

## Лекция 22

# Обработка бинарных файлов

---

Преподаватель Палехова Ольга Александровна,  
кафедра О7 БГТУ «Военмех»

# Бинарные (двоичные) файлы

---

## Особенности:

- можно читать и записывать любой кусок памяти (просто байты);
- данные хранятся во внутреннем **машинном формате** (в текстовом редакторе не прочитать).

## Следствия:

- однотипные данные будут занимать одинаковые объемы памяти;
- создание, просмотр и редактирование файла выполняется специализированной программой.

# Бинарные (двоичные) файлы

**Задача 1.** Дан бинарный файл, содержащий целые числа в формате *int*. Поменять знак у каждого пятого числа на противоположный.

**Особенность:** тестовый файл сначала нужно создать.

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

```
int main()
```

```
{
```

```
    FILE *f;
```

```
    int n, x;
```

```
    if ( (f = fopen ( "int.bin", "wb" )) == NULL )
```

```
    {
```

```
        perror("Ошибка открытия файла"); return 2;
```

```
    }
```

```
    printf ("Сколько чисел записать в файл?\n");
```

```
    scanf ("%d", &n);
```

```
    printf ("Введите %d чисел\n", n);
```

```
    while ( n-- )
```

```
    {
```

```
        scanf ("%d", &x);
```

```
        fwrite (&x, sizeof(int), 1, f);
```

```
    }
```

```
    fclose (f);
```

```
    return 0;
```

```
}
```

создать бинарный файл

записать данные  
без преобразования

# Обработка бинарного файла

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

```
void print_file (FILE *);
```

прототип функции печати  
содержимого файла

```
int main()
```

```
{
```

```
FILE *f;
```

```
int size = sizeof(int), offset = 4*size, end_file, x;
```

```
if ( (f = fopen ( "int.bin", "rb+" )) == NULL )
```

```
{
```

```
    perror("Ошибка открытия файла"); return 1;
```

```
}
```

открыть бинарный файл  
для редактирования

```
printf ("Исходный файл:\n");
```

```
print_file (f);
```

печать исходного файла

```
end_file = ftell(f);
```

размер файла

```
while ( offset < end_file )
```

```
{
```

```
    fseek (f, offset, SEEK_SET);
```

переместиться к нужному числу

```
    fread (&x, size, 1, f);
```

прочитать

```
    x = -x;
```

```
    fseek (f, -size, SEEK_CUR);
```

вернуться к началу числа

```
    fwrite (&x, size, 1, f);
```

записать число

```
    offset += 5*size;
```

```
}
```

```
printf ("Измененный файл:\n");
```

вычислить положение следующего

```
print_file (f);
```

```
fclose (f);
```

печать измененного файла

```
return 0;
```

```
}
```

# Бинарные (двоичные) файлы

Функция *печати* содержимого файла.

```
void print_file (FILE * f)
{
    int x, sz = sizeof(int);
    rewind (f);
    while ( fread (&x, sz, 1, f) )
        printf ("%d ", x);
}
```

вернуться к началу  
файла

прочитать данные  
без преобразования

# Бинарный файл записей

**Задача 2.** Дан файл, содержащий сведения о клиентах одного кредитора. Структура записи: ФИО, номер телефона, возраст, размер кредита. Получить сведения о самом пожилом клиенте, задолжавшем более указанной суммы.

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

```
struct person
```

```
{
    char name[30];
    char phone[13];
    char age;
    float money;
};
```

описание  
структурного типа

```
void add (char*); /* добавить запись в файл */
void edit (char*); /* редактировать запись */
void show (char*); /* просмотреть содержимое файла */
void search (char*); /* найти сведения о клиенте */
void del (char*); /* удалить запись из файла */
```

# Бинарный файл записей

---

```
int main (void)
{
    char filename[50];
    int menu;
    puts ("Имя файла:");
    gets (filename);
    do
    {
        system ("CLS");
        puts ("1. Добавление записи");
        puts ("2. Редактирование записи");
        puts ("3. Просмотр записей");
        puts ("4. Поиск");
        puts ("5. Удаление записи");
        puts ("6. Выход");
        scanf ("%d%c", &menu);
        switch (menu)
        {
            case 1 : add (filename); break;
            case 2 : edit (filename); break;
            case 3 : show (filename); break;
            case 4 : search (filename); break;
            case 5 : del (filename);

        }
    }
    while (menu!=6);
    return 0;
}
```

# Бинарный файл записей

---

```
/*заполнение одной записи (ввод данных о клиенте)*/
struct person input_client (void)
{
    struct person client;
    int temp;
    puts ("ФИО");
    gets (client.name);
    puts ("Телефон");
    gets (client.phone);
    puts ("Возраст");
    scanf ("%d", &temp);
    client.age = temp;
    puts ("Сумма");
    scanf ("%f", &client.money);
    return client;
}

/* функция вывода на экран одной записи */
void print1 (struct person data)
{
    printf ("ФИО : %s\nТелефон : %s\nВозраст : %d\nСумма : %.2f\n",
           data.name, data.phone, data.age, data.money);
}
```



