**Министерство образования науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский авиационный институт (национальный исследовательский университет)»

«Информационные технологии и прикладная математика»

**Курсовой проект**

**По курсу «Практикум на ЭВМ»**

**2 семестр**

**Задание 5:**

«Обработка последовательной файловой структуры»

|  |  |
| --- | --- |
| Группа: | М8О-106Б-21 |
| Студент: | Орусский В.Р. |
| Преподаватель: | Дубинин А.В. |
| Оценка: |  |
| Дата: |  |

**Оглавление**

[Введение 3](#_Toc104734215)

[Постановка задачи 4](#_Toc104734216)

[Вариант 1: 4](#_Toc104734217)

[Теоретическая часть 4](#_Toc104734218)

[Описание алгоритма 5](#_Toc104734219)

[Использованные в программе переменные 5](#_Toc104734220)

[Пользовательские структуры данных в программе: 5](#_Toc104734221)

[Исходники программы 7](#_Toc104734222)

[Протокол исполнения и тесты 10](#_Toc104734223)

[Вывод 11](#_Toc104734224)

# Введение

Изучить работу с файлами с помощью языка программирования Си, обработку аргументов, переданных из командной строки. Сгенерировать файл заданной структуры.

# Постановка задачи

Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей (15- 20). Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие для 2–3 значений параметров запроса p и распечатать результат.

## Вариант 1:

Найти всех владельцев двухпроцессорных ПК, имеющих не более внешних устройств.

# Теоретическая часть

Для выполнения курсовой работы будем использовать основные функции для работы с файлами в СП Си. Для начала работы с файлом необходимо его открыть с помощью команды «fopen», возвращающей указатель на файл. В функцию передаются путь до файла (в дальнейшем «имя файла») и режим, в котором будет открыт файл:

«r» - режим открытия файла для чтения;

«w» - режим для записи в файл. Если файл с таким именем уже существует, то его содержимое стирается и файл рассматривается как пустой;

«a» - режим дополнения существующего файла. Если файл не существует, то создаётся пустой.

Для записи (чтения) массива данных в файл будем использовать функцию «fwrite» («fread»), которая принимает указатель на массив данных, который нужно записать в файл, размер в байтах каждого элемента массива, количество элементов массива и указатель на файл.

После завершения работы с файлом его необходимо закрыть с помощью функции «fclose», которой передаётся имя файла.

# Описание алгоритма

**«сreate» – создание БД**

Для создания таблицы необходимо проверить, существует ли файл с указанным именем. Для этого откроем файл на чтение, если указатель на файл не нулевой, значит файл существует и необходимо завершить программу, иначе открываем файл с ключом «a» (создадим пустой файл с указанным именем).

**«add» – добавление элемента в БД**

Для добавления записи в файл проверим, существует ли он. Если существует, считываем некоторое число записей нашей БД и с помощью функции fwrite записываем в файл по одной записи.

**«removes» – удаление БД**

Для удаления таблицы достаточно воспользоваться функцией «remove», которая стирает файл с указанным именем

**«print» – печать БД**

Для вывода таблицы воспользуемся функцией «fread», которая будем читать из файла записи, а используя форматированный вывод, будем печатать элементы структуры в виде таблицы

**«CountUsers» - анализ БД на кол-во нужных пользователей**

Данная команда проходит по всем записям БД и проверяет в них кол-во периферийных девайсов, если меньше нужно размера, то счётчик «count» увеличивается.

# Использованные в программе переменные

|  |  |  |
| --- | --- | --- |
| Название | Тип данных | За что отвечает |
| fileName | char[] | Имя файла, с которым работаем |
| file | FILE\* | Файл с базой данных |
| records | integer | Кол-во записей |
| count | integer | Подсчёт кол-ва пользователей, удовлетворяющих условию задания |
| p | integer | Кол-во пользователей |

# Пользовательские структуры данных в программе:

|  |  |
| --- | --- |
| Название | Поля |
| PC | Характеристики ПК |

# Исходники программы

Файл «DataBase.h» – заголовочный файл, в котором объявлены все структуры и функции.

#ifndef **database\_h**#define **database\_h**#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <limits.h>  
  
*typedef struct* {  
 *char* surname[50];  
 *int* numberOfProcessors;  
 *char* typeOfProcessors[4]; *// 1 - x32, 2 - x64  
 int* memoryCapacity;  
 *char* typeOfVideoController[9]; *// 1 - built-in, 2 - external, 3 - AGP, 4 - PCI  
 int* videoMemoryCapacity;  
 *char* videoMemoryType[9]; *// SCSI/IDE(1) or ATA/SATA(2)  
 int* numberOfHardDrives;  
 *int* capacityOfHardDrives;  
 *int* peripherals;  
 *char* OC[8]; *// 1 - windows, 2 - linux, 3 - macOS*} PC;  
  
*void* create();  
*void* add();  
*void* print();  
*void* removes();  
*void* CountUsers();  
  
#endif

Файл «DataBase.c» - реализация указанных выше функций.

