

3. Работа с `fstream` - особености при отваряне на файл за четене и писане. Двоични файлове. Запазване на обекти в двоичен файл. Четене на обекти от двоичен файл.

## `fstream`

- ~~позволява~~ позволява писане и четене едновременно
- има само един указател  
`tellg() == tellp();`

⚠ при потоците работят с вътрешен буфер, който при достигане на определен капацитет записва. Когато извикан `put()` трябва да синхронизиране за да направим състоянието правилно `flush()`.

Пр: `fstream fs("file.txt", ios::in | ios::out);`

`fs.put("a");`

`fs.flush();`

`fs.seekg(ios::begin, 0);`

`fs.get();`

## Двоични файлове

- поредица от байтове
- ~~работи~~ използва само него
- по-малък от нормален файл
- по-бърз - ~~връх~~ при четене от него

⚠ `ios::binary`

- отваря файл в двоичен формат
- чете байт по байт  
и чете `'\n'` и `'\r'`,  
с което мата специален символ

Пр: `file.txt`  
121

Число → стринг → число

`file.dat`

`*[A-Z][0-9]{2,4}` → число

## Запазване на обекти в двоични файлове

- ключова ~~данна~~ <sup>ф-ция</sup> `write(<указател>, <дължина>);`

`const char*`  
от указател,  
избран е  
защото е байт и солов, защото  
има да бъде променен

↑  
брои `byte-ове`,  
които трябва да  
бъдат запазени



Пр: `int x=16;`  
`.write( (const char *) &x, sizeof(int));`

⚠ в двоични файлове чисел в endianness-а на системата

file.dat  

01	00	00	00
----	----	----	----

 ← hex view of the file

Пр: писане на:

- статичен обект
- обект с външен ресурс
- статичен масив
- масив от обекти с динамичен ресурс

Четење на обекти от двоичен файл

- ф-ция `.read (<указател>, <должина>);`  
↓  
char без const, защото ще пишем  
в тези данни  
размер, който искаме да прочетем

Пр: `ifstream ifs ("file.dat", ios::binary | ios::in);`  
`int z;`  
`ifs.read( (const char *) &z, sizeof(z));`

`z = 16;`

Пр: четене на:

- статичен обект
- обект с външен ресурс
- статичен масив
- масив от обекти с динамичен ресурс