**Отчет по лабораторной работе № 6**

**по практикуму по циклу дисциплин "Информатика"**

Выполнил студент группы М8О-114БВ-24: Дробышев Егор Павлович, № по списку 9

Контакты e-mail: [tru.899@yandex.ru](mailto:tru.899@yandex.ru)

Работа выполнена: «1» марта 2025 г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Отчет сдан « » 2025г., итоговая оценка

Подпись преподавателя

1. **Тема:** «Обработка последовательной файловой структуры на языке Си»
2. **Цель работы:** Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом.
3. **Задание**: Найти абитуриентов-медалистов, не набравших проходной балл p (Вариант – 22)

# Оборудование: оборудование ПЭВМ студента, если использовалось:

Процессор \_Intel Core i5\_ с ОП 8 Гб НМД 256 Гб. Монитор 1920x1080~60Hz. Другие устройства не использовались

# Программное обеспечение: программное обеспечение ЭВМ студента:

Операционная система семейства \_Linux\_, наименование \_Ubuntu\_ версия \_24.04

интерпретатор команд \_GNU bash\_ версия \_5.2.21(1)\_.

Редактор текстов emacs версия 29.3

Утилиты операционной системы: gcc, gdb

Местонахождение и имена файлов программ и данных на домашнем компьютере:

/home/tru\_\_

1. **Идея, метод, алгоритм решения задачи** (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Необходимо обработать простую базу данных из заданного текстового файла. Реализация проекта:

1. Создание заголовочного файла students.h, содержащий структуры данных: фамилию – char surname[50]; инициалы – char init[3], пол – char sex; номер школы – int school\_num; наличие медали – char medal; зачет по сочинению – char zachet; балл по математике – int matan; балл по информатике – int infa; балл по русскому – int rus; суммарный балл – int summary.
2. Написание программы dump.c, которая переводит текстовый файл в бинарный. Открывает текстовый файл для чтения. Открывает бинарный файл для записи. Считывает данные студентов построчно (fscanf). Записывает структуру Student в бинарный файл (fwrite). Закрывает файлы.
3. Написание программы main.c, которая считывает данные из бинарного файла, выводит все данные файла в виде таблицы (параметр –f), а также проводит выборку абитуриентов-медалистов, не набравших проходной балл p (параметр -p), задающийся пользователем.
4. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

student.h – заголовочный файл со структурой

#ifndef \_\_student\_h\_\_

#define \_\_student\_h\_\_

#include <stdio.h>

#include <stdlib.h>

typedef struct {

    char surname[50];

    char init[3];

    char sex;

    int school\_num;

    char medal;

    char zachet;

    int matan;

    int infa;

    int rus;

    int summary;

} Student;

#endif

dump.c – программа перевода txt в bin

#include <stdio.h>

#include <stdlib.h>

#include "student.h"

void usage() {

    printf("Usage: program\_name filename.txt filename.bin\n");

}

int read\_student(Student \**s*, FILE \**in*) {

    int result = fscanf(*in*, "%s %s %c %d %c %c %d %d %d %d",

*s*->surname, *s*->init, &*s*->sex,

                        &*s*->school\_num, &*s*->medal, &*s*->zachet,

                        &*s*->matan, &*s*->infa, &*s*->rus, &*s*->summary);

    return result == 10;

}

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

    if (*argc* != 3) {

        usage();

        return 1;

    }

    FILE \*in = fopen(*argv*[1], "r");

    if (!in) {

        perror("Error opening input file");

        return 2;

    }

    Student s;

    FILE \*out = fopen(*argv*[2], "w");

    if (!out) {

        perror("Error\n");

        fclose(in);

        return 3;

    }

    while (read\_student(&s, in)) {

        fwrite(&s, sizeof(s), 1, out);

    }

    fclose(in);

    fclose(out);

    return 0;

}

main.c – программа, считывает данные из бинарного файла

#include "student.h"

#include<string.h>

void usage() {

    printf("Usage: program\_name filename [-f | -p <user\_parameter>]\n");

}

void table\_header() {

    printf("%-15s|%-4s| %-4s|%-6s|%-5s|%-5s|%-5s|%-3s|%-3s |%-4s|\n",

           "Surname", "Init", "Sex", "School", "Medal", "Zachet",

           "Matan", "Infa", "Rus", "Summary");