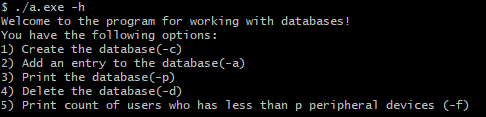
#include <stdio.h>  
#include <string.h>  
#include "DataBase.h"  
  
*// Создаём БД  
void* create() {  
 *char* fileName[40];  
 FILE \*file;  
 printf("Enter the name of the file you want to create:\n");  
 scanf("%s", fileName);  
 file = fopen(fileName, "rb");  
 *if* (file != **NULL**) {  
 printf("Such file already exists\n");  
 } *else* {  
 file = fopen(fileName, "ab");  
 }  
 printf("Creation was successful!\n");  
 fclose(file);  
}  
  
*// Добавляем записи в БД  
void* add() {  
 *int* records;  
 *char* fileName[40];  
 FILE \*file;  
 PC add;  
 printf("Enter the name of the file you want to add the records:\n");  
 scanf("%s", fileName);  
 file = fopen(fileName, "rb");  
 *if* (file == **NULL**) {  
 printf("Such file not exists!\n");  
 } *else* {  
 file = fopen(fileName, "ab");  
 printf("Enter how many records do you want to add:\n");  
 scanf("%d", &records);  
 *for* (*int* i = 0; i < records; i++) {  
 printf("Enter student's surname:\n");  
 scanf("%s", add.surname);  
 printf("Enter student's PC number of processors:\n");  
 scanf("%d", &add.numberOfProcessors);  
 printf("Enter student's PC type of processors:\n");  
 printf("1. x32\n2. x64\n");  
 scanf("%s", add.typeOfProcessors);  
 printf("Enter student's PC RAM capacity(GB):\n");  
 scanf("%d", &add.memoryCapacity);  
 printf("Enter student's PC type of video controller:\n");  
 printf("1. built-in\n2. external\n3. AGP\n4. PCI\n");  
 scanf("%s", add.typeOfVideoController);  
 printf("Enter student's PC video memory capacity(GB):\n");  
 scanf("%d", &add.videoMemoryCapacity);  
 printf("Enter student's PC video memory type(GB):\n");  
 printf("1. SCSI/IDE\n2. ATA/SATA\n");  
 scanf("%s", add.videoMemoryType);  
 printf("Enter student's PC number of hard drives:\n");  
 scanf("%d", &add.numberOfHardDrives);  
 printf("Enter student's PC capacity of hard drives(GB):\n");  
 scanf("%d", &add.capacityOfHardDrives);  
 printf("Enter student's PC number of peripherals:\n");  
 scanf("%d", &add.peripherals);  
 printf("Enter student's PC OC:\n");  
 printf("1) Windows\n2) Linux\n3) MacOS\n");  
 scanf("%s", add.OC);  
 fwrite(&add, *sizeof*(PC), 1, file);  
 }  
 printf("Records added successfully!\n");  
 fclose(file);  
 }  
}  
  
*// Выводим БД  
void* print() {  
 *char* fileName[40];  
 printf("Enter the name of the file you want to print:\n");  
 scanf("%s", fileName);  
 FILE \*file;  
 PC read;  
 file = fopen(fileName, "rb");  
 *if* (file == **NULL**) {  
 printf("Such file not exists\n");  
 } *else* {  
 printf(  
 "|Student's surname|Num Proc|Type of proc| RAM |Type video control|Video memory|Video memory type|Num HD|Capacity HD|Num peripherals|OC|\n");  
 printf(  
 "-------------------------------------------------------------------------------------------------------------------------------\n");  
 *while* (fread(&read, *sizeof*(PC), 1, file) != **EOF** && !feof(file)) {  
 printf("|%17s|%8d|%12s|%3dGB|%18s|%10dGB|%17s|%4dGB|%9dGB|%15d|%2s|\n",  
 read.surname,  
 read.numberOfProcessors,  
 read.typeOfProcessors,  
 read.memoryCapacity,  
 read.typeOfVideoController,  
 read.videoMemoryCapacity,  
 read.videoMemoryType,  
 read.numberOfHardDrives,  
 read.capacityOfHardDrives,  
 read.peripherals,  
 read.OC);  
 }  
 fclose(file);  
 }  
}  
  
*// Удаление БД (файла)  
void* removes() {  
 *char* fileName[40];  
 printf("Enter the name of the file you want to delete:\n");  
 scanf("%s", fileName);  
 FILE \*file;  
 file = fopen(fileName, "rb");  
 *if* (!file) {  
 printf("Such file not exists!\n");  
 } *else* {  
 remove(fileName);  
 printf("The file was successfully deleted!\n");  
 fclose(file);  
 }  
}  
  
*// Подсчёт пользователей, имеющих  
void* CountUsers() {  
 *char* fileName[40];  
 *int* p;  
 printf("Enter the name of the file you want to know how many users have no more than \"p\" peripheral devices:\n");  
 scanf("%s", fileName);  
 printf("Enter 'p':\n");  
 scanf("%d", &p);  
 FILE \*file;  
 PC f;  
 *int* count = 0;  
 file = fopen(fileName, "rb");  
 *if* (!file) {  
 printf("Such file not exists!\n");  
 } *else* {  
 *while* (fread(&f, *sizeof*(PC), 1, file) != **EOF** && !feof(file)) {  
 *if* (f.peripherals <= p) {  
 count++;  
 }  
 }  
 printf("%d", count);  
 fclose(file);  
 }  
}

Файл «main.c» - файл для взаимодействия с пользователем и обработки аргументов командной строки.

