

# Программирование на языке C++

...

Работа с файлами

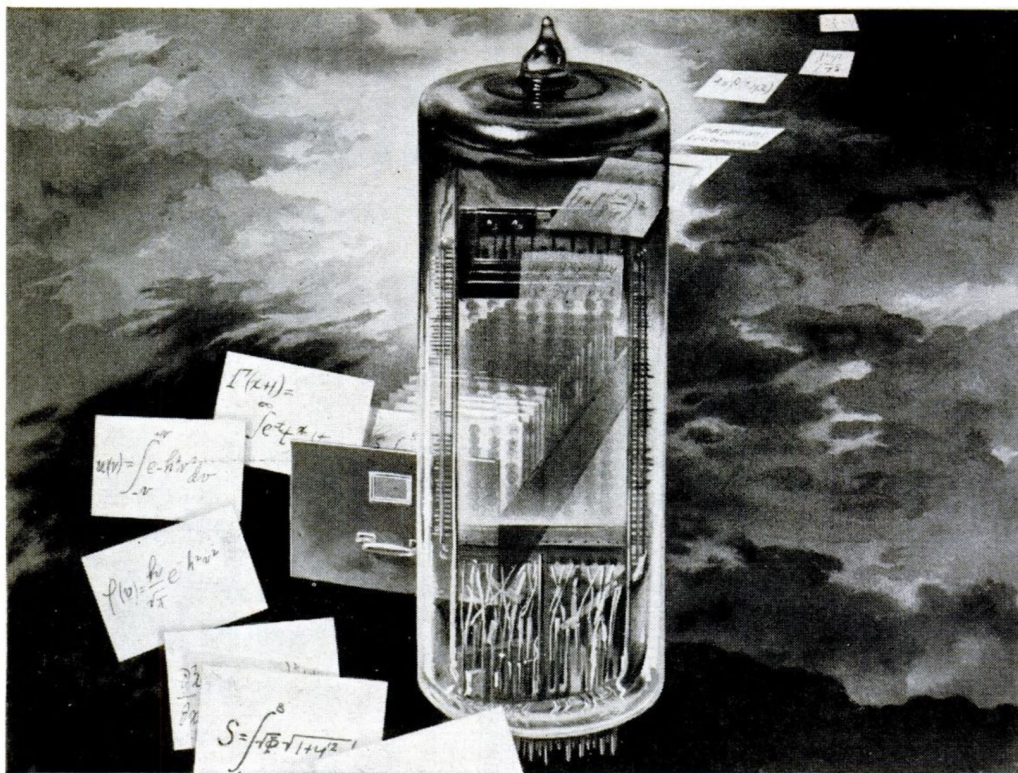
# Файл

Именованная область данных на носителе информации, используемая как базовый объект взаимодействия с данными в операционных системах

Слово file впервые применено к компьютерной системе хранения в 1950 году. Реклама памяти на запоминающих ЭЛТ фирмы RCA в журнале Popular Science гласила:

“...результаты бесчисленных вычислений можно держать «в картотеке» (англ. on file) и получать снова. Эта «картотека» теперь существует в запоминающей трубке, разработанной в лабораториях RCA. Она электрически сохраняет цифры, отправленные в вычислительную машину, и держит их в хранилище, заодно запоминая новые — ускоряя интеллектуальные решения в лабиринтах математики.”

Реклама памяти на запоминающих  
ЭЛТ фирмы RCA в журнале Popular  
Science (1950)



New RCA electron tube gives today's amazing  
computing machines an indispensable memory.

***Tube with a memory keeps answers on file***

# Перфокарта (перфорационная карта)

Носитель информации из тонкого картона, представляет информацию наличием или отсутствием отверстий в определённых позициях карты



Компьютеры первого поколения использовали перфокарты в качестве основного носителя при хранении и обработке данных. В течение 1970-х - начале 1980-х годов, они использовались только для хранения данных и постепенно были замещены магнитными лентами.

Поначалу словом **File** называли  
само устройство памяти, а не его  
содержимое

Пример - IBM 350 Disk Storage Unit  
Model 1 (1956) использовался в  
вычислительной машине IBM 305,  
и назывался Disk Files



# Файловая система

Способ организации данных на носителях информации

Файловая система определяет, где и каким образом на носителе будут записаны файлы, и предоставляет операционной системе доступ к этим файлам

В 1961 году в операционной системе Burroughs MCP и MIT Compatible Time-Sharing System была представлена концепция "file system", которая управляла несколькими виртуальными "files" на носителе информации.

