

Лабораторная работа №10

Хранение данных в файлах

Цель работы: закрепить навыки работы со структурами и массивом структур, как простейшей линейной коллекцией составных данных. Изучить способы сохранения и повторного использования данных путем их записи в текстовые файлы.

Краткие теоретические сведения

Структура **FILE** содержит всю внутреннюю информацию о состоянии относительно соединения со связанным файлом, включая индикатор позиции файла и информацию буферизации. Прежде чем работать с файлом, его нужно открыть для доступа, т.е. создать и инициализировать область данных, которая содержит информацию о файле: имя, путь и т.д.

В языке Си это выполняет функция **fopen()**, которая связывает физический файл на носителе с логическим именем в программе. Логическое имя –это указатель на файл, т.е. на область памяти, где хранится информация о файле. Указатели на файлы необходимо объявлять:

```
FILE *указатель на файл;
```

Функция открытия файла:

```
FILE *fopen(char *fname, char *mode);
```

fname – указатель на имя файла в файловой системе в обычной форме: полный путь к файлу или сокращенное имя (если файл в текущем каталоге).

Mode – параметр, задающий тип операций, допустимых с файлом, записываемый следующими символьными константами (таблица 1).

Таблица 1 – Пояснения символьных констант параметра mode

Mode	Описание
r	Чтение. Файл должен существовать.
w	Запись нового файла. Если файл с таким именем уже существует, то его содержимое будет потеряно.
a	Запись в конец файла. Операции позиционирования (fseek, fsetpos, frewind) игнорируются. Файл создаётся, если не существовал.
r+	Чтение и обновление. Можно как читать, так и писать. Файл должен существовать.
w+	Запись и обновление. Создаётся новый файл. Если файл с таким именем уже существует, то его содержимое будет потеряно. Можно как писать, так и читать.
a+	Запись в конец и обновление. Операции позиционирования работают только для чтения, для записи игнорируются. Если файл не существовал, то будет создан новый.

Функция **fopen** возвращает указатель на поток, описываемой **FILE** и **NULL**, если при открытии файла произошла ошибка.

По умолчанию файл открывается в текстовом режиме (**t**), указание **b** в строке **mode** позволит открыть файл в двоичном режиме.

После работы доступ к файлу необходимо закрыть с помощью функции:

```
int fclose(указатель файла);
```

Работа с файлом обязательно включает следующие шаги:

1. Объявление указателя на используемый файл

```
FILE *f_my;
```

2. Открытие файла и обработка возможных ошибок

```
if(!(f_my = fopen("myfile.txt", "r+t"))){
```

```

        puts("\n Ошибка открытия файла!");
        return;
    }

```

3. Работа с файлом

4. Заккрытие файла

```

fclose(f_my);

```

Прототипы большинства функций по обработке файлов описаны в библиотеках `stdio.h` и `io.h`.

Для работы с текстовыми файлами в консольном приложении удобнее всего пользоваться функциями `fprintf()` и `fscanf()`, параметры и выполняемые действия аналогичны функциям `printf()` и `scanf()`, только первым параметром добавлен указатель файла, к которому применяется данная функция:

```

int fprintf (FILE *stream, const char *template, ...)
int fscanf (FILE *stream, const char *template, ...)

```

Примеры работы с текстовыми файлами

Пример 1. Создание файла и запись в него фразы “Hello, World!”

