Praca na plikach cz.2

Musisz poznać jeszcze tylko kilka sztuczek i będziesz wiedział już wszystko, co trzeba wiedzieć o technikach pracy z plikami.

Wczesne zamykanie pliku

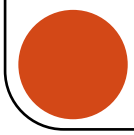
Na początku tego rozdziału napisałam, jak otworzyć plik, ale jeszcze do tej pory nie wyjaśniłem, jak go potem zamknąć. Nie zapomniałem o tym, po prostu robienie tego nie jest konieczne. Otwarte pliki są automatycznie zamykane, gdy program kończy wykonywanie bloku, w którym zostały utworzone ich strumienie.

```
void f()
{
    ofstream strumien("C:/Nanoc/data.txt");    // Plik jest otwarty

    // Operacje na pliku

}  // Koniec bloku, a więc w tym miejscu plik zostanie automatycznie zamknięty
```

Nie trzeba nic robić, aby zamknąć plik. I całe szczęście, bo dzięki temu nigdy o tym nie zapomnimy.

Czasami jednak trzeba zamknąć plik wcześniej, niż powinno to nastąpić automatycznie. W takim przypadku należy użyć funkcji `close()`.

```
void f()
{
    ofstream strumien("C:/Nanoc/data.txt");    // Plik jest otwarty

    // Operacje na pliku

    strumien.close(); // Zamknięcie pliku
                      // Od tego miejsca nie można dokonywać zapisu w pliku
}
```



Można też zadeklarować strumień, ale otwarcie pliku odłożyć na później. Służy do tego funkcja `open()`.

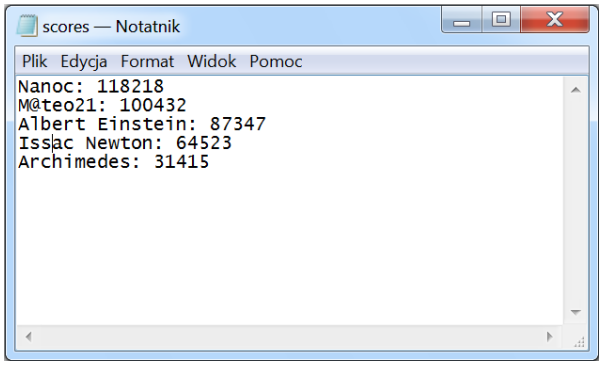
```
void f()
{
    ofstream strumien;    // Strumień bez przypisanego pliku

    strumien.open("C:/Nanoc/data.txt");  // Otwarcie pliku C:/Nanoc/data.txt

    // Operacje na pliku

    flux.close(); // Zamknięcie pliku
                  // Od tego miejsca nie można dokonywać zapisu w pliku
}
```

Pragnę podkreślić, że wykonywanie powyższych czynności jest rzadko konieczne. Zazwyczaj wystarczy bezpośrednie otwarcie pliku i pozostawienie go do automatycznego zamknięcia.

> Niektórzy programiści lubią używać funkcji `open()` i `close()`, mimo że jest to niepotrzebne. Dzięki temu łatwiej jest się zorientować, w którym miejscu plik jest otwierany, a w którym zamykany, ale tak naprawdę to kwestia gustu.

Kursor w pliku

Na chwilę zagłębimy się trochę bardziej w szczegóły techniczne, aby dokładnie poznać mechanizm odczytywania zawartości plików. Gdy otworzymy plik w edytorze tekstu, np. Notatniku, znajdziemy w nim kursor wskazujący miejsce, w którym aktualnie się znajdujemy. Na poniższym rysunku kursor znajduje się za drugim „s" w słowie „Issac" w czwartym wierszu.

