



ФАЙЛЫ

Лекция №12

ИСХОДНЫЕ ПРЕДПОСЫЛКИ

- Необходимость хранения информации во внешней памяти
- Необходимость обмена информацией с внешней памятью
- Необходимость распознавания формата данных



ФАЙЛ НА ЯЗЫКЕ СИ

- Последовательность данных
- С открытым файлом связывается поток ввода-вывода
 - Выводимая информация записывается в поток
 - Вводимая информация считывается из потока
- Поток связан со структурой типа `FILE`, определенной в `stdio.h`



ФОРМАТЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ В ФАЙЛАХ

○ Текстовый

- Удобен для восприятия
- Избыточен

○ Двоичный

- Компактен
- Требует специального программного обеспечения для восприятия



ФУНКЦИИ ДЛЯ РАБОТЫ С ФАЙЛАМИ

- `fopen()`
- `fclose()`
- `fscanf`
- `fprintf`



ОТКРЫТИЕ ФАЙЛА

- FILE* fopen(name, type)
- name — имя файла
- type — тип доступа
 - "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, если файл не существует, то он создаётся.



ПРИМЕР РАБОТЫ С ФАЙЛОМ

```
#include <stdio.h>
int main() {
    FILE *fp;
    char name[]="my.txt";
    if(fp = fopen(name, "r")!=NULL) { // открыть файл
        //удалось?
        ...           // требуемые действия над данными
    } else printf("Не удалось открыть файл");
    fclose(fp);
    return 0;
}
```



ПЕРЕНОС ИНФОРМАЦИИ С ФАЙЛА В ФАЙЛ

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    FILE *S1, *S2;
    int x, y;
    system("chcp 1251");
    system("cls");
    printf("Введите число: ");
    scanf("%d", &x);
    S1 = fopen("S1.txt", "w");
    fprintf(S1, "%d", x);
    fclose(S1);
    S1 = fopen("S1.txt", "r");
    S2 = fopen("S2.txt", "w");
    fscanf(S1, "%d", &y);
    y += 3;
    fclose(S1);
    fprintf(S2, "%d\n", y);
    fclose(S2);
    return 0;
}
```



ПОТОКОВЫЙ ВВОД-ВЫВОД

- заголовочный файл `<fstream>`
- `<ifstream>` - файловый ввод ;
- `<ofstream>` - файловый вывод.



ЭТАПЫ РАБОТЫ С ФАЙЛОМ

- создать объект класса `fstream` (возможно, `ofstream` или `ifstream`);
- связать объект класса `fstream` с файлом, который будет использоваться для операций ввода-вывода;
- осуществить операции ввода-вывода в файл;
- закрыть файл.

- ```
#include <fstream>
using namespace std;
int main()
{
 ofstream fout;
 fout.open("file.txt");
 fout << "Привет, мир!";
 fout.close();
 return 0;
}
```



# ПРОИЗВОЛЬНЫЙ ДОСТУП К ФАЙЛУ

- Установка указателя
- `ifstream &seekg(Смещение, Позиция);`
- `ofstream &seekp(Смещение, Позиция);`
- Определения положения указателя
- `streampos tellg()` - позиция для ввода
- `streampos tellp()` - позиция для вывода



# КОНСТАНТЫ РЕЖИМА ОТКРЫТИЯ ФАЙЛОВ

- `ios::in` открыть файл для чтения
- `ios::out` открыть файл для записи
- `ios::ate` при открытии переместить указатель в конец файла
- `ios::app` открыть файл для записи в конец файла
- `ios::trunc` удалить содержимое файла, если он существует
- `ios::binary` открытие файла в двоичном режиме



# ПРИМЕР ПРОИЗВОЛЬНОГО ДОСТУПА

- ```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    char s[80];
    fstream inOut;
    inOut.open("file.txt", ios::out);
    inOut << "строка текста" << endl;
    inOut.seekp(8, ios::beg);
    inOut << "еще строка текста";
    inOut.close();
    inOut.open("file.txt", ios::in);
    inOut.seekg(-6, ios::end);
    inOut >> s;
    inOut.close();
    cout << s;
    cin.get();
    return 0;
}
```



ДВОИЧНЫЕ ФАЙЛЫ - ЗАПИСЬ

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#define ERROR_FILE_OPEN -3
void main() {
    FILE *output = NULL;
    int number;
    output = fopen("D:/c/output.bin", "wb");
    if (output == NULL) {
        printf("Error opening file");
        getch();
        exit(ERROR_FILE_OPEN);
    }
    scanf("%d", &number);
    fwrite(&number, sizeof(int), 1, output);
    fclose(output);
    _getch();
}
```