Имена файлов в Compatible Time-Sharing System состояли из двух частей, «основного имени» и «дополнительного имени»  
(расширение имени файла)

# Работа с файлами в C++

Стандартная библиотека `fstream`

Библиотека `fstream` предоставляет 3 типа данных для работы с файлами:

`ofstream` - выходной файловый поток для создания файлов и записи информации в них

`ifstream` - входной файловый поток для чтения информации из файлов

`fstream` - общий файловый поток. Предоставляет возможности как для чтения информации из файлов, так и записи

# Открытие и закрытие файла

```
void open (const char* filename,  
           ios_base::openmode mode);
```

```
void close();
```

*Любой открытый файл будет  
закрыт автоматически в момент  
разрушения объекта `fstream`*

```
#include <fstream>  
  
int main () {  
    std::fstream fs;  
  
    fs.open ("test.txt", std::fstream::in | std::fstream::out);  
  
    fs << "Some portion of data" << std::endl;  
    fs << "Pi = " << 3.1415f << std::endl;  
  
    fs.close();  
}
```



# Режим открытия файла

```
void open (const char* filename,  
           ios_base::openmode mode);
```

<b>in</b> (Input)	File open for reading: the internal stream buffer supports input operations
<b>out</b> (Output)	File open for writing: the internal stream buffer supports output operations
<b>binary</b> (Binary)	Operations are performed in binary mode rather than text
<b>ate</b> (At End)	The output position starts at the end of the file
<b>app</b> (Append)	All output operations happen at the end of the file, appending to its existing contents
<b>trunc</b> (Truncate)	Any contents that existed in the file before it is open are discarded

# Проверка был ли открыт файл

`bool is_open();`

```
#include <iostream>
#include <fstream>

int main () {
    std::fstream fs;
    fs.open("output.dat");
    if (!fs.is_open())
    {
        std::cout << "output.dat wasn't opened";
        return 0;
    }

    fs << "Some important data";
    fs.close();
}
```

# Посимвольное чтение файла

`istream& get (char& c);`

```
#include <iostream>
#include <fstream>

int main () {
    std::ifstream ifs("input.dat");
    if (!ifs.is_open())
    {
        std::cout << "input.dat wasn't opened";
        return 0;
    }

    char c;
    while (ifs.get(c)) {
        std::cout << c;
    }

    ifs.close();
}
```

# Построчное чтение файла

`istream& getline (char* s,  
streamsize n);`

```
#include <iostream>
#include <fstream>

int main () {
    std::ifstream ifs("input.dat");
    if (!ifs.is_open())
    {
        std::cout << "input.dat wasn't opened";
        return 0;
    }

    char str[1024];
    while (ifs.getline(str, 1024)) {
        std::cout << str << std::endl;
    }

    ifs.close();
}
```

# Чтение файла с разделителями

`istream& getline (char* s,  
streamsize n, char delim );`

```
#include <iostream>
#include <fstream>

int main () {
    std::ifstream ifs("delimInput.dat");
    if (!ifs.is_open())
    {
        std::cout << "delimInput.dat wasn't opened";
        return 0;
    }

    char str[1024];
    while (ifs.getline(str, 1024, ';')) {
        std::cout << str << std::endl;
    }

    ifs.close();
}
```

# Чтение файла целиком

`istream& read(char* s, streamsize n);`

`basic_istream& seekg( off_type off,  
std::ios_base::seekdir dir);`

`pos_type tellg();`

```
#include <iostream>
#include <fstream>

int main () {
    std::ifstream ifs("input.dat");
    if (!ifs.is_open())
    {
        std::cout << "input.dat wasn't opened";
        return 0;
    }

    ifs.seekg(0, std::ios::end);
    const size_t fileSize = ifs.tellg();

    std::string str(fileSize, ' ');
    ifs.seekg(0);
    ifs.read(&str[0], fileSize);

    std::cout << str << std::endl;

    ifs.close();
}
```

# Установка позиции в файле

```
basic_istream& seekg( off_type off,  
    std::ios_base::seekdir dir);
```

`ios_base::beg`

From the beginning of the stream's buffer

`ios_base::cur`

From the current position in the stream's buffer

`ios_base::end`

From the end of the stream's buffer