МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ»

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ФАКУЛЬТЕТ №8 «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРИКЛАДНАЯ МАТЕМАТИКА»

КАФЕДРА 806 «ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ПРОГРАММИРОВАНИЕ»

**Курсовой проект**

На тему

«Обработка последовательной файловой структуры на Си»

|  |  |
| --- | --- |
| Курс / Семестр: | 1 / 2 |
| Группа: | М8О-108Б-19 |
| ФИО студента: | Горохов М.А. |
| ФИО преподавателя: | Поповкин А.В. |
| Подпись: |  |
| Оценка: |  |
| Дата сдачи: |  |
| Дата проверки: |  |

Москва

2020

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc19304886)

[1. ЦЕЛИ ПРОЕКТА](#_Toc19304887) 4

[2. ПОСТАНОВКА ЗАДАЧИ 5](#_Toc19304890)

[3. СТРУКТУРА ПРОЕКТА 6](#_Toc19304894)

[4. ПРОГРАММНЫЙ КОД И ТЕСТОВЫЕ ДАННЫЕ 8](#_Toc19304894)

[5. ЗАКЛЮЧЕНИЕ](#_Toc19304891) 11

[6. ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ](#_Toc19304898) 12

# ВВЕДЕНИЕ

В программировании есть много случаев, когда может понадобиться больше одной переменной для представления определённого объекта. В этих случаях полезно использовать структуры данных, позволяющие сгруппировывать переменные разных типов в единое целое. Польза структур данных продемонстрирована в этом курсовом проекте на примере языка программирования Си и на примере задания о составлении сведений состава комплектующих ПЭВМ в студенческой группе.

# ЦЕЛИ ПРОЕКТА

Изучить структуры данных на языке программирования Си на примере реализации простой структуры данных для ПЭВМ студентов. Научиться хранить данные о ПЭВМ как в текстовых файлах, так и в бинарных.

# ПОСТАНОВКА ЗАДАЧИ

Составить последовательную структуру данных для представления простейшей базы данных на файлах в СП Си на примере сведений о составе комплектующих личных ПЭВМ в студентческой группе. Реализовать программу, сохраняющая данные в бинарный файл, и программу, выводящая данные из бинарного файла в приемлимый для пользователя вид, то есть в виде таблицы.

Программа должна уметь выводить аннотированный список неукомплектованных компьютеров

# СТРУКТУРА ПРОЕКТА

Проект состоит из 5 файлов: программы, сохраняющей данные в бинарный файл ***input.c***, программы, выводящей содержимое бинарного файла в приемлимы вид ***output.c***, заголовочного файла с описанием структуры ***studentpc.h***, входного файла ***data1.txt***, и бинарного файла ***data1.bin***, генерирующегося в результате работы первой программы.

Структура содержит следующие пункты: Owner\_Name, CPU\_Model, CPU\_Cores, CPU\_Clocks, RAM\_Model, RAM\_Value, GPU\_Model, GPU\_Value, Storage\_Model, Storage\_Value, OS, Is\_Completed?.

Всего 12 пунктов.

Формат программы ***input.c***:

*./input <Входной текстовый файл> <Выходной бинарный файл>*

Пример работы программы ***input.c***:

[revammark@Raft kp1]$ date

Чт 23 апр 2020 11:42:54 MSK

[revammark@Raft kp1]$ ./input

Введите ещё два аргумента в формате:

./input <Входной текстовый файл> <Выходной бинарный файл>

[revammark@Raft kp1]$ cat data1.txt

Gorokhov\_Mikhail AMD\_Ryzen\_3\_2200U 4 2500 Crucial\_CT32G4RFD4266 32000 Ryzen\_3\_2200U 0 SSD 128000 Arch\_Linux y

Ivanov\_Fedor Intel\_i3\_11580 20 5000 Crucial\_CT64G4RFD4266 128000 Geforce\_2080Ti 20000 SSD 256000 Linux\_From\_Scratch y

Keker Celeron\_3060 1 700 No 0 No 0 HDD 60000 Windows\_95 n

Antoshka Arduino\_Uno 1 1400 Samsung\_DDR2 128 No 0 SSD 512 Linux n

Ivanov\_Fedor Intel\_i3\_11580 20 5000 Crucial\_CT64G4RFD4266 128000 No 0 SSD 256000 Linux\_From\_Scratch n

Gorokhov\_Mikhail AMD\_Ryzen\_5\_2600X 12 3200 Crucial\_CT32G4RFD4266 16000 Radeon\_470 4096 SSD 256000 Linux\_From\_Scratch y[revammark@Raft kp1]$

[revammark@Raft kp1]$ ./input data1.txt data1.bin

[revammark@Raft kp1]$ ls

data1.bin data1.txt input input.c k6789-2013.djvu kp1\_magorkhov.docx output output.c studentpc.h test test.c

Формат работы программы ***output.c***:

*./output <Бинарный файл> <ключ -f/-p>*

-f - вывести аннотированный список всех компьютеров.

