

4 СТРУКТУРЫ, РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ

Введение

Программа должна поддерживать следующие функции. Чтение массива записей из текстового файла, навигация по массиву записей в двух направлениях, добавление записей в массив, редактирование указанной записи, удаление указанной записи, сохранение массива записей в файл, и обработку данных в соответствии с заданием. Для хранения массива записей использовать **динамический массив структур**. Все алгоритмические подзадачи реализовывать в виде отдельных процедур и функций. Для тестирования заготовить файлы, содержащие не менее 10 записей.

Общий сценарий работы приложения следующий: Запускаем приложение - массив данных пуст: пользователь может открыть уже существующий файл и при этом массив данных будет прочитан из файла или перейти к работе с данными, т.е. фактически заполнить массив данных с нуля. Обеспечиваем пользователю возможность работы с данными (навигация, добавление, редактирование и удаление записей). После пользовательской работы с данными даём пользователю возможность сохранить данные в указанный файл или выйти без сохранения.

Для взаимодействия с пользователем предусмотреть двухуровневое текстовое меню. Меню первого уровня (работа с массивом данных): Загрузить данные из файла, очистить массив данных, сохранить данные в файл, сохранить файл результат, работа с данными, выход. При выборе пункта меню «работа с данными», попадаем в меню второго уровня. Меню второго уровня (работа с записями): предыдущая запись, следующая запись, добавить запись после текущей, удалить текущую запись, изменить текущую запись, назад. При выборе пункта меню «назад» попадаем в меню первого уровня.

Работа с текстовыми файлами

Стандартный сценарий работы с файлами включает следующие этапы:

1. Открытие файла.
2. Работа с файлом (чтение/запись данных).
3. Закрытие файла.

Для программиста открытый файл представляется как последовательность считываемых или записываемых данных. При открытии файла с ним связывается поток ввода-вывода. Выводимая информация записывается в поток, вводимая информация считывается из потока. Когда поток открывается для ввода-вывода, он связывается со стандартной структурой типа FILE, которая определена в `stdio.h`. Структура FILE содержит необходимую информацию о файле.

Открытие файла осуществляется с помощью функции `FILE *open(const char * fname, const char * mode)`: `fname` – имя открываемого файла, `mode` – режим открытия файла (см. таблицу), возвращаемое значение – указатель на открытый поток, если открытие файла произошло с ошибкой, то возвращается значение NULL.

Таблица 1. Режимы открытия файла

"r"	открыть файл для чтения (файл должен существовать)
"w"	открыть пустой файл для записи; если файл существует, то его содержимое теряется
"a"	открыть файл для записи в конец (для добавления); файл создается, если он не существует
"r+"	открыть файл для чтения и записи (файл должен существовать)
"w+"	открыть пустой файл для чтения и записи; если файл существует, то его содержимое теряется
"a+"	открыть файл для чтения и дополнения, все операции записи выполняются в конец файла, если файл не существует, то он создаётся

Чтение данных из файла производится с помощью функции **int fscanf(FILE * stream, const char * format, ...)**: **stream** – указатель на файловый поток, из которого будет производиться чтение, **format** – строка формата, **возвращаемое значение** – количество полей, которые успешно преобразованы (в соответствии с форматом) и присвоены.

Достижение конца файла проверяется с помощью функции **int feof(FILE * stream)**: **stream** – указатель на файловый поток, **возвращаемое значение** – 0, если конец файла не достигнут, ненулевое значение в противном случае.

Запись данных в файл производится с помощью функции **int fprintf(FILE * stream, const char * format, ...)**: **stream** – указатель на файловый поток, в который будет производиться запись, **format** – строка формата, **возвращаемое значение** – число записанных байт, если произошла ошибка ввода/вывода, то возвращает -1.

Закрытие файлового потока осуществляется с помощью функции **FILE * fclose(FILE * stream)**: **stream** – указатель на файловый поток, **возвращаемое значение** – 0, если поток успешно закрыт, в противном случае возвращается ненулевое значение. Попытка закрытия уже закрытого файлового потока является ошибкой.

