



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

**КУДИНОВ АЛЕКСАНДР
АЛЕКСАНДРОВИЧ**

Оглавление

Постановка задания	3
Описание программы	5
Исходный код программы	5
Описание файлов программы	5
Описание работы программы	6
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/Alkud/MIPT_IoT_C_Basic/tree/main/COURSEWORK

Описание файлов программы

Программа состоит из следующих файлов:

`main.c` – основной файл программы с точкой входа, функцией `main`. Содержит только логику запуска функций;

`include/program_arguments.h` – состав, описание и функции обработки аргументов командной строки.

`include/data_types.h` – описание структурных типов данных, используемых при обработке дат и значений температуры.

`include/csv_reader.h` – прототипы функций чтения данных из csv-файлов.

`include/temp_api.h` – прототипы функций, используемых для обработки данных температуры (сортировка, вывод на печать, сравнение, поиск, добавление новых данных)

`src/csv_reader.c` – описание функций чтения csv-файлов.

`src/temp_api.c` – описание функций, используемых для обработки данных.

`Makefile` – файл инструкция для утилиты сборки `make` (Linux).

`/bin/temp_stat_cli` – собранный исполняемый файл программы.

Описание работы программы

Программа является консольным приложением, запускается из командной строки.

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

Обязательный аргумент «-f» - имя файла.

Если кроме него другие аргументы не переданы, будет выведена статистика температуры по месяцам и по годам.

Если требуется только статистика за конкретный год, месяц или день, нужно указать, соответственно набор аргументов:

- «-y» если нужна статистика за конкретный год,
- «-y» и «-m» если нужна статистика за конкретный месяц конкретного года,
- «-y», «-m» и «-d», если нужна статистика за конкретный день конкретного месяца конкретного года,

Если требуется статистика за определённый интервал, нужно указать начало («-y», «-m» и «-d») интервала и конец («-Y», «-M» и «-D») интервала.

Начало и конец задаются с любой – но обязательно одинаковой – точностью – до дня, месяца или года.

Также дополнительно возможно посмотреть по N записей с минимальными и максимальными значениями температур.

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 добавлены опции «-y», «-m», то выведется полная статистика по всему файлу;

«-y xxxx» - указание года для получения статистики по конкретному году, либо начало интервала, где xxxx – год;

«-m xx» - указание месяца для получения статистики по конкретному месяцу (либо начало интервала), где xx – месяц;

«-d xx» - указание дня для получения статистики по конкретному дню (либо начало интервала), где xx – день;

«-Y xxxx» - указание года конца интервала для получения статистики, где xxxx – год;

«-M xx» - указание месяца конца интервала для получения статистики, где xx – месяц;

«-D xx» - указание дня конца интервала для получения статистики, где xx – день;

«-t xx» - дополнительный вывод по xx записей с минимальными и максимальными значениями температур.

Примеры написания командной строки для запуска:

```
temp_stat_cli
```

```
temp_stat_cli -h
```

```
temp_stat_cli -f temperature_big.csv
```

```
temp_stat_cli -f temperature_big.csv -y 2020 -m 4 -d 12
```

```
temp_stat_cli -f temperature_big.csv -y 2020 -m 4 -Y 2022 -m 11
```

```
temp_stat_cli -f temperature_big.csv -t 10
```

Демонстрация работы программы

Рисунок 1. Запуск программы с опцией «-h»

```
• (base) user@user-XPS-Ubuntu:~/Documents/mipt_iot_homeworks/MIPT_IoT_C_Basic/COURSEWORK$ ./bin/temp_stat_cli -h
Чтение данных температурного датчика. Вывод статистики по месяцам или за указанный период
Аргументы командной строки:
  -h      вывод справки
  -f      имя файла с данными
  -t {N}  дополнительно: сортировка по температуре
           и вывод N самых холодных и самых тёплых дней (1 по умолчанию)
интервал статистики задаётся:
  либо годом, месяцем и днём
  -y YYYY год (опционально)
  -m MM   месяц (опционально)
  -d DD   день (опционально)
  либо границами интервала [y, m, d](начало), [Y, M, D](конец)
  -y YYYY год (опционально)
  -m MM   месяц (опционально)
  -d DD   день (опционально)
  -Y YYYY год (обязательно)
  -M MM   месяц (опционально)
  -D DD   день (опционально)
```