-p - вывести аннотированный список неукомплектованных компьютеров.

Пример работы программы ***./output.c -f***:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Owner\_Name | CPU\_Model | CPU\_Cores | CPU\_Clocks | RAM\_Model | RAM\_Value | GPU\_Model | GPU\_Value | Storage\_Model | Storage\_Value | OS | Is\_Completed? |
| Gorokhov\_Mikhail | AMD\_Ryzen\_3\_2200U | 4 | 2500MHz | Crucial\_CT32G4RFD4266 | 32000 MiB | Ryzen\_3\_2200U | 0 MiB | SSD | 128000 MiB | Arch\_Linux | y |
| Ivanov\_Fedor | Intel\_i3\_11580 | 20 | 5000MHz | Crucial\_CT64G4RFD4266 | 128000MiB | Geforce\_2080Ti | 20000 MiB | SSD | 256000 MiB | Linux\_From\_Scratch | y |
| Keker | Celeron\_3060 | 1 | 700 MHz | No | 0 MiB | No | 0 MiB | HDD | 60000 MiB | Windows\_95 | n |
| Antoshka | Arduino\_Uno | 1 | 1400MHz | Samsung\_DDR2 | 128 MiB | No | 0 MiB | SSD | 512 MiB | Linux | n |
| Ivanov\_Fedor | Intel\_i3\_11580 | 20 | 5000MHz | Crucial\_CT64G4RFD4266 | 128000MiB | No | 0 MiB | SSD | 256000 MiB | Linux\_From\_Scratch | n |
| Gorokhov\_Mikhail | AMD\_Ryzen\_5\_2600X | 12 | 3200MHz | Crucial\_CT32G4RFD4266 | 16000 MiB | Radeon\_470 | 4096 MiB | SSD | 256000 MiB | Linux\_From\_Scratch | y |

Пример работы программы ***./output.c -p***:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Owner\_Name | CPU\_Model | CPU\_Cores | CPU\_Clocks | RAM\_Model | RAM\_Value | GPU\_Model | GPU\_Value | Storage\_Model | Storage\_Value | OS | Is\_Completed? |
| Keker | Celeron\_3060 | 1 | 700 MHz | No | 0 MiB | No | 0 MiB | HDD | 60000 MiB | Windows\_95 | n |
| Antoshka | Arduino\_Uno | 1 | 1400MHz | Samsung\_DDR2 | 128 MiB | No | 0 MiB | SSD | 512 MiB | Linux | n |
| Ivanov\_Fedor | Intel\_i3\_11580 | 20 | 5000MHz | Crucial\_CT64G4RFD4266 | 128000MiB | No | 0 MiB | SSD | 256000 MiB | Linux\_From\_Scratch | n |

# ПРОГРАММНЫЙ КОД И ТЕСТОВЫЕ ДАННЫЕ

Код программы ***input.c***:

#include <stdio.h>

#include <stdlib.h>

#include "studentpc.h"

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

if(argc!=3) {

printf("Введите ещё два аргумента в формате:\n./input <Входной текстовый файл> <Выходной бинарный файл>\n");

return 2;

}

Student\_PC pc;

FILE \*output;

FILE \*input;

input = fopen(argv[1],"r");

output = fopen(argv[2],"wb");

while(fscanf(input,"%s %s %d %d %s %d %s %d %s %d %s %c\n",

pc.owner,

pc.cpuModel, &pc.cpuCores, &pc.cpuClocks,

pc.ramModel, &pc.ramValue,

pc.gpuModel, &pc.gpuValue,

pc.storageModel, &pc.storageValue,

pc.os, &pc.isCompleted

)!=EOF) {

fwrite(&pc,sizeof(pc),1,output);

}

fclose(input);

fclose(output);

return 0;

}

Код программы ***ouput.c***:

#include <stdio.h>

#include <stdlib.h>

#include "studentpc.h"

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

if(argc != 3) {

printf("Inputed wrong parameters. Must be: programm <bin file> -p/-f \n");

return 2;

}

FILE\* input;

input = fopen(argv[1], "rb");