ДВОИЧНЫЕ ФАЙЛЫ - ЧТЕНИЕ

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#define ERROR_FILE_OPEN -3
void main() {
    FILE *input = NULL;
    int number;
    input = fopen("D:/c/output.bin", "rb");
    if (input == NULL) {
        printf("Error opening file");
        getch();
        exit(ERROR_FILE_OPEN);
    }
    fread(&number, sizeof(int), 1, input);
    printf("%d", number);
    fclose(input);
    _getch();
}
```



УПРАВЛЕНИЕ УКАЗАТЕЛЕМ В ДВОИЧНОМ ФАЙЛЕ

- `int fseek (FILE * stream, long int offset, int origin);`
- `fseek` принимает в качестве аргументов указатель на поток и сдвиг в `offset` байт относительно `origin`. `origin` может принимать три значения
- **SEEK_SET** - начало файла
- **SEEK_CUR** - текущее положение файла
- **SEEK_END** - конец файла. К сожалению, стандартом не определено, что такое конец файла, поэтому полагаться на эту функцию нельзя.



ЗАПИСЬ ЧИСЛА И ЧТЕНИЕ ЕГО

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#define ERROR_FILE_OPEN -3
void main() {
    FILE *iofile = NULL;
    int number;

    iofile = fopen("D:/c/output.bin", "w+b");
    if (iofile == NULL) {
        printf("Error opening file");
        getch();
        exit(ERROR_FILE_OPEN);
    }
    scanf("%d", &number);
    fwrite(&number, sizeof(int), 1, iofile);
    fseek(iofile, 0, SEEK_SET);
    number = 0;
    fread(&number, sizeof(int), 1, iofile);
    printf("%d", number);
    fclose(iofile);
    _getch();
}
```



ФУНКЦИИ УПРАВЛЕНИЯ ПОЗИЦИЕЙ В ДВОИЧНОМ ФАЙЛЕ

- `int fgetpos (FILE * stream, fpos_t * pos);`
- `int fsetpos (FILE * stream, const fpos_t * pos);`



//ПРИМЕР ИСПОЛЬЗОВАНИЯ ФУНКЦИИ FGETPOS

```
#include <cstdio>
#include <iostream>
int main()
{ FILE * ptrFile = fopen ("file.txt","r");
  fpos_t pos;
  if (ptrFile == NULL) perror ("Ошибка открытия файла");
  else
  { std::cout << "первый символ: " << (char)fgetc(ptrFile) << "\n";
    fgetpos (ptrFile,&pos);           // получить текущую позицию в файле
    for (int counter = 0; counter < 4; counter++)
    { fsetpos (ptrFile,&pos);          // изменить позицию
      std::cout << "Второй символ: " << (char)fgetc (ptrFile) << "\n";
    }
    fclose (ptrFile);                 // закрыть файл
  }
  return 0;
}
```



//ПРИМЕР ИСПОЛЬЗОВАНИЯ ФУНКЦИИ FSETPOS

```
#include <stdio>
#include <iostream>
int main()
{
    FILE * ptrFile = fopen("file.txt","w");
    fpos_t position;
    fgetpos (ptrFile, &position);           // определить текущее
        положение в файле
    fputs("Посмотрел фильм "Вспомнить всё". Вспомнил, что не
        выключил газ.",
        ptrFile);                           // записать строку в файл
    fsetpos(ptrFile, &position);             // изменить положение
        указателя
    fputs ("Сходил на ", ptrFile);          // заменить слово
        "Посмотрел" на фразу "Сходил на "
    fclose (ptrFile);
    return 0;
}
```



ВЫВОДЫ

- Для работы с файлом формируется поток ввода или вывода
- Имеется возможность работать с текстовыми ил или двоичными файлами
- Есть возможность прямого доступа к элементам файла и управления указателем на текущий элемент файла

