



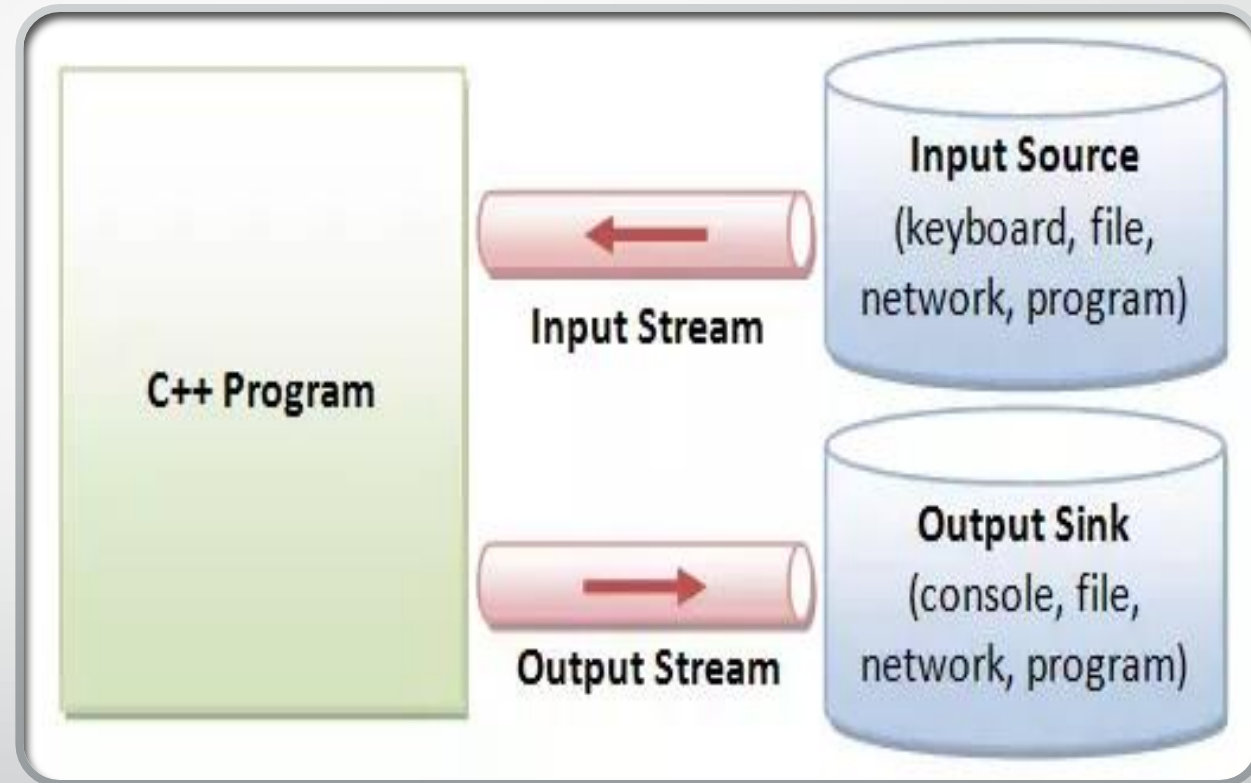
Работа с файлове



Потоци

Какво е поток

Потока е връзката, която осъществява предаването на информация между някакъв обект (клавиатура, монитор, файл, мрежа, принтер и т.н.) и нашата програма.



Видове потоци

Потоците могат да се разделят по начин по-който работят

- Потоци за вход (cin, ifstream, etc)
- Потоци за изход (cout, ofstream, etc)
- Потоци за вход и изход (fstream)

Защо потоци

Потоците освен, че ни позволяват връзката на нашата програма с периферни устройства, те позволяват и по-бързата обработка на информацията пристигаща или отиваща към устройствата. Освен нужните параметри, за да знае потока с какво е свързан и къде се намира, има и т.н. буфер, който представлява масив, в който се съхранява информацията от файла. Точно с този буфер работи програмата, а не директно с файла.

Пример с познатия cin

cin buffer

...

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

 ...



Потоци за файлове

Кои са потоците за файлове

- `ofstream` – поток за изход към файл/ запис към файл
- `ifstream` – поток за вход от файл/ четене от файл
- `fstream` – поток за вход и изход от/към файл

*Забележка – горе изброените потоци само задават по подразбиране потока да е в това състояние (да чете, да пише или и двете). Ние можем да нагласим потока както си искаме чрез флагове при отваряне на файла.