# Бинарный файл записей

Функция **добавления** записи в файл

```
void add (char* file)
{
    FILE * f;
    struct person client;
    system ("CLS");
    if ((f=fopen (file, "ab"))==NULL)
    {
        perror ("Ошибка открытия файла");
        getchar();
        return;
    }
    client = input_client();
    fwrite (&client, sizeof (client), 1, f);
    fclose (f);
}
```

открыть бинарный файл  
для добавления данных

записать структуру  
целиком

# Бинарный файл записей

Функция *редактирования* записи

```
void edit (char* file)
{
    FILE * f;
    struct person client;
    char name[30], yes;
    system ("CLS");
    if ((f=fopen (file, "rb+"))==NULL)
    {
        perror ("Ошибка открытия файла"); getchar();
        return;
    }
    printf ("ФИО: "); gets (name);
    while (fread(&client, sizeof(client), 1, f))
        if (!strcmp(client.name, name))
        {
            printf ("client");
            puts ("Редактировать? (y/n)");
            do
            {
                yes = getchar(); getchar();
            }
            while (yes!='y' && yes!='Y' && yes!='n' && yes!='N');
```

открыть бинарный файл  
для чтения и  
редактирования

прочитать структуру  
целиком

# Бинарный файл записей

```
printl (client);  
puts ("Редактировать? (y/n)");  
do  
{  
    yes = getchar(); getchar();  
}  
while (yes!='y' && yes!='Y' && yes!='n' && yes!='N')  
if (yes=='y' || yes=='Y')  
{  
    client = input_client();  
    fseek (f, -sizeof(client), SEEK_CUR);  
    fwrite (&client, sizeof (client), 1, f);  
    fseek (f, 0, SEEK_CUR);  
}
```

вернуться к началу  
записи и перезаписать

для перехода от  
записи к чтению

```
puts ("Поиск завершен"); getchar();  
fclose(f);  
}
```

!

При переходах от чтения к записи и обратно  
нужно вызывать функцию позиционирования

# Бинарный файл записей

Функция *печати* содержимого файла

```
void show (char* file)
{
    FILE * f;
    struct person client;
    int k=0;
    system ("CLS");
    if ((f=fopen (file, "rb"))==NULL)
    {
        perror ("Ошибка открытия файла");
        getchar();
        return;
    }
    puts ("|   N   |           ФИО           |   Телефон   | Возраст |   Сумма   |");
    puts ("-----");
    while (fread(&client, sizeof(client), 1, f)==1)
        printf ("|%3d | %-30.30s |%13s |%8d |%10.2f |\n",
                ++k, client.name, client.phone, client.age, client.money);
    puts ("-----");
    fclose (f);
    getchar();
}
```

открыть бинарный файл  
для чтения данных

# Бинарный файл записей

Функция *поиска* данных

открыть бинарный файл  
для чтения данных

```
void search (char* file)
{
    FILE * f;
    struct person client, required;
    float money;
    int max_age = 0;
    system ("CLS");
    if ((f=fopen (file, "rb"))==NULL)
    {
        perror ("Ошибка открытия файла"); getchar();
        return;
    }
    puts ("Сумма?"); scanf ("%f",&money);
    while (fread(&client, sizeof(client), 1, f)==1)
        if (client.money > money && client.age > max_age)
        {
            max_age = client.age;
            required = client;
        }
    if (max_age == 0) puts ("Никто столько не должен");
    else
        printf ("required");
    fclose(f);
    getchar();getchar();
}
```

# Бинарный файл записей

## Функция *удаления* записи из файла

```
void del (char* file)
{
    FILE *f1, *f2;
    unsigned int n, n_file, i = 0;
    struct person client;
    system ("CLS");
    if ((f1=fopen (file, "r"))==NULL)
    {
        perror ("Ошибка открытия файла"); getchar();
        return;
    }
    fseek (f1, 0, SEEK_END);
    n_file = ftell(f1) / sizeof (client);
    if (n_file == 0)
    {
        puts ("Файл пуст"); getchar();
        return;
    }
    puts ("Номер удаляемой записи?");
    scanf ("%u", &n);
```

открыть исходный бинарный файл для чтения данных

# Бинарный файл записей

```
if ( n > n_file )
{
    puts ("Некорректный номер");
    getchar();getchar();
    fclose (f1);
    return;
}
rewind (f1);
if ((f2=fopen ("temp.dat", "wb"))==NULL)
{
    perror ("Ошибка открытия файла");
    getchar();
    return;
}
while ( fread(&client, sizeof(client), 1, f1)==1 )
{
    i++;
    if ( i != n )
        fwrite(&client, sizeof(client), 1, f2);
}
fclose (f1);
fclose (f2);
remove (file);
rename ("temp.dat", file);
puts ("Запись удалена");
getchar();getchar();
}
```

создать новый  
бинарный файл

переписать в новый файл все  
записи, кроме удаляемой

!

Высокоуровневой функции  
уменьшения размера файла  
в языке Си нет