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    //С помощью переменной file будем осуществлять доступ к
    файлу
    FILE *file;
    //Открываем текстовый файл с правами на запись
    file = fopen("C:/Intel/test.txt", "w+t");
    //Пишем в файл
    fprintf(file, "Hello, World!");
    //Закрываем файл
    fclose(file);
    getchar();
    return 0;
}

```

```
}
```

Пример 2. Запись в файл фразы “Hello, World!” с последующим переоткрытием файла и чтения в буфер.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *file;
    char buffer[128];

    file = fopen("C:/Intel/test.txt", "w");
    fprintf(file, "Hello, World!");
    fclose(file);
    file = fopen("C:/Intel/test.txt", "r");
    if (file != NULL)
    {
        fgets(buffer, 127, file);
        printf("%s", buffer);
        fclose(file);
    }
    else
    {
        printf("Ошибка открытия файла!");
    }
    getchar();
    return 0;
}
```

Пример 3. Использование функции freopen

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE *file;
    char buffer[128];

    file = fopen("C:/Intel/test.txt", "w");
    fprintf(file, "Hello, World!");
    freopen("C:/Intel/test.txt", "r", file);
```

```

    fgets(buffer, 127, file);
    printf("%s", buffer);
    fclose(file);
    getchar();
    return 0;
}

```

Пример 4. Обработка ошибок открытия файла

```

#include <stdio.h>
#include <stdlib.h>
#define ERROR_OPEN_FILE -3
int main()
{
    FILE *file;
    char buffer[128];

    file = fopen("C:/Intel/test.txt", "w");
    if (file == NULL) {
        printf("Error opening file");
        getchar();
        exit(ERROR_OPEN_FILE);
    }
    fprintf(file, "Hello, World!");
    freopen("C:/Intel/test.txt", "r", file);
    if (file == NULL) {
        printf("Error opening file");
        getchar();
        exit(ERROR_OPEN_FILE);
    }
    fgets(buffer, 127, file);
    printf("%s", buffer);
    fclose(file);
    getchar();
    return 0;
}

```

Пример 5. Решение задания на лабораторную работу.

X	«Студент»: ФИО, группа, возраст Вывести сведения о самом молодом студенте.
---	--

```

#include <stdio.h>

```

```

#include <stdlib.h>
#include <string.h>
typedef struct Student
{
    char FIO[50];
    char Group[50];
    int age;
} Student;

void scan_str_from_file(FILE *input, char *buffer)
{
    fgets(buffer, 511, input);
    int length = strlen(buffer);
    for (int index = 0; index < length; index++) {
        if (buffer[index] == '\n') {
            buffer[index] = '\0';
            break;
        }
    }
}

int main()
{
    printf("Выберите действие:\n");
    printf("1.Записать сведения о студентах в файл\n");
    printf("2.Прочитать сведения о студентах из файла и
найти самого молодого\n");
    printf("Для выхода введите любую другую цифру\n");
    printf("Ваш ответ: ");
    int ans;
    scanf("%d", &ans);

    if(ans == 1)
    {
        int N;
        printf("Введите число студентов: ");
        scanf("%d", &N);
        getchar();

        Student pin[N];

        for(int i = 0; i < N; i++)
        {
            printf("Введите информацию о студенте №%d\n",
i + 1);

            printf("Введите фамилию студента: ");
            fgets(pin[i].FIO, 50, stdin);

```

```

        printf("Введите группу студента: ");
        fgets(pin[i].Group, 50, stdin);
        printf("Введите возраст студента: ");
        scanf("%d", &pin[i].age);
        getchar();
    }

    FILE *file = fopen("file.txt", "w");
    if(file == NULL)
    {
        printf("Ошибка открытия файла");
        getchar();
        return -3;
    }
    char N_str[4];
    sprintf(N_str, "%d\n", N);
    fprintf(file, N_str);
    for(int i = 0; i < N; i++)
    {
        fprintf(file, pin[i].FIO);
        fprintf(file, pin[i].Group);
        char age_str[4];
        sprintf(age_str, "%d\n", pin[i].age);
        fprintf(file, age_str);
    }
    fclose(file);
    printf("Файл успешно записан!");
}
else if(ans == 2)
{
    FILE *file = fopen("file.txt", "r");
    if(file == NULL)
    {
        printf("Ошибка открытия файла");
        getchar();
        return -3;
    }

    int N;
    fscanf(file, "%d\n", &N);
    printf("Число студентов в файле: %d\n", N);
    Student pin[N];

