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

Зеленин Евгений Владимирович

Оглавление

[Постановка задания 3](#_heading=h.3znysh7)

[Описание программы 5](#_heading=h.2et92p0)

[Исходный код программы 5](#_heading=h.tyjcwt)

[Описание файлов программы 5](#_heading=h.3dy6vkm)

[Описание работы программы 6](#_heading=h.1t3h5sf)

[Примеры написания командной строки для запуска: 6](#_heading=h.4d34og8)

[Демонстрация работы программы 7](#_heading=h.2s8eyo1)

[Сборка программы утилитой make 8](#_heading=h.17dp8vu)

# 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 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](https://drive.google.com/file/d/1BBWwFSzVtGRPgvout8EmWXsdVUEVRtYG/view?usp=drive_link) лежат два файла:

* temperature\_big.csv — файл со статистикой за год
* temperature\_big\_errors.csv — файл со статистикой за год с ошибками
* 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/molnija3d/m3d_mipt_study/tree/main/CW> * <https://github.com/molnija3d/m3d_mipt_study/tree/List-version/CW> - реализация с двусвязным списком вместо массива |

## Описание файлов программы

Проект состоит из следующих директорий:

Папка “.\bin” – содержит исполняемые файлы и входные данные;

Папка “.\include” – содержит файлы заголовков;

Папка “.\src” – содержит исходные коды;

В корне проекта расположен файл для сборки – Makefile.

Программа состоит из следующих файлов:

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

include\temp\_api.h – файл прототипов функций программы;

src\temp\_api.c – файл описания функций программы;

include\temp\_sys.h – файл прототипов вспомогательных функций программы;

src\temp\_sys.c – файл описания вспомогательных функций программы;

makefile – файл с инструкциями по сборке для утилиты mingw32-make;

tmp\_stat.exe – исполняемый файл программы.

## Описание работы программы

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

При запуске без указания ключей, программа кратко выведет информацию о своем назначении и предложении ввести ключ «-h» для получения инструкций.

Программа проверяет входные данные на соответствие спецификации. Когда значения полей не соответствуют диапазону, в консоль выводится сообщение с номером позиции и номером строки, в которой найдена ошибка. Код 9 выводится во всех остальных случаях.

Ошибочные строки не участвуют в дальнейшей обработке.

Программа допускает применение следующих ключей:

* «-h» - получение информации о возможных ключах запуска с кратким описанием их назначения;
* «-f file\_name» - указание файла для обработки, где file\_name – имя файла. Если в этом режиме не добавлена опция «-m», то выведется полная статистика по всему файлу;
* «-m xx» - указание месяца для получения статистики по конкретному месяцу, где xx - месяц;
* сборка программы в режиме отладки осуществляется командой “mingw32-make.exe -f .\Makefile\_debug”

### Примеры написания командной строки для запуска:

tmp\_stat.exe

tmp\_stat.exe –h

tmp\_stat.exe –f temperature\_small.csv

tmp\_stat.exe –f temperature\_big.csv –m 3

tmp\_stat.exe –f temperature\_big\_errors.csv –m 3

Демонстрация работы программы

|  |
| --- |
| ***Образцы снимков:*** |
| Рисунок 1. Запуск программы с опцией «-h»    Рисунок 2. Запуск программы с указанием файла и опцией выбора месяца    Рисунок 3. Запуск программы с указанием имени файла |

### Сборка программы утилитой make

|  |
| --- |
| *Прикрепите в окошке ниже снимок сборки программы утилитой make.* |
| Рисунок 6 – Сборка с помощью утилиты make    Рисунок 5 – Очистка от файлов предыдущих сборки    Рисунок 6 – сборка приложения с опцией -debug для отладки |