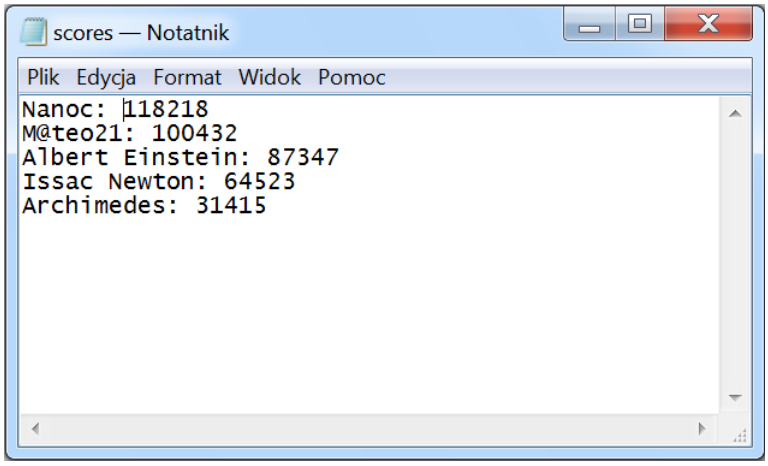


Gdy naciśniesz jakiś klawisz na klawiaturze, w miejscu, w którym znajduje się teraz kursor pojawi się nowa litera. Analogicznie w języku C++ istnieje odpowiednik kursora, za pomocą którego można wybrać miejsce do wpisywania danych. Spójrz na poniższy kod:

```
ifstream plik("C:/Nanoc/scores.txt")
```

Instrukcja ta spowoduje otwarcie pliku C:/Nanoc/scores.txt i umieszczenie kursora na samym jego początku. Gdy wczytamy pierwsze słowo z tego pliku, otrzymamy łańcuch znaków `Nanoc:`. Po tej

operacji „kursor C++” przesunie się na początek następnego słowa, w miejsce pokazane na poniższym rysunku:



Następnym wczytanym słowem będzie `118218` itd. Oznacza to, że w ten sposób zawartość pliku można przeglądać tylko **sekwencyjnie.** To nie jest zbyt praktyczne.

Na szczęście istnieją sposoby na zmienianie pozycji kursora w pliku. Można na przykład przesunąć go do 20 miejsca od początku pliku albo o 32 znaki do przodu względem bieżącego położenia. Dzięki temu można odczytać dokładnie te fragmenty pliku, które nas interesują.

Aby to jednak zrobić, najpierw trzeba dowiedzieć się, w którym miejscu pliku aktualnie się znajdujemy. Dopiero potem możemy przesunąć kursor. Poniżej dowiesz się, jak to zrobić.

Sprawdzanie pozycji kursora w pliku

Istnieją specjalne funkcje do sprawdzania, który bajt w pliku wskazuje aktualnie kursor. Inaczej mówiąc funkcje te pozwalają sprawdzić, na którym znaku w danej chwili jest kursor. Piszę funkcje, nie funkcja, ponieważ są ich dwie: po jednej dla strumienia wejściowego i wyjściowego. Niestety ich nazwy brzmią dość dziwnie: `tellg()` (dla strumienia `ifstream`) i `tellp()` (dla strumienia `ofstream`). Na szczęście używa się ich tak samo.

```
ofstream plik("C:/Nanoc/data.txt");

int pozycja = plik.tellp(); // Sprawdzamy pozycję

cout << "Jesteśmy na znaku nr " << pozycja << " w pliku." << endl;
```

Przesuwanie kursora

Do przesuwania kursora również służą dwie funkcje: `seekg()` (dla strumienia `ifstream`) i `seekp()` (dla strumienia `ofstream`).

Tych funkcji również używa się tak samo, a więc poniżej przedstawiam przykład użycia tylko jednej z nich:

```
strumien.seekp(liczbaZnakow, pozycja);
```

Parametr *pozycja* może przyjmować jedną z trzech wartości:

- `ios::beg` — początek pliku
- `ios::end` — koniec pliku
- `ios::cur` — bieżąca pozycja

Aby na przykład ustawić się dziesięć znaków za początkiem pliku, należy napisać instrukcję `strumien.seekp(10, ios::beg);`. A żeby przesunąć kursor o 20 znaków dalej od miejsca, w którym się on aktualnie znajduje, należy napisać `strumien.seekp(20, ios::cur);`. Chyba proste, prawda?

Sprawdzanie długości pliku

Aby sprawdzić długość pliku, należy wykorzystać wiedzę zdobytą w dwóch poprzednich podrozdziałach, tzn. najpierw trzeba przenieść kursor na koniec pliku, a następnie „spytać” strumień, gdzie jest kursor:

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ifstream plik("C:/Nanoc/najlepszeWyniki.txt"); // Otwarcie pliku
    plik.seekg(0, ios::end); // Przejście na koniec pliku

    int dlugosc;
    dlugosc = plik.tellg();  // Sprawdzamy pozycję, która odpowiada długości
pliku!

    cout << "Długość pliku w bajtach wynosi: " << dlugosc << "." << endl;

    return 0;
}
```

To wszystko, jeśli chodzi o podstawowe wiadomości. Możesz zacząć zgłębiać dalsze tajniki świata plików i języków programowania.