Пример чтения данных из файла:

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

int main() {

    setlocale(LC_ALL, "");

    FILE *input = NULL; //определяем указатель на файловый поток

    input = fopen("D:\\test.txt", "r"); //открываем файл
    if (input == NULL) //ошибка при открытии файла
    {
        printf("Error opening file");
        getch();
        return(-1);
    }
}
```



```

char firstname[255], lastname[255];
unsigned age;

while (!feof(input)) //читаем данные пока не дойдём до конца файла
{
    int r = fscanf(input, "%s\t%s\t%u\n", firstname, lastname, &age);
    if (r == 3) //успешно считано 3 значения, выводим на печать
    {
        printf("First: %s\nLast: %s\nAge: %u\n\n", firstname, lastname, age);
    }
}

fclose(input); //закрываем файловый поток

getchar();
return(0);
}

```

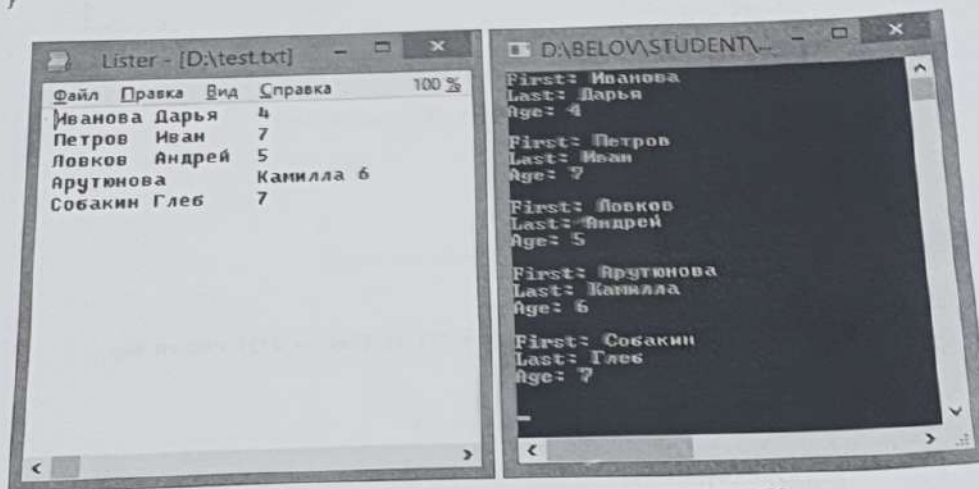


Рисунок 1. Исходные данные и вывод программы

Пример простого двухуровневого меню:

```

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

#define SIZE 10

// основная функция, которая возвращает коды клавиш, в том числе и управляющих
// написана с использованием функции getch() из библиотеки conio.h, главное отличие от
// getchar() - символ не печатается на экране
// кому интересны особенности работы getch() читаем английский текст ниже, переводить
// лень:
// getch() function returns two keycodes for arrow keys (and some other special keys).
// It returns either 0 (0x00) or 224 (0xE0) first, and then returns a code identifying
// the key that was pressed.

```

```

// For the arrow keys, it returns 224 first followed by 72 (up), 80 (down), 75 (left) and
// 77 (right).
// If the num-pad arrow keys (with NumLock off) are pressed, getch() returns 0 first
// instead of 224.
// Список кодов некоторых управляющих клавиш (остальные, если не хватит - гуглятся)
// 13: Enter
// 27: Esc
// 72: Up arrow
// 75: Left arrow
// 77: Right arrow
// 80: Down arrow
// 82: Ins
// 83: Del
int get_key()
{
    int key = getch();
    if ((key == 0) || (key == 224)) key = getch();
    return key;
}

int menu1()
{
    printf("\n");
    printf("Просмотр Enter\nВыход Esc");
    while(true)
    {
        int key = get_key();
        if ((key == 27) || (key == 13)) return key;
    }
}

int menu2()
{
    printf("\n");
    printf("Вперед ->\nНазад <-\nВыход: Esc");
    while(true)
    {
        int key = get_key();
        if ((key == 27) || (key == 75) || (key == 77)) return key;
    }
}

void print_current(const char** data, int i)
{
    system("cls");
    printf("\n");
    printf("\t\t%s", data[i]);
    printf("\n");
}

int main()
{
    const char* data[SIZE] = {"Лось", "Олень", "Заяц", "Волк", "Лошадь", "Слон",
    "Собака", "Динозавр", "Мышь", "Лиса"};
    int current;

    setlocale(LC_ALL, "");
    while (true)
    {
        system("cls");
        printf("Вас приветствует чудо-программа с двухуровневым текстовым
меню.\n");
        printf("Данная реализация не претендует на идеал и демонстрирует
необходимый\n");
    }
}

```

