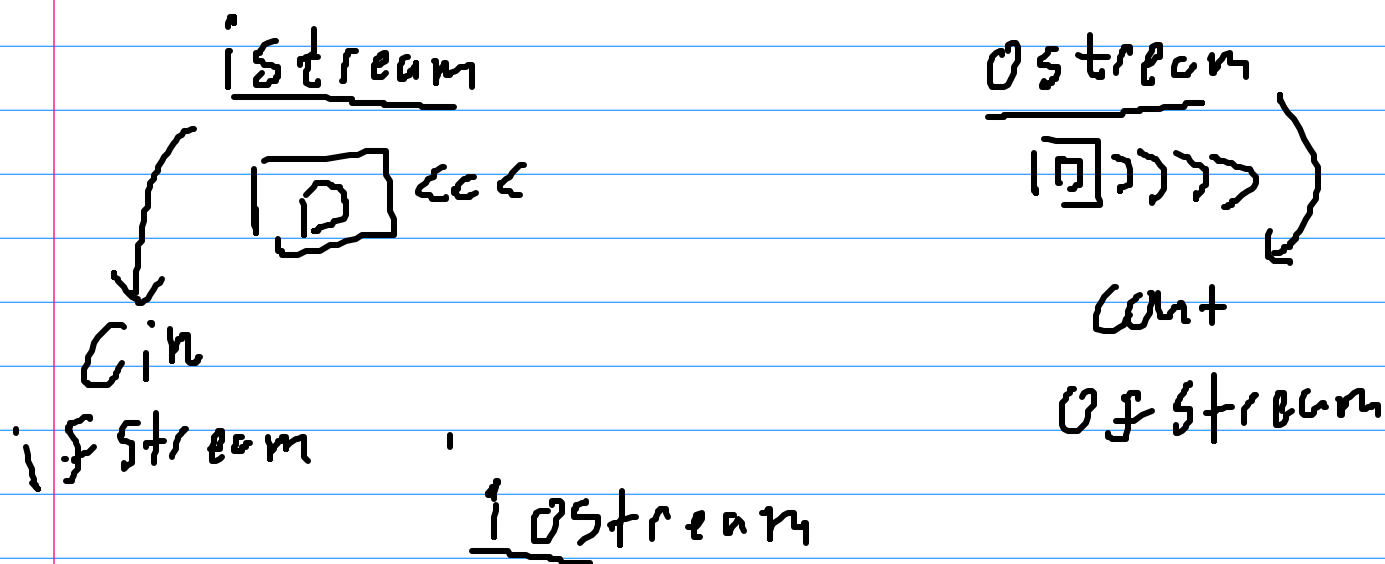


3

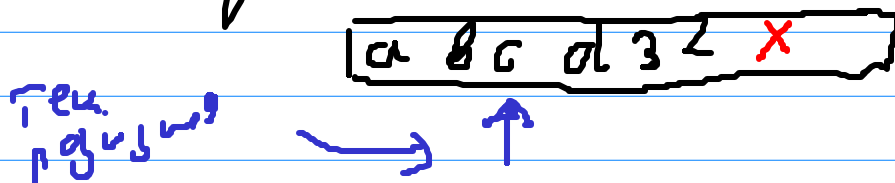
## Потоци в C++



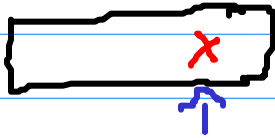
1

## Функции на състояния

- `bad()` - Някоя операция не  
успешно  
загуба на информация.
- `fail()` - Последната оп. е  
неуспешна

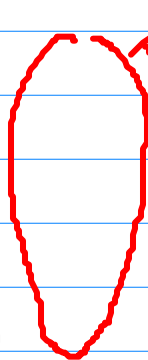




• eof() → текущата позиция е  
край на файла.



• good() - операция с указател  
• clear() - изчистване съст.  
на поток

fstream - поток за въход/изход

fstream (A.txt, ) →   
ifstream (A.txt, ) →   
ofstream (A.txt, ) →

↑   
от го-   
дана

↑   
режим за   
работа

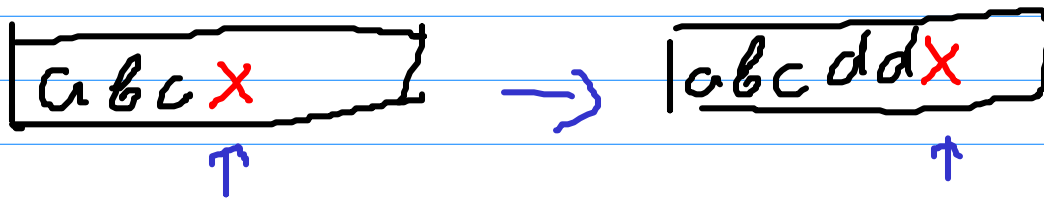
указател  
size\_t

• ios::in - поток за четене  
(по подразбиране за ifstream)

• ios::out - поток за писане  
(по подразбиране за ofstream)

• ios::binary - поток 3 а РАБОТА  
с двоични файл

изход {  
• ios::trunc - Ако файлът  $\exists$ ,  
то ще го изтрие.  
• ios::app - създа файл и  
поставя указателя  
(put указателя)  
в края



• ios::ate - поставя put указателя  
в края.

• Възможна го  
браниме указателя  
назад

ios::out  $\rightarrow 2$  0000.0010

ios::app  $\rightarrow 3$  0000.1000

→ 0 0 1 0 1 0 = 10  
 ↑ ↑ ↑  
 app dir in

Параметры  
и подков

Позиционирование в файле

istream f

0	1	2	3	4	5	6	7	8	9	10	11
1	2	3	5	8	7	1	2	4	X		

↑  
get

f.tellg() - Текущая позиция  
на get указателя

"местен" на get указателя

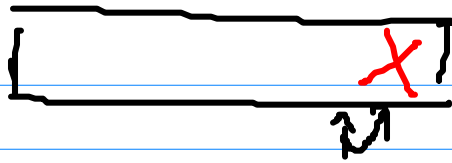
f.seek(offset, direction);

→ Отметка указателя

с offset позиции от direction

- От начала файла ios::~~base~~::beg
- От края файла ios::~~base~~::end

• ОТ ТРЕКЪЩАТА ПОЗНАНА : os::base::chr



~~за~~ Функция, която приема  
име на файл и  
върща големината у.

- Двоични опции



f

ofstream("file", ios::binary)

f.write { указател към почетен / адрес / позиция, докъдето да се запише }

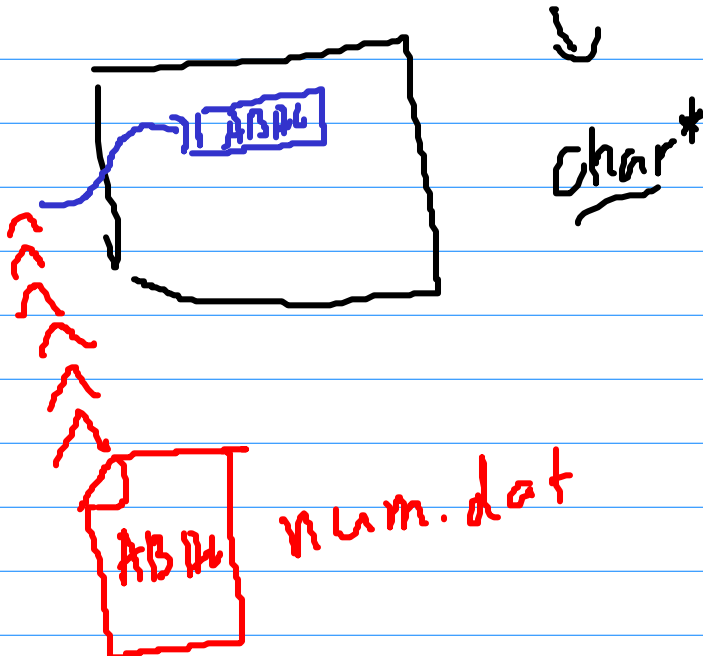
f.write(const char\*, 30);

int n = 10;