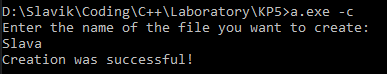
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include "DataBase.h"  
  
*void* print\_help() {  
 printf("Welcome to the program for working with databases!\nYou have the following options:\n");  
 printf("1) Create the database(-c)\n2) Add an entry to the database(-a)\n3) Print the database(-p)\n");  
 printf("4) Delete the database(-d)\n5) Print count of users who has less than p peripheral devices (-f)\n");  
}  
  
*int* main(*int* argc, *char* \*argv[]) {  
 *if* (argc != 2) {  
 printf("Need exactly two arguments\n");  
 exit(1);  
 }  
 *if* (!strcmp(argv[1], "-a")) {  
 add();  
 } *else if* (!strcmp(argv[1], "-p")) {  
 print();  
 } *else if* (!strcmp(argv[1], "-d")) {  
 removes();  
 } *else if* (!strcmp(argv[1], "-h")) {  
 print\_help();  
 } *else if* (!strcmp(argv[1], "-c")) {  
 create();  
 } *else if* (!strcmp(argv[1], "-f")) {  
 CountUsers();  
 } *else* {  
 printf("Unknown command\n");  
 }  
 *return* 0;  
  
}

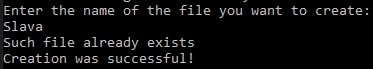
# Протокол исполнения и тесты

Тест №1

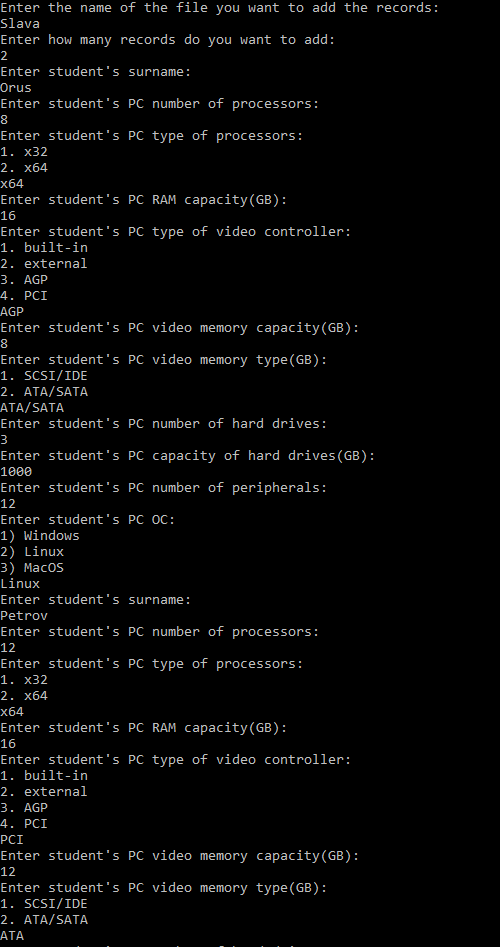
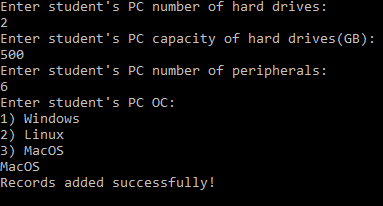


Тест №2

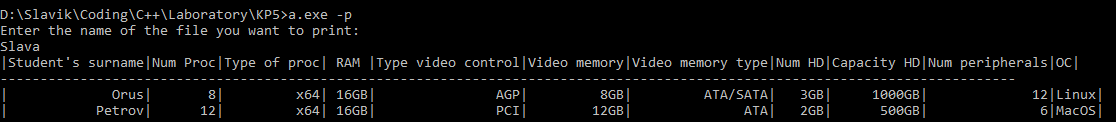




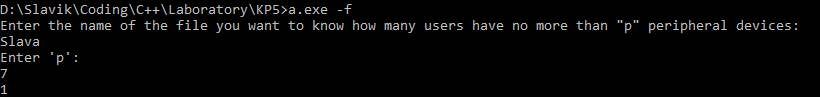
Тест №3



Тест №4



Тест №5



# Вывод

Для выполнения данной работы был изучен процесс работы с файлами с помощью СП Си, что является очень важным навыков в дальнейшем.

В процессе написания работы была изучена обработка и функции для работы с параметрами командной строки, что в свою очередь поможет в дальнейшем разрабатывать ПО для ОС.