    for(int i = 0; i < N; i++)
    {
        char buffer[50];

```

```

        scan_str_from_file(file, pin[i].FIO);
        scan_str_from_file(file, pin[i].Group);

        fgets(buffer, 50, file);
        pin[i].age = atoi(buffer);
    }

    printf("Сведения о студентах:\n");
    for(int i = 0; i < N; i++)
    {
        printf("Запись #%d\n", i);
        printf("ФИО: %s\n", pin[i].FIO);
        printf("Группа: %s\n", pin[i].Group);
        printf("Возраст: %d\n", pin[i].age);
    }

    int mini = 0;
    for(int i = 1; i < N; i++)
    {
        if(pin[i].age < pin[mini].age)
            mini = i;
    }

    printf("Сведения о самом молодом студенте:\n");
    printf("ФИО: %s\n", pin[mini].FIO);
    printf("Группа: %s\n", pin[mini].Group);
    printf("Возраст: %d\n", pin[mini].age);
}
getchar();
return 0;
}

```

Задание на лабораторную работу 10

Написать программу, согласно следующим требованиям (см. Пример 5):

1. Хранение массива структур (минимум 10 элементов) согласно вариантам, приведенным в таблице 1.
2. Ввод данных в массив структур с клавиатуры.
3. Сохранение массива структур в текстовый файл (каждый элемент структуры с новой строки).
4. Заполнение массива структур из текстового файла.
5. Обработка массива структур согласно заданию варианта.

№ вар.	Задача
1	<p>«Человек»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира). Вывести сведения о самом молодом человеке.</p>
2	<p>«Школьник»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); школа; класс. Вывести сведения про всех учеников пятых классов.</p>
3	<p>«Студент»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); ВУЗ; курс; группа; средний бал; специальность. Вывести сведения про всех студентов, у которых средний балл ниже 70 баллов.</p>
4	<p>«Покупатель»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); номер кредитной карточки; банковского счета. Вывести данные о покупателях с города Муром.</p>
5	<p>«Пациент»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); номер больницы; отделение; номер медицинской карты; диагноз; группа крови. Вывести данные про пациентов с 18 отделения.</p>
6	<p>«Владелец автомобиля»: фамилия; имя; отчество; номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира) марка автомобиля; номер автомобиля; номер техпаспорта. Вывести данные про автомобили марки «Ваз».</p>
7	<p>«Военнослужащий»: фамилия; имя; отчество; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); национальность; дата рождения (год, месяц число); должность; звание. Вывести данные про военнослужащих в звании «лейтенант».</p>

8	<p>«Рабочий»: фамилия; имя; отчество; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); национальность; дата рождения (год, месяц число); № цеха; табельный номер; образование; год поступления на работу. Вывести данные про рабочих, поступивших на работу в 2010 году.</p>
9	<p>«Владелец телефона»: фамилия; имя; отчество; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); № телефона. Вывести данные про владельцев телефона номер, которого начинается на 720.</p>
10	<p>«Абитуриент»: фамилия; имя; отчество; пол; национальность; дата рождения (год, месяц число); домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); оценки по экзаменам; проходной балл. Вывести данные про абитуриентов, проходной балл которых равен больше 4.</p>
11	<p>«Государство»: название страны; столица; государственный язык; население; площадь территории; денежная единица; государственный строй; глава государства. Вывести данные про государства, население которых больше 20 млн жителей.</p>
12	<p>«Автомобиль»: марка; цвет; серийный номер; регистрационный номер; год выпуска; год техосмотра; цена. Вывести данные про автомобили, которым больше 2 лет.</p>
13	<p>«Товар»: наименование; стоимость; срок хранения; сорт; дата выпуска; срок годности. Вывести данные про товары срок годности которых истекает в этом году.</p>
14	<p>«Кинолента»: название; режиссер (фамилия; имя); год выхода; страна; стоимость; доход; прибыль. Вывести данные про фильмы режиссера Ежи Гофмана.</p>
