

# Файлове

Трифон Трифонов

Структури от данни и програмиране,  
спец. Компютърни науки, 2 поток, 2015/16 г.

16 октомври 2015 г.



dreamstime.com

# Какво е файл?

- Блок информация, записана на траен носител
- Разлика между масив и файл
- Файлови системи
- Метаданни на файла

# Файлът като поток

- Последователен достъп
- Еднопосочно обхождане
- Еднократна обработка
- Краен поток
- Файлът може да играе ролята на
  - производител (входни файлове)
  - консуматор (изходни файлове)

# Файлът не е само поток

- Пряк достъп
- Разширяване при запис
- Едновременно четене и запис

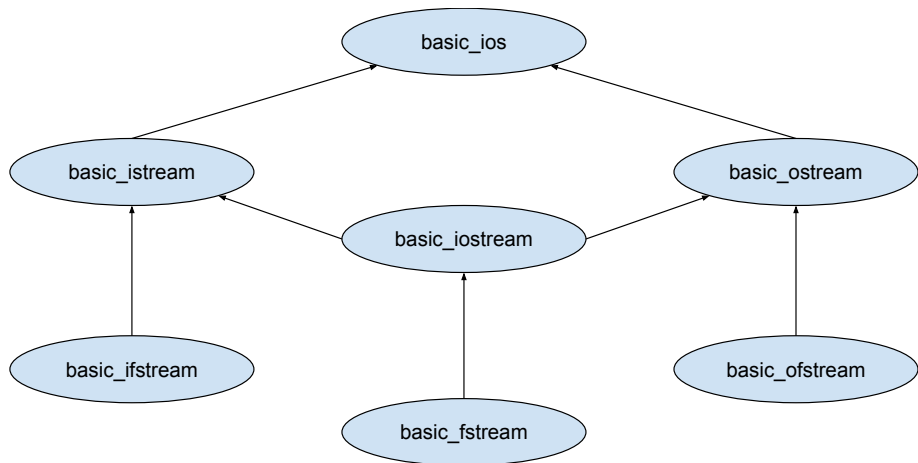
# Текстови файлове

- Форматиран вход и изход
- Само последователен достъп
- Еднократно обхождане
- Интерпретиране на данните във файла като текст (ASCII, Unicode или др.)
- Прилича на низ

# Двоични файлове

- Неформатиран (суров) вход и изход
- Позволява пряк достъп
- Многократно обхождане
- Интерпретацията на данните във файла зависи от конкретната задача
  - масив от числа
  - структура
  - масив от структури

## Поточна йерархия в C++



## Входни файлове

```
ifstream(char const*, openmode = ios::in )
```

- `void open(char const*, openmode = ios::in)`
- `void close()`
- `ios::binary` — суров (неформатиран) вход

Примери:

```
ifstream fi("email.txt", ios::in );  
ifstream fi("lolcat.jpg", ios::in | ios::binary );
```



## Исходни файлове

```
ofstream(char const*, openmode = ios::out|ios::trunc)
```

- `void open(char const*, openmode)`
- `void close()`
- `ios::trunc` — отрязва (унищожава) файла
- `ios::ate` — вмъкването става в края
- `ios::app` — вмъкването винаги е в края

Примери:

```
ofstream fo("page.html", ios::out );  
ofstream fo("application.log", ios::out | ios::app );  
ofstream fo("file.dat", ios::out | ios::binary );
```

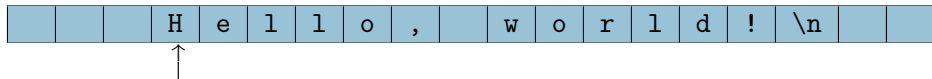
## Входно-изходни файлове

```
fstream(char const*, openmode = ios::in | ios::out)
```

Пример:

```
fstream f( "essay.txt" );  
f.getline(line, 100);  
f << "Ignore the following text, please!";
```

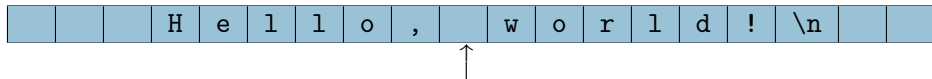
## Файлов указател



A diagram illustrating the memory layout of a string. The string "Hello, world! \n" is stored in a sequence of memory cells. Each character, including the comma, space, and backslash, occupies its own cell. The backslash and 'n' are shown together in a single cell, representing the escape sequence for a newline character. An arrow points to the first cell containing the character 'H', indicating the starting address of the string.

			H	e	l	l	o	,		w	o	r	l	d	!	\n		
--	--	--	---	---	---	---	---	---	--	---	---	---	---	---	---	----	--	--

## Файлов указател



## Пряк достъп до файлове

Отправна точка за преместване на текущата позиция:

seekdir	beg	cur	end
---------	-----	-----	-----

Селектори:

```
streampos tellg() const
```

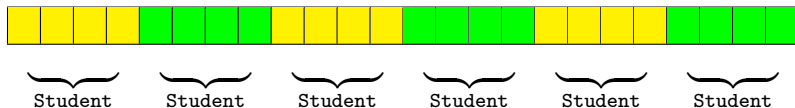
```
streampos tellp() const
```

Мутатори:

```
istream& seekg(streampos, seekdir = beg)
```

```
ostream& seekp(streampos, seekdir = beg)
```

# Блокова организация



```
class Student { ... };
```

```
Student s;
```

```
f.seekp( i * sizeof (Student) );
```

```
f.write((char const*)&s, sizeof(Student));
```

```
Student sa[3];
```

```
f.seekg( j * sizeof(Student) );
```

```
f.read( (char*)sa, 3 * sizeof(Student));
```

# Задача “СУСИ”

- Да се въведе списък от студенти
- Да се запише в текстов файл `students.txt`
- От `students.txt` да се прочетат студентите, които не са скъсани и да се запишат в главната книга `main.bk`
- В главната книга да се повиши с 1.0 оценката на студент с даден Ф№