```
printf("минимум в оформлении 4-й лабораторной работы. В тайне автор  
надеется,\n");  
printf("что найдутся те, кто сделает лучше, краше, интереснее...\n");  
int m1 = menu1();  
if (m1 == 27) break;  
current = 0;  
while (true)  
{  
    print_current(data, current);  
    int m2 = menu2();  
    if (m2 == 27) break;  
    switch(m2)  
    {  
        case 75:  
            if (current > 0) current--;  
            break;  
        case 77:  
            if (current < SIZE-1) current++;  
            break;  
    }  
}  
return 0;  
}
```


Варианты

1. В файле хранится информация о студентах: фамилия, имя, номер группы, количество пропусков занятий по неуважительной причине. Создать три новых файла, в которые поместить, упорядоченную по ФИО, информацию о студентах:
 - вызвать в деканат (количество пропусков от K_1 до K_2);
 - объявить выговор (количество пропусков от K_2 до K_3);
 - отчислить (количество пропусков больше K_3).Значения K_1 , K_2 и K_3 задаются пользователем.
2. В файле содержатся сведения о студентах: фамилия, имя, отчество, номер группы и изучаемый язык. Создать три файла («англичан», «французов», «немцев») в которые записать, упорядоченные по ФИО, данные о студентах заданной группы, изучающих соответствующие языки.
3. В файле содержатся данные о туристических маршрутах: название, начальный и конечный пункты, вид маршрута (водный, пеший, горный и т.д.), категория сложности (1-5), продолжительность в днях. Вывести в новый файл данные о маршрутах с категорией сложности не выше указанной, для указанного начального пункта. Данные упорядочить по видам маршрутов, продолжительности.
4. В файле хранится информация о книгах: шифр, ФИО автора, название книги, год издания. В новый файл переписать информацию о книгах, в названии которых встречается заданное слово, упорядочить по названию книги.
5. В файле хранится информация о книгах: шифр, ФИО автора, название книги, издательство, год издания. В новый файл переписать информацию о книгах, выпущенных заданным издательством, упорядочить по году издания, названию книги.
6. В файле хранится информация о студентах: фамилия, имя, номер группы и отметки (4 экзамена и 3 зачёта), полученные в последнюю сессию. В новый файл переписать информацию о лучших студентах (средний балл за экзамены, не меньше, чем средний балл по курсу и сданы все зачеты). Упорядочить данные по ФИО студентов.
7. В файле содержатся сведения о студентах, проживающих в общежитии: фамилия студента, номер комнаты, в которой он проживает, номер группы, в которой он учится, дата прописки. В новый файл переписать, упорядоченную по ФИО, информацию о студентах, которые поселились в общежитии позднее указанной даты.
8. В файле содержатся сведения о телевизорах: марка, модель, размер экрана по диагонали, дата покупки (число, месяц, год), срок гарантии (в месяцах). В новый файл переписать информацию о тех телевизорах, у которых закончился гарантийный срок (на текущую дату), упорядочить, по марке, модели.

9. В файле хранится
название ФИО
ист. В кон
книгах
выд

номер группы, количество
файлов, в которых

9. В файле хранится информация о книгах, взятых в библиотеке: инвентарный номер, название, ФИО автора, дата последней выдачи книги, признак того, сдана книга или нет. В новый файл поместить информацию о всех задолженных на текущую дату книгах, упорядочить по названию. Несданная книга считается задолженной, если выдана более двух недель назад.
10. В файле содержатся сведения об автомобилях: марка, модель, гос. номер, фамилия, имя, отчество владельца и величина пробега. В новый файл переписать, упорядоченную по ФИО владельца, информацию об автомобилях заданной марки, с пробегом менее заданного значения.
11. В файле содержатся сведения о сотрудниках различных учреждений: название учреждения, фамилия, имя, отчество сотрудника, номер телефона. В новый файл вывести, упорядоченные по ФИО сотрудника, данные о всех сотрудниках заданного учреждения.
12. В файле содержатся сведения об игрушках: название, стоимость и возрастные категории детей, для которых игрушка предназначена. В новый файл вывести данные об игрушках для заданной возрастной категории, стоимость которых выше средней стоимости в этой возрастной категории, упорядочить по названию игрушек.
