

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра прикладной математики и кибернетики

Отчет по лабораторной работе
по дисциплине «Операционные системы»

Лабораторная работа №1

Выполнил: студент 3 курса
группы ИП-811
Мироненко К. А.

Проверил: профессор
кафедры ПМиК
Малков Е. А.

ОГЛАВЛЕНИЕ

1. Постановка задачи.....	3
2. Примеры работы программы	4
<i>Приложение</i> Листинг.....	7

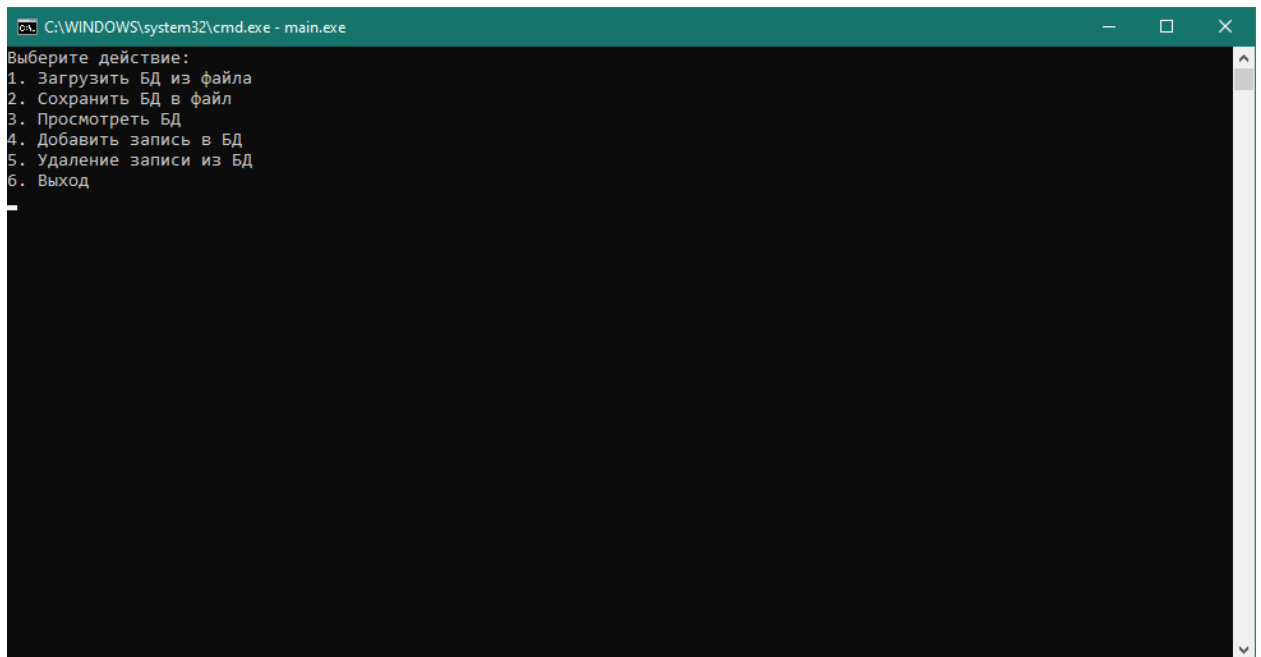
1. Постановка задачи

Цель: проверка знаний, необходимых для усвоения курса Операционные системы.

