

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
1. ЦЕЛИ ПРОЕКТА.....	4
2. ПОСТАНОВКА ЗАДАЧИ.....	5
3. СТРУКТУРА ПРОЕКТА.....	6
4. ПРОГРАММНЫЙ КОД И ТЕСТОВЫЕ ДАННЫЕ.....	8
5. ЗАКЛЮЧЕНИЕ.....	11
6. ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ.....	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.

1. 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.

2. 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 студенческой 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 неукomплектованных компьютеров

3. СТРУКТУРА ПРОЕКТА

Проект состоит из 5 файлов: программы, сохраняющей данные в бинарный файл *input.c*, программы, выводящей содержимое бинарного файла в приемлемый 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 неукomплектованных 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	2500M Hz	Crucial_CT32G4RFD4266	32000 MiB	Ryzen_3_2200U	0 MiB	SSD	128000 MiB	Arch_Linux	y
Ivanov_Fedor	Intel_i3_11580	20	5000M Hz	Crucial_CT64G4RFD4266	12800 MiB	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	1400M Hz	Samsung_DDR2	128 MiB	No	0 MiB	SSD	512 MiB	Linux	n
Ivanov_Fedor	Intel_i3_11580	20	5000M Hz	Crucial_CT64G4RFD4266	12800 MiB	No	0 MiB	SSD	256000 MiB	Linux_From_Scratch	n
Gorokhov_Mikhail	AMD_Ryzen_5_2600X	12	3200M Hz	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	1400M Hz	Samsung_DDR2	128 MiB	No	0 MiB	SSD	512 MiB	Linux	n
Ivanov_Fedor	Intel_i3_11580	20	5000M Hz	Crucial_CT64G4RFD4266	12800 MiB	No	0 MiB	SSD	256000 MiB	Linux_From_Scratch	n

4. 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\n");
    printf("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\n");
    printf("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

```

5. 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 студенческой 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 неукomплектованных компьютеров.

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

6. 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](#)
3. В.Е. Зайцев, Конспект Лекций по курсу “Фундаментальная Информатика и Языки и методы программирования”.