15	<p>«Рейс»: марка автомобиля; номер автомобиля; пункт назначения; грузоподъемность (в тоннах); стоимость единицы груза; общая стоимость груза. Вывести данные про автомобили, грузоподъемность которых больше 2 тонн.</p>

16	<p>«Книга»: название; автор (фамилия; имя); год выхода; издательство; себестоимость; цена; прибыль. Вывести данные про книги авторов, фамилия которых начинается с буквы «К».</p>
17	<p>«Здание»: адрес; тип здания; количество этажей; количество квартир; срок эксплуатации; срок до капитального ремонта (25 лет - срок эксплуатации). Вывести данные про здания срок эксплуатации, которых больше 50 лет.</p>
18	<p>«Программист»: фамилия; имя; отчество; пол; национальность; дата рождения (год, месяц число); образование; номер телефона. Вывести сведения о программистах, которым меньше 25 лет.</p>
19	<p>«Ученый»: фамилия; имя; отчество; пол; национальность; дата рождения (год, месяц число); ученая степень, должность, номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира). Вывести сведения про ученых кандидатов технических наук.</p>
20	<p>«Пенсионер»: фамилия; имя; отчество; пол; национальность; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира). Вывести сведения про всех пенсионеров, которые на пенсии больше 5 лет.</p>
21	<p>«Футболист»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; название команды; номер в команде; амплуа; результативность (количество голов); количество игр. Вывести сведения про футболистов, которые провели за свою команду больше 50 матчей.</p>
22	<p>«Манекенщица»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира). Вывести данные про самую молодую манекенщицу.</p>

23	<p>«Международная компания»: название; интернет сайт; адрес главного офиса (почтовый индекс, страна, область, район, город, улица, дом, квартира) продолжительность пребывания на мировом рынке; количество сотрудников; количество филиалов в Европе. Вывести международные компании, количество сотрудников у которых больше 10000.</p>
24	<p>«Телохранитель»: фамилия; имя; отчество; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); дата рождения (год, месяц число). Вывести данные про старшего телохранителя.</p>
25	<p>«Зоопарк»: Название животного; количество вида; адрес зоопарка (почтовый индекс, страна, область, район, город, улица, дом, квартира); общее количество животных, количество работников. Вывести сведения про зоопарки, в которых есть уссурийские тигры.</p>
26	<p>«Программное обеспечение»: название; название компании производителя; год выхода; цена. Вывести данные про программное обеспечение, которое дороже 2000 рублей.</p>
27	<p>«Мультфильм»: название; режиссер (фамилия; имя); год выхода; страна; стоимость; доход; прибыль. Вывести данные про мультфильмы компании «Walt Disney».</p>
28	<p>«Баскетболист»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; название команды; номер в команде; амплуа; результативность (количество очков); количество игр. Вывести сведения про баскетболистов, которых забросили за свою команду больше 150 очков.</p>
29	<p>«Область»: название области; областной центр; население; площадь территории; губернатор. Вывести данные про области, население которых меньше 2 млн жителей.</p>
30	<p>«Мотоцикл»: марка; цвет; серийный номер; регистрационный номер; год выпуска; год техосмотра; цена. Вывести данные про мотоциклы марки «Harley Davidson».</p>

Содержание отчета

1. Титульный лист.
2. Название и цели работы.
3. Индивидуальное задание
4. Исходный код программы
5. Снимок экрана с открытым в любом текстовом редакторе полученного файла со структурами
6. Снимки экрана с результатами работы программы.
7. Вывод по работе

Список рекомендуемой литературы

1. Двойнишников, С. В. Основы программирования (язык С) : учебное пособие / С. В. Двойнишников, К. Ф. Лысаков. — Новосибирск : Новосибирский государственный университет, 2018. — 157 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/93475.html> (дата обращения: 18.12.2021). — Режим доступа: для авторизир. пользователей. С. 44-60, 73-78