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

}

void student\_print(Student \**s*) {

    printf("%-15s| %-3s|  %-3c| %-5d|  %-3c|   %-3c|  %-3d| %-3d| %-3d|  %-5d|\n",

*s*->surname, *s*->init, *s*->sex, *s*->school\_num,

*s*->medal, *s*->zachet, *s*->matan, *s*->infa,

*s*->rus, *s*->summary);

}

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

    if (*argc* < 3) {

        usage();

        return 1;

    }

    Student s;

    FILE \*in = fopen(*argv*[1], "r");

    if (!in) {

        perror("Error opening input file");

        return 2;

    }

    FILE \*out = fopen("temp.bin", "w");

    if (!out) {

        perror("Error opening output file");

        fclose(in);

        return 3;

    }

    int p\_value = 0;

    int flag\_found = 0;

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

        table\_header();

        while (fread(&s, sizeof(s), 1, in) == 1) {

            student\_print(&s);

            fwrite(&s, sizeof(s), 1, out);

        }

    }

    else if (*argc* == 4 && strcmp(*argv*[2], "-p") == 0) {

        p\_value = atoi(*argv*[3]);

        table\_header();

        while (fread(&s, sizeof(s), 1, in) == 1) {

            if (s.sex == 'M' && s.medal == 'Y' && (s.matan + s.infa + s.rus) < p\_value) {

                student\_print(&s);

                flag\_found = 1;

            }

        }

        if (!flag\_found) {

            printf("No students with medal and summary < %d\n", p\_value);

            }

    } else {

        usage();

    }

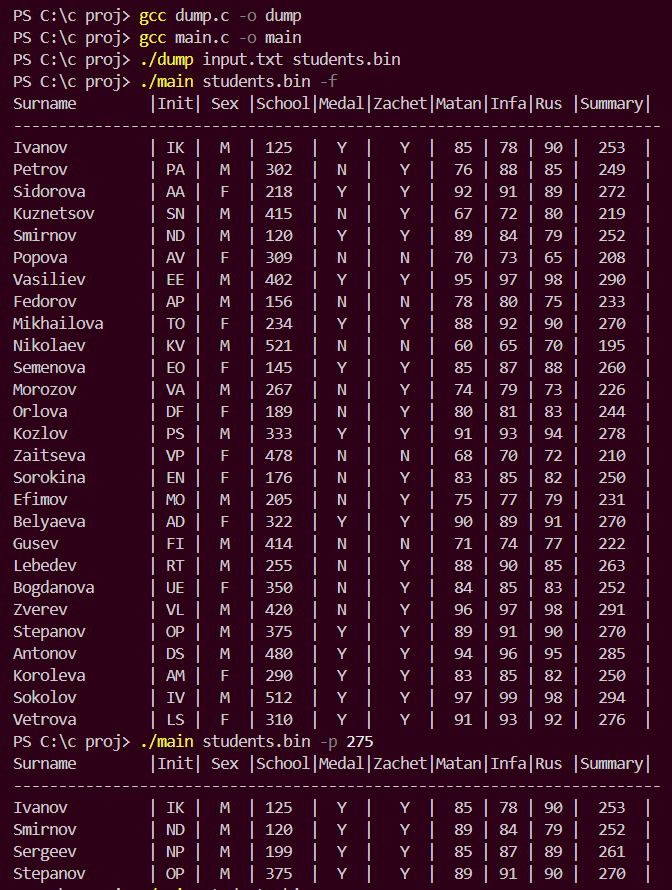
    fclose(in);

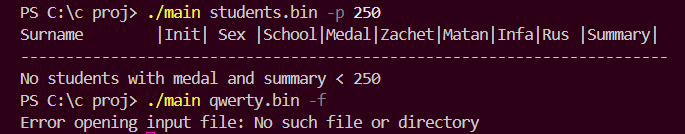
    fclose(out);

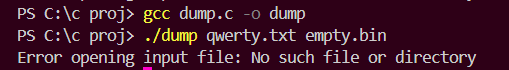
    return 0;

}

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).







1. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. или дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Замечания автора по существу работы

1. **Выводы:**

В ходе выполнения лабораторной работы я научился разрабатывать последовательную структуру данных для хранения и обработки простой базы данных в файлах на языке Си. Недочёты при выполнении задания могут быть устранены следующим образом:

**Подпись студента** \_\_\_\_\_\_\_\_\_\_\_\_