Student\_PC pc;

if(strcmp(argv[2], "-f") == 0) {

printf("--------------------------\n");

printf("Owner\_Name CPU\_Model CPU\_Cores CPU\_Clocks RAM\_Model RAM\_Value GPU\_Model GPU\_Value Storage\_Model Storage\_Value OS Is\_Completed?\n");

while(fread(&pc, sizeof(pc), 1, input)) {

printf("%-25s %-22s %-2d %-4dMHz %-22s %-6dMiB %-22s %-8dMiB %-22s %-8dMiB %-22s %-1c\n",

pc.owner,

pc.cpuModel, pc.cpuCores, pc.cpuClocks,

pc.ramModel, pc.ramValue,

pc.gpuModel, pc.gpuValue,

pc.storageModel, pc.storageValue,

pc.os, pc.isCompleted

);

}

}

else if(strcmp(argv[2], "-p") == 0) {

printf("--------------------------\n");

printf("Owner\_Name CPU\_Model CPU\_Cores CPU\_Clocks RAM\_Model RAM\_Value GPU\_Model GPU\_Value Storage\_Model Storage\_Value OS Is\_Completed?\n");

while(fread(&pc, sizeof(pc), 1, input)) {

if(pc.isCompleted == 'y') { continue;}

printf("%-25s %-22s %-2d %-4dMHz %-22s %-6dMiB %-22s %-8dMiB %-22s %-8dMiB %-22s %-1c\n",

pc.owner,

pc.cpuModel, pc.cpuCores, pc.cpuClocks,

pc.ramModel, pc.ramValue,

pc.gpuModel, pc.gpuValue,

pc.storageModel, pc.storageValue,

pc.os, pc.isCompleted

);

}

}

else {

printf("Wrong parameters\n");

return 1;

}

return 0;

}

Код программы ***studentpc.h***:

#ifndef \_\_studentpc.h\_\_

#define \_\_studentpc.h\_\_

#include <stdio.h>

#define LENGTHOWNER 25

#define LENGTHMODEL 22

typedef struct {

char owner[LENGTHOWNER];

char cpuModel[LENGTHMODEL];

unsigned int cpuCores;

unsigned int cpuClocks;

char ramModel[LENGTHMODEL];

unsigned int ramValue;

char gpuModel[LENGTHMODEL];

unsigned int gpuValue;

char storageModel[LENGTHMODEL];

unsigned int storageValue;

char os[LENGTHMODEL];

char isCompleted;

} Student\_PC;

#endif

Данные текстового файла ***data1.txt***:

Gorokhov\_Mikhail AMD\_Ryzen\_3\_2200U 4 2500 Crucial\_CT32G4RFD4266 32000 Ryzen\_3\_2200U 0 SSD 128000 Arch\_Linux y

Ivanov\_Fedor Intel\_i3\_11580 20 5000 Crucial\_CT64G4RFD4266 128000 Geforce\_2080Ti 20000 SSD 256000 Linux\_From\_Scratch y

Keker Celeron\_3060 1 700 No 0 No 0 HDD 60000 Windows\_95 n

Antoshka Arduino\_Uno 1 1400 Samsung\_DDR2 128 No 0 SSD 512 Linux n

Ivanov\_Fedor Intel\_i3\_11580 20 5000 Crucial\_CT64G4RFD4266 128000 No 0 SSD 256000 Linux\_From\_Scratch n

Gorokhov\_Mikhail AMD\_Ryzen\_5\_2600X 12 3200 Crucial\_CT32G4RFD4266 16000 Radeon\_470 4096 SSD 256000 Linux\_From\_Scratch y

# ЗАКЛЮЧЕНИЕ

Я составил последовательную структуру данных для представления простейшей базы данных на файлах в СП Си на примере сведений о составе комплектующих личных ПЭВМ в студентческой группе. Реализовал программу, сохраняющую данные в бинарный файл, и программу, выводящую данные из бинарного файла в приемлимый для пользователя вид, то есть в виде таблицы.

Программа умеет выводить аннотированный список неукомплектованных компьютеров.

Я освоил работу со структурами в Си, работу с фалйами, с бинарными файлами и основы релаизации Системы базы данных.

# ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Керниган Б., Ритчи Д. Язык программирования Си = The C programming language. — 2-е изд. — М.: Вильямс, 2007. — С. 304. — ISBN 0-13-110362-8.
2. Эндрю Таненбаум*, Structured Computer Organization,* [ISBN 0-13-148521-0](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:%D0%98%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8_%D0%BA%D0%BD%D0%B8%D0%B3/0131485210)
3. В.Е. Зайцев, Конспект Лекций по курсу “Фундаментальная Информатика и Языки и методы программирования”.