13. В файле содержатся данные о сотрудниках предприятия: фамилия, имя, отчество, должность, дата поступления на работу. В новый файл вывести данные о сотрудниках, у которых в заданном году круглая дата работы на предприятии (5, 10, 15 лет и т.д.), упорядочить по ФИО сотрудника.
14. В файле содержатся данные о владельцах квартир: фамилия, имя, отчество, год, номер квартиры, сумма задолженности по квартплате. В новый файл вывести данные о владельцах, имеющих задолженность по квартплате, больше заданного значения, рассчитать суммарную задолженность, упорядочить по ФИО владельца.
15. В файле содержатся данные о банковских счетах: фамилия, имя и отчество владельца счета, номер счёта, количество средств на счёте, дата открытия счёта. Вывести в новый файл данные о счетах, открытых после указанной даты, рассчитать сумму средств на этих счетах. Данные упорядочить по ФИО владельца.
16. В файле содержатся данные о банковских счетах: фамилия, имя и отчество владельца счета, номер счёта, количество средств на счёте, дата открытия счёта. Вывести в новый файл данные о счетах, количество средств, на которых превышает среднее количество средств по всем счетам. Данные упорядочить по ФИО владельца.
17. В файле содержатся данные телефонной книги: фамилия, имя, отчество и номер абонента. В новый файл переписать данные обо всех абонентах, фамилия которых начинается на указанную букву, упорядочить их по ФИО.
18. В файле содержатся данные о блюдах: название блюда, категория блюда, стоимость. Вывести в новый файл данные о блюдах каждой категории, стоимость которых выше средней стоимости блюд данной категории. Данные упорядочить по названию категории блюда.

19. В файле содержатся сведения об абитуриентах: фамилия, имя, отчество, баллы (1-5) за 3 вступительных экзамена. В новый файл переписать данные о тех абитуриентах, средний балл которых удовлетворяет заданному проходному баллу, упорядочить по ФИО абитуриентов.

20. В файле содержатся данные о продуктах: название товара, цена за единицу, количество, дата изготовления, срок годности (в днях). Вывести в новый файл данные о продуктах, просроченных на текущую дату, рассчитать общую стоимость просроченных продуктов. Данные упорядочить по наименованию.

21. В файле содержатся данные о турах: город, категория отеля (1-5), количество дней, стоимость тура. Вывести в новый файл данные о турах, стоимость проживания в день для которых ниже, чем средняя стоимость проживания в день по всем предложениям для этого города. Данные упорядочить по городам.

22. В файле хранится информация о студентах: фамилия, имя, отчество, номер группы и отметки (4 экзамена и 3 зачёта), полученные в последнюю сессию. Создать три новых файла, в которые поместить информацию о студентах, упорядоченную по ФИО, которые:

- будут получать стипендию (все оценки 4 или 5, сданы зачеты);
- будет назначен «исправительный срок» (две и менее «задолженности»);
- будут отчислены по результатам сессии (три и более «задолженности»);

«задолженность» - оценка 1 или 2 или не сданный зачет.

23. В файле содержатся сведения о пассажирах авиалайнера: номер рейса, фамилия, имя, отчество, дата вылета, номер и серия паспорта. В новый файл переписать информацию о пассажирах, прилетевших заданным рейсом в заданный диапазон дат, упорядочить по ФИО пассажиров.

24. В файле содержатся сведения об автомобилях: марка, гос. номер, фамилия владельца, дата выпуска автомобиля и дата прохождения последнего техосмотра (месяц, год). В новый файл переписать информацию об автомобилях, у которых просрочен техосмотр на текущую дату (автомобили проходят техосмотр через три года после выпуска, затем через два года, затем ежегодно). Данные упорядочить по фамилии владельца.

25. В файле содержатся данные о владельцах квартир: фамилия, имя, отчество, дата регистрации права собственности, номер квартиры. В новый файл вывести данные о владельцах, которые зарегистрировали право собственности в указанный диапазон дат, упорядочить по ФИО владельца.

26. В файле содержатся данные о многоквартирных домах: адрес, число квартир, общая площадь квартир, наименование обслуживающей организации. Вывести в новый файл данные о домах, число квартир в которых больше указанного значения, данные упорядочить по наименованию обслуживающей организации, общей площади квартир.