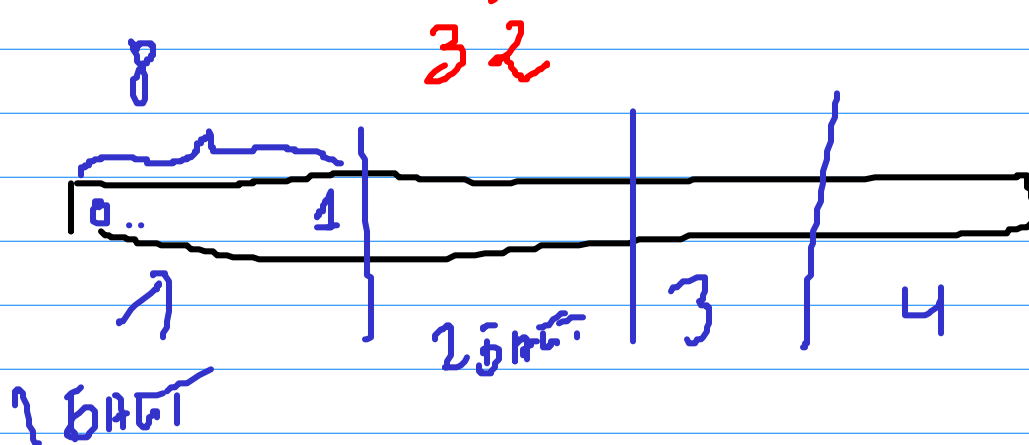
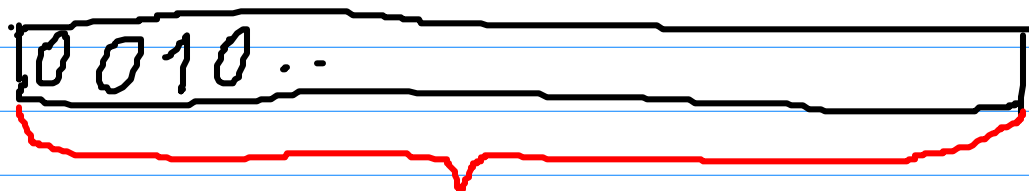
int\* p = &n

f.read (указатель  
к бул  
номер

концов файла  
и по byte



int 32 бит



бит 0, 1 с-т

байт  $2^8 = 256$

00100000

10000100

Битове и записване

в шестнадесетична система

decimal десет: 0, 1, 2, ..., 9 (10) |

binary дв: 0, 1 (2)

hex шестн 0, 1, ..., 9, A, B, C, D, E, F (16)

00

000

0

0

01

001

1

02

010

1

03

011

.

04

100

9

05

:

A

06

111

B

07

1000

C

08

:

D

09

:

E

10

:

F

11

15

10

12

16

10

13

17

10

14

18

10

15

19

10

:  
 2 9  
 3 0  
 9 9  
 1 0 0

1 1  
 :  
 1 F  
 2 0  
 :  
 2 F  
 3 0  
 :  
 FF  
 1 0 0

байтоте: 256 прзн. с-та  
 [0...255]  
 00 FF

12 34 5 → 

00	00	30	30
----	----	----	----

12 99 9 → 

00	00	32	67
----	----	----	----

~ 0<sup>8</sup> → 0<sup>8</sup> → 110010 11000111

(!) • write значења байтоте  
 в одржен рег-



НАД-геснѣт ѡѣт е зѣлѣцаѣ  
нѣрѣѣ!

67 32 0 0 6 0

→ int int int int int int