МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Факультет № 8

«Информационные технологии и прикладная математика»

Кафедра 806

**КУРСОВОЙ ПРОЕКТ**

**по курсу «Практикум на ЭВМ»**

на тему

“Обработка последовательной файловой структуры на языке Си”

Выполнил:

Студент 1 курса, гр. М8О-106Б-19

Кондратьев Егор Алексеевич

Москва

2020 г.

# **Содержание**

[**Содержание** 2](#_Toc43848807)

[**Задание** 2](#_Toc43848808)

[**Вариант 13** 3](#_Toc43848809)

[**Общий метод решения** 4](#_Toc43848810)

[**Используемые переменные** 5](#_Toc43848811)

[**Входные значения** 6](#_Toc43848812)

[**Код** 7](#_Toc43848813)

[**Заключение** 10](#_Toc43848814)

[**Список использованных источников** 10](#_Toc43848815)

# **Задание**

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).

Действие по выборке данных из файла оформить в виде отдельной программы с параметрами запроса, вводимыми из стандартного входного текстового файла, или получаемыми из командной строки UNIX. Второй способ задания параметров обязателен для работ, оцениваемых на хорошо и отлично. Параметры задаются с помощью ключей —f (распечатка файла) или -р <parameter> (параметры конкретного варианта задания). Получение параметров из командной строки производится с помощью стандартных библиотечных функций argc и argv ,

В процессе отладки и тестирования рекомендуется использовать команды обработки текстовых файлов ОС UNIX и переадресацию ввода-вывода. Сгенерированные и отформатированные тестовые данные необходимо заранее поместить в текстовые файлы и распечатывать при протоколировании. Рекомендуется подобрать реальные или правдоподобные тестовые данные. Имя файла с бинарными данными является обязательным параметром второй программы.

Отчет должен содержать оценку пространственной и временной сложности использованного алгоритма, В состав отчета также рекомендуется включить графическую иллюстрацию структуры файла и запроса на выборку.

# **Вариант 13**

|  |  |
| --- | --- |
| Содержимое и структура файла | Информация об успеваемости студентов данной группы по всем предметам: фамилия, инициалы, пол, номер группы, отметки по экзаменам и зачетам. |
| Действия | Выяснить, сколько студенток группы р имеют ровно одну пятерку |

# **Общий метод решения**

Задание: составить две программы, одна из которых читает файл в текстовом

представлении и записывает в новый файл в бинарном представлении, а вторая программа должна выполнить действие на полученном бинарном файле и вывести результат.

Сложность по времени O(n), по памяти O(k) для обеих программ, где n - число

строк во входном файле, k - размер буфера.

# **Используемые переменные**

|  |  |  |
| --- | --- | --- |
| **Переменные** | **Тип** | **Значение** |
| p | int | Номер группы |
| argc | Количество параметров при запуске программы |
| group | Номер группы студентов |
| count | Количество студентов, получивших одну пятерку |
| sname | char | Фамилия студента |
| name | Инициалы студента |
| ex1 | Оценка за первый экзамен студента |
| ex2 | Оценка за второй экзамен студента |
| zachot | Оценка за зачет студента |
| argv[] | Параметр при argc |

# **Входные значения**

Corbin B.R. m 106 5 4 -

Boone S.Q. w 107 5 4 -

Henrik R.E. w 107 5 4 -

Graham J.E. w 106 4 4 -

Weston A.G. m 106 4 4 -

Isidro T.n. m 106 4 4 -

Saul B.H. w 106 5 4 -

Grey B.L. w 106 5 4 -

Osman S.B. m 106 4 4 -

Braydon G.C. w 106 4 4 -

Raphael G.V. w 106 4 4 -

Rhett H.N. w 106 3 4 -

Porter W.T. w 106 4 5 +

Luciano G.U. w 106 2 4 -

Oakley P.Y. w 106 2 5 –

# **Код**

k1.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct

{

    char sname[80];

    char name[5];

    char sex[2];

    int group;

    int ex1;

    int ex2;

    char zachot[2];

} student;

int main(int argc, char \*argv[])

{

    if (argc != 3)

    {

        fprintf(stderr, "Enter input and output files(text and binary files).\n");

        return 1;

    }

    FILE \*in = fopen(argv[1], "r");

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

    if (!in)

    {

        fprintf(stderr, "Can't read from input file.\n");

        return 1;

    }

    if (!out)

    {

        fprintf(stderr, "Can't write in output file.\n");

        return 1;

    }

    student s;

    while (fscanf(in, "%s%s%s%d%d%d%s\n", s.sname, s.name, s.sex, &s.group, &s.ex1, &s.ex2, s.zachot) == 7)

    {

        fwrite(s.sname, sizeof(char) \* 80, 1, out);

        fwrite(s.name, sizeof(char) \* 5, 1, out);

        fwrite(s.sex, sizeof(char) \* 2, 1, out);

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

        fwrite(&s.ex1, sizeof(s.ex1), 1, out);

        fwrite(&s.ex2, sizeof(s.ex2), 1, out);

        fwrite(s.zachot, sizeof(char) \* 2, 1, out);

    }

    fclose(in);

    fclose(out);

    return 0;

}

k2.c

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct

{

    char sname[80];

    char name[5];

    char sex[2];

    int group;

    int ex1;

    int ex2;

    char zachot[2];

} student;

int readbin(FILE \*in, student \*s)

{

    int read = 0;

    read += fread(s->sname, sizeof(char) \* 80, 1, in);

    read += fread(s->name, sizeof(char) \* 5, 1, in);

    read += fread(s->sex, sizeof(char) \* 2, 1, in);

    read += fread(&(s->group), sizeof(s->group), 1, in);

    read += fread(&(s->ex1), sizeof(s->ex1), 1, in);

    read += fread(&(s->ex2), sizeof(s->ex2), 1, in);

    read += fread(s->zachot, sizeof(char) \* 2, 1, in);

    return read;

}

int main(int argc, char \*argv[])

{

    if (argc != 3)

    {

        fprintf(stderr, "Enter binary file and group(3 digits)\n");

        return 1;

    }

    student students;

    int p = (argv[2][0] - '0') \* 100 +

            (argv[2][1] - '0') \* 10 +

            (argv[2][2] - '0');

    int count = 0;

    FILE \*in = fopen(argv[1], "rb");

    if (!in)

    {

        fprintf(stderr, "Can't read from input file.\n");

        return 1;

    }

    while (readbin(in, &students) == 7)

    {

        if (strncmp(students.sex, "w", 1) == 0 && students.group == p && ((students.ex1 != 5 && students.ex2 == 5) || (students.ex1 == 5 && students.ex2 != 5)))

        {

            count++;

        }

    }

    printf("In this group %d girl(s), who has(have) one <<5>> mark\n", count);

    fclose(in);

    return 0;

}

# **Заключение**

В ходе работы я научился разрабатывать последовательную структуру данных для представления простейшей базы данных на файлах Си в соответствии с определенной задачей. Составил программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей (15-20). Распечатал содержимое сгенерированного файла в виде таблицы и выполнил над ним заданное действие для 2—3 значений параметров запроса р и распечатал результат.

# **Список использованных источников**

1.  Гайсаpян С. С., Зайцев В. Е . Г147 Кypс инфоpматики: Учеб. пособие. — М.: Изд-во Вузовская книга, 2012. — 424 с.

2. Параметры функции main (argc, argv) [Электронный ресурс]. - URL

<http://cppstudio.com/post/421/>

3. Функция strncmp [Электронный ресурс]. - URL

<http://cppstudio.com/post/715/>