Закачане на файл към поток

Един файл може да се „закачи“ с даден поток в нашата програма по следните начини:

- При създаване на потока с неговия конструктор
- Със функцията `open`

```
std::ifstream inpt("test.txt");  
std::ofstream outp;  
  
outp.open("example.txt", std::ostream::in | std::ostream::trunc);
```

Път до файла + неговото име

Пътят може да бъде **относителен** спрямо текущия проект *или* да бъде **цял път** напр.:

C:\Programs\SDP\Solution\file.txt

Понеже обекта приема символен низ трябва да се **escape-нат** backslash-овете (\) т.е.:

C:\\Programs\\test.txt

Допълнителни флагове, с които
настройваме потока

Тези флагове може да се изреждат с | .
Повече за отделните флагове на
следващия слайд.

Флагове при създаване на поток

- `std::fstream::in` – Указва на потока, че може да чете от файла
- `std::fstream::out` – Указва на потока, че може да прави промени по файла
- `std::fstream::ate` – Задава позицията на записването в края на файла
- `std::fstream::app` – Задава позицията на записването да е само в края на файла, т.е. не можете да местите в по-късен етап къде да записвате
- `std::fstream::trunc` – Изчиства цялата информация от файла, която е съществувала преди отварянето на файла

Флагове указващи състоянието на потока

Всеки поток има флагове, които указват текущото състояние на потока. Те биват:

- **goodbit** – флаг указващ, че състоянието на потока е коректно. Той е в състояние true ако не е вдигнат друг флаг.
- **badbit** – флаг указващ, че има фатална грешка в потока (напр. прочитане след края на файла)
- **eof** – end of file, указва дали е стигнат края на файла
- **failbit** – указва, че в потока е настъпила нефатална грешка (напр. очаквали сме число а е дошъл символен низ)

Флагове указващи състоянието на потока

Нека потока ни се казва *fstr*. Стойностите на флаговете могат да се достъпват със съответните член функции:

- `fstr.goodbit()`, `fstr.badbit()`, `fstr.eof()`, ... etc.

Можем и да настройваме ръчно състоянията на флаговете чрез следните член функции:

- `fstr.clear()` – сваля всички флагове за грешка
- `fstr.clear(state)` – сваля всички флагове и активира този зададен в скобите
- `fstr.setstate(state)` – задава състоянието на подадения флаг

Извличане на информация от поток

`fstream` предоставя форматирано извличане от потока с оператора `>>` и неформатирано извличане с `getline`, като те работят еквивалентно на тези използвани при `cin`.

Извличането на информация се осъществява само когато няма вдигнати флагове за грешка.

Ако при форматирано извличане има фатално разминаване на типа данните, които програмата очаква, се вдига флага `failbit` и се прекратява тегленето от потока, докато не се предприемат мерки. Информацията в потока остава същата като тази преди разминаването.

Извличането на информация от потока започва от позицията на която е указателя за извличане. Неговата позиция може да се промени чрез `seekg()`, а за да разберете къде е в момента чрез `tellg()`.

Вмъкване на информация във файл

Може да се вмъква информация във файл с оператора <<. Действа еквивалентно както и при cout. Отново важат манипулаторите от iomanip библиотеката.

Вмъкването на информация започва от мястото където е поставен указателя за вмъкване и презаписва върху текущата информация на файла. Ако е стигнал края просто увеличава размера на файла и слага там подаденото.

Позицията на указателя за записване може да се промени чрез член функцията seekp(), а нейната позиция може да се определи чрез tellp().

Затваряне на файл и освобождаване на потока

Освобождаването на потока от файла се извърша с член функцията `close()` или ако вашия обект от `<fstream>` излезе извън областта му на живот. При затваряне на файл и освобождаване на потока всяка чакаща промяна (при записване във файла) направена в буфера се отразява във файла на вашия диск.

Някои основни съвети как да боравите с потоци

- Уверете се че правилно сте свързали потока с файл, то не си казва само ако има грешка, използвайте флаговете за проверка.
- Дори и един поток да е зададен за вход и изход не използвайте и двете действия едно след друго без да сте сигурни, че сте свършили работа с предишната операция коректно и сте проверили за възникнали грешки.
- След въвеждане на нова информация във файла, ако сте приключили с въвеждането, използвайте **flush()** или освободете потока от файла с **close()**. Не винаги вашата текуща промяна е запазена във файла, а само си седи в буфера и чака.
- При възникнали грешки преценете как да се справите с тях и дали са фатални за вашата работа (едно е да сте стигнали до края на файла друго е да не съществува такъв)

Използвани източници

- Документация на библиотеката - <http://www.cplusplus.com/reference/fstream/fstream/>
- Лекция за използване на файлови потоци - <http://umich.edu/~eecs381/handouts/filestreams.pdf>
- fstream not creating new file - <https://stackoverflow.com/questions/17260394/fstream-not-creating-new-file>
- Примери за използване на файлове - https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm