

Изготвил:
гл.ас. д-р. Нора Ангелова

Файлове

Файлове

- Редица от байтове с номерация, която започва от 0.

Работа с файлове

- Включване на библиотека за работа с файлове

```
#include <fstream>
```

- Декларация
- Отваряне
- Четене/Писане
- Затваряне

Декларация

- `ifstream iFileName;` - за извличане (четене)
- `ofstream oFileName;` - за вмъкване (писане)
- `fstream ioFileName;` - за извличане и вмъкване

Отваряне на файл

```
fileName.open("file", режим за достъп);
```

- `iFileName.open("file", ios::in);`
- `oFileName.open("file", ios::out);`
- `ioFileName.open("file",
ios::in|ios::out);`

Режим за достъп

ios::in (default for ifstream)	отваря файл за извличане
ios::out (default for ofstream)	отваряне на файл за вмъкване. Допуска се вмъкване на произволни места във файла. Ако файлът съществува, съдържанието се изтрива.
ios::app	отваря за вмъкване и установява указателя put в края на файла.
ios::ate	отваря за вмъкване и установява указателя put в края на файла. Допуска вмъкване на произволни места.
ios::trunc	ако файлът съществува, съдържанието се изтрива.
ios::binary	превключва режима от текстов в двоичен.

Режим за достъп

- **ios::nocreate** – отваря за вмъкване само ако файлът с указаното име съществува.
- **ios::noreplace** – отваря за вмъкване само ако файлът с указаното име не съществува.

Функции

- `open(<име_файл>, <режим_на_работа>{ |<друг_режим>});`

`ofstream <обект>;`

`<обект>.open(<име_файл>, <режим_на_работа>{ |<друг_режим>});`

ИЛИ

`ofstream <обект>(<име_файл>, <режим_на_работа>{ |<друг_режим>});`

Пример:

`ofstream file(<име_файл>, <режим_на_работа>{ |<друг_режим>});`

`// Проверка дали отварянето е успешно`

`if (!file) {`

`cerr << "File couldn't be opened!\n";`

`return 1;`

`}`

Open

- `ios::in|ios::ate !!!`

Комбинацията `ios::in | ios::ate` установява `put` и `get` указателите в края на файла.

Съществуват реализации, където опитите за извличане след подходящо позициониране на `get` указателя са успешни, но опитите за вмъкване са неуспешни.

- `ios::in|ios::app !!!`

Комбинацията `ios::in|ios::app` зависи от реализацията. При някои реализации не отваря файла. При други отваря файла и позиционира `put` и `get` в началото му. Опитите за извличане след подходящо позициониране на `get` указателя са успешни. Операциите за вмъкване са само в края на файла.

Функции

- **close()** – затваря файла прикрепен към потока.
Член-функция на класовете `fstream`, `ifstream`, `ofstream`.
- **istream& seekg**(streamoff p, ios::seekdir r) –
премества get указателя, с p байта относно режима на
позициониране r (ios::beg, ios::end, ios::cur)
- **ostream& seekp**(streamoff p, ios::seekdir r) –
премества put указателя, с p байта относно относно режима на
позициониране r (ios::beg, ios::end, ios::cur)

Функции

- **streampos tellg()** – връща текущата позиция на get указателя на файла.
- **streampos tellp()** – връща текущата позиция на put указателя на файла.

Пример:

```
file.seekg(0, ios::end);  
long loc = file.tellg();
```

Файлове

- Файлове с последователен достъп
- Файлове с пряк достъп (бинарни)

Файлове с последователен достъп

Компонентите на тези файлове са редица от СИМВОЛИ завършващи с '\n'.

За да бъде достигнат и прочен елементът с пореден номер n , трябва последователно да бъдат прочетени всички предшестващи го елементи.

```
abcd 1 da 9 '\n'  
..... '\n'  
..... '\n'  
..... '\n'  
..... '\n'
```

Файлове с пряк достъп

Търсеният елемент се достига направо (по адреса му), без да е необходимо да се прочетат предшестващите го елементи.

Данните, които записваме не се третират като символи.

Пример:

Използват се когато искаме да запишем обекти от даден клас.

обект1 памет

обект2 памет

обект3 памет

...

обектх памет

Двоични файлове

- Неформатиран вход и изход
- Позволяват пряк достъп

Край