Рисунок 2. Запуск программы с указанием файла и опцией выбора месяца

```
• (base) user@user-XPS-Ubuntu:~/Documents/mipt_iot_homeworks/MIPT_IoT_C_Basic/COURSEWORK$ ./bin/temp_stat_cli -f stat_test_data.csv -y 2020 -m 4
Статистика температуры за 2020.04:
мин.    : -20,00 градусов
макс.   : +1,00 градусов
средняя : -6,00 градусов
```

Рисунок 3. Запуск программы с указанием имени файла

```
• (base) user@user-XPS-Ubuntu:~/Documents/mipt_iot_homeworks/MIPT_IoT_C_Basic/COURSEWORK$ ./bin/temp_stat_cli -f stat_test_data.csv
Статистика месячных температур:
год.мес. | мин. | макс. | сред.
-----+-----+-----+-----
2020.01 | -15,00 | -9,00 | -11,25
2020.02 | -15,00 | +1,00 | -7,00
2020.03 | -10,00 | -6,00 | -7,75
2020.04 | -20,00 | +1,00 | -6,00
2020.05 | +3,00 | +17,00 | +9,75
2020.06 | +20,00 | +28,00 | +24,50
2020.07 | +22,00 | +29,00 | +25,25
2020.08 | +21,00 | +28,00 | +23,50
-----+-----+-----+-----

Статистика годовых температур:
год | мин. | макс. | сред.
---+---+---+---
2020 | -20,00 | +29,00 | +6,38
-----+-----+-----+-----
```

Рисунок 4. Запуск программы с указанием файла и опцией выбора интервала для вывода статистики

```
• (base) user@user-XPS-Ubuntu:~/Documents/mipt_iot_homeworks/MIPT_IoT_C_Basic/COURSEWORK$ ./bin/temp_stat_cli -f stat_test_data.csv -y 2020 -m 4 -Y 2020 -M 7
Статистика температуры за период 2020.04 - 2020.07:
мин.    : -20,00 градусов
макс.   : +29,00 градусов
средняя : +13,38 градусов
```


Рисунок 5. Запуск программы с указанием файла и опцией вывода минимальных и максимальных температур

```

• (base) user@user-XPS-Ubuntu:~/Documents/mipt_iot_homeworks/MIPT_IoT_C_Basic/COURSEWORK$ ./bin/temp_stat_cli -f stat_test_data.csv -t 5
Статистика месячных температур:
год.мес. | мин. | макс. | сред.
-----+-----+-----+-----
2020.01 | -15,00 | -9,00 | -11,25
-----+-----+-----+-----
2020.02 | -15,00 | +1,00 | -7,00
-----+-----+-----+-----
2020.03 | -10,00 | -6,00 | -7,75
-----+-----+-----+-----
2020.04 | -20,00 | +1,00 | -6,00
-----+-----+-----+-----
2020.05 | +3,00 | +17,00 | +9,75
-----+-----+-----+-----
2020.06 | +20,00 | +28,00 | +24,50
-----+-----+-----+-----
2020.07 | +22,00 | +29,00 | +25,25
-----+-----+-----+-----
2020.08 | +21,00 | +28,00 | +23,50
-----+-----+-----+-----

Статистика годовых температур:
год | мин. | макс. | сред.
-----+-----+-----+-----
2020 | -20,00 | +29,00 | +6,38
-----+-----+-----+-----

Сортировка по температуре:
самые холодные дни:
-----+-----+-----+-----
2020.04.03 | 00:01 | -20
-----+-----+-----+-----
2020.01.03 | 00:02 | -15
-----+-----+-----+-----
2020.02.11 | 00:01 | -15
-----+-----+-----+-----
2020.01.14 | 00:01 | -11
-----+-----+-----+-----
2020.01.01 | 00:01 | -10
-----+-----+-----+-----
самые тёплые дни:
-----+-----+-----+-----
2020.06.29 | 00:02 | +26
-----+-----+-----+-----
2020.07.10 | 00:02 | +27
-----+-----+-----+-----
2020.06.21 | 00:03 | +28
-----+-----+-----+-----
2020.08.25 | 01:03 | +28
-----+-----+-----+-----
2020.07.18 | 00:01 | +29
-----+-----+-----+-----

```

Сборка программы утилитой make

```
• (base) user@user-XPS-Ubuntu:~/Documents/mipt_iot_homeworks/MIPT_IoT_C_Basic/COURSEWORK$ make  
gcc -std=c99 -ggdb -I include src/csv_reader.c src/temp_api.c main.c -o bin/temp_stat_cli
```