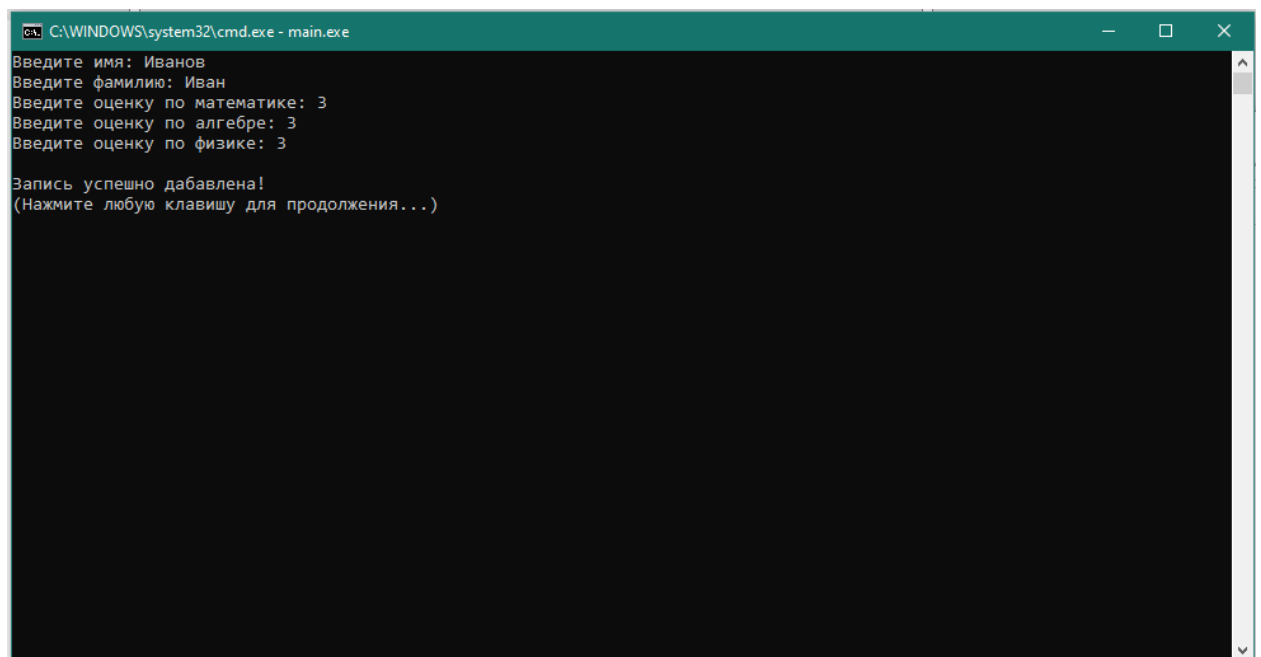
Задание: написать программу для манипуляции данными на основе рекурсивных структур, реализовать функции вставки, удаления и навигации для списка; реализовать сериализацию списка. (Например: электронный журнал успеваемости студентов.)

Инструментарий: любая среда разработки C/C++..

2. Примеры работы программы



(Меню программы)



(Добавление записи)

```
C:\WINDOWS\system32\cmd.exe - main.exe
1. Иванов Иван 3 3 3
2. Петров Петр 3 4 5
3. Козлова Анна 6 6 6

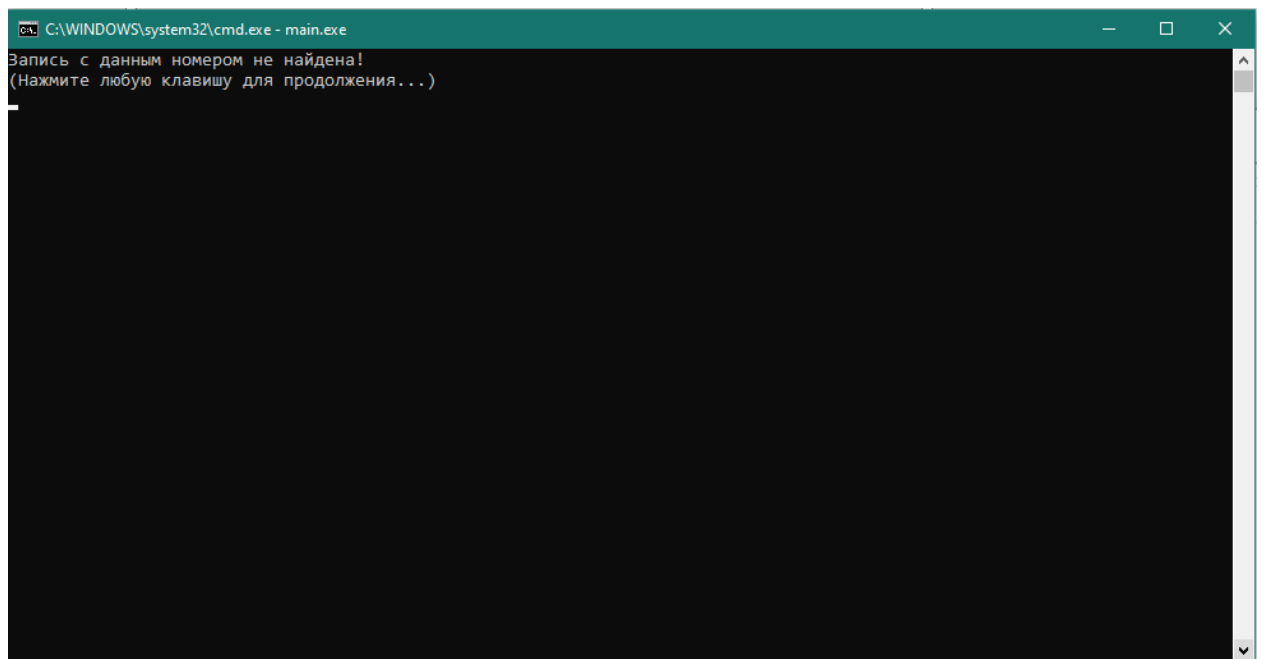
(Нажмите любую клавишу для продолжения...)
```

(Вывод базы)

```
C:\WINDOWS\system32\cmd.exe - main.exe
1. Петров Петр 3 4 5
2. Козлова Анна 6 6 6

Введите номер записи, которую вы хотите удалить: 1
```

(Удаление записи)



(Удаление, при вводе некорректного значения)

Приложение. Листинг

main.c

```
#include <windows.h>
#include <stdio.h>
#include <locale.h>

int main(int argc, char* argv[])
{
    setlocale(LC_ALL, "");

    char systemPath[MAX_PATH], tempPath[MAX_PATH];
    SYSTEM_INFO systemInformation;
    MEMORYSTATUSEX memory;
    memory.dwLength = sizeof(memory);

    GetWindowsDirectory(systemPath, MAX_PATH);
    GetTempPath(MAX_PATH, tempPath);
    GetSystemInfo(&systemInformation);
    GlobalMemoryStatusEx(&memory);

    printf("\n~~~~~\n");
    printf("\n Системный каталог Windows:    %s", systemPath);
    printf("\n Каталог временных файлов Windows: %s", tempPath);
    printf("\n~~~~~\n");
    printf("\n Количество ядер процессора: %d", systemInformation.dwNumberOfProcessors);
    printf("\n Архитектура процессора:    ");
    switch (systemInformation.wProcessorArchitecture)
    {
        case 0:
            printf("x86");
            break;
        case 9:
            printf("x64");
            break;
        default:
            printf("неизвестно");
            break;
    }
    printf("\n\n Нижняя граница пользовательского адресного пространства: %p",
systemInformation.lpMinimumApplicationAddress);
    printf("\n Верхняя граница пользовательского адресного пространства: %p",
systemInformation.lpMaximumApplicationAddress);
    printf("\n~~~~~\n");
    printf("\n Физическая память:");
    printf("\n Всего:    %llu Мб", memory.ullTotalPhys / 1048576);
    printf("\n Доступно:  %llu Мб", memory.ullAvailPhys / 1048576);
    printf("\n Используется: %llu Мб", (memory.ullTotalPhys - memory.ullAvailPhys) /
1048576);
    printf("\n\n Виртуальная память:");
    printf("\n Всего:    %llu Мб", memory.ullTotalVirtual / 1048576);
    printf("\n Доступно:  %llu Мб", memory.ullAvailVirtual / 1048576);
```

```
        printf("\n  Используется: %llu Мб", (memory.ullTotalVirtual - memory.ullAvailVirtual) /
1048576);
        printf("\n\n~~~~~\n\n~~~~~\n\n");

        return 0;
}
```