесячную температуру  15,500
2. Максимальную температуру    30
3. Минимальную температуру     1
-----
В месяце  February термометр зафиксировал:
1. Среднемесячную температуру  15,500
2. Максимальную температуру    30
3. Минимальную температуру     1
-----
В месяце  March термометр зафиксировал:
1. Среднемесячную температуру  15,500
2. Максимальную температуру    30
3. Минимальную температуру     1
-----
В месяце  April термометр зафиксировал:
1. Среднемесячную температуру  15,500
2. Максимальную температуру    30
3. Минимальную температуру     1
-----
```
