

نحوه کار با فایل در C++

در این نوشتار قصد داریم به طور خلاصه با کار کردن با فایل‌ها آشنا شویم. کار با فایل‌ها در C++ تا حد زیادی شبیه به کار با ورودی و خروجی استاندارد (کنسول) است. برای کار با فایل، باید `fstream` را در برنامه خود `include` کنید.

اولین برنامه: نوشتن در یک فایل

اولین برنامه‌ای که در این قسمت می‌نویسیم، برنامه زیر است که آن را توضیح می‌دهیم:

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

int main()
{
    ofstream fout("file.txt"); ❶❷❸
    fout << "Welcome to the real world." << endl; ❹
    fout.close(); ❺
    return 0;
}
```

- ❶ `ofstream` مخفف `output file stream` است و از آن برای نوشتن خروجی در فایل استفاده می‌شود.
- ❷ `fout` نام متغیری است که از آن برای کار با فایل استفاده می‌شود. در واقع `fout` یک متغیر از نوع `ofstream` است. نام این متغیر هر چیزی می‌تواند باشد! `myFileStream` یا هر اسم مجاز دیگر.
- ❸ `file.txt` نام فایلی است که قرار است خروجی در آن نوشته شود. این فایل در کنار فایل `cpp` شما باید قرار داشته باشد و اگر وجود نداشته باشد، به صورت خودکار ایجاد می‌گردد. بنابراین نگران نباشید!
- ❹ همان‌گونه که `cout << "salam"` عبارت `salam` را در خروجی چاپ می‌کرد، `fout << "salam"` اکنون همین عبارت را فقط در یک فایل می‌نویسد! یعنی دقیقاً به همان صورتی که در کنسول خروجی می‌دهیم، می‌توانیم در فایل هم بنویسیم!
- ❺ در انتهای کار با فایل باید آن را `close` کرد. این کار هم به وسیله دستور `fout.close()` انجام می‌شود! بعد از بستن فایل دیگر نمی‌توان به آن دسترسی داشت و برای دسترسی مجدد باید آن را دوباره `open` کرد.

خوب! الان با اجرای این برنامه، یک فایل به نام `file.txt` در کنار برنامه‌ی شما ایجاد می‌شود.

اگر از ویژوال استودیو استفاده می‌کنید، احتمالاً باید در یکی از فولدرهای debug یا release در کنار فایل اجرایی برنامه دنبال این فایل بگردید!

دومین برنامه: خواندن از فایل

به برنامه زیر توجه کنید:

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

int main()
{
    ifstream fin("file.txt"); ❶❷
    char ch;

    while(!fin.eof()) ❸
    {
        fin.get(ch); ❹
        cout << ch; ❹
    }

    fin.close(); ❺

    return 0;
}
```

۱. ifstream به معنی input file stream است و برای خواندن فایل از آن استفاده می‌شود! fin هم نام متغیری است از نوع ifstream که از آن برای خواندن از فایل استفاده می‌کنیم. نام فایل هم file.txt است.
۲. دقت کنید که ifstream کنترل نمی‌کند که فایل با نام file.txt وجود داشته باشد یا نه (از فایلی که وجود ندارد که نمی‌توانیم بخوانیم!). این قضیه را باید خودمان کنترل کنیم که در ادامه به آن اشاره می‌کنیم.
۳. fin.eof() کنترل می‌کند که آیا در جریان خواندن فایل به انتهای آن رسیده‌ایم یا نه! EOF مخفف End of File است. پس شرط while یعنی تا زمانی که به انتهای فایل نرسیده‌ایم!
۴. fin.get(ch) یک کاراکتر از فایل می‌خواند و در متغیر ch قرار می‌دهد. cout << ch هم آن را چاپ می‌کند.
۵. در آخر هم باید فایل را ببندیم.

به این ترتیب، توانستیم فایل را بخوانیم و آن را در خروجی چاپ کنیم! (اگر این برنامه را زیر برنامه قبلی بنویسید، چیزی که در آن فایل نوشته بودید در خروجی نمایان می‌شود).

دستور get در برنامه بالا، کاراکتر به کاراکتر از فایل می‌خواند و در خروجی می‌نویسد. این نوع خواندن، گاهی اوقات مطلوب ما نیست و برای ما ساده‌تر این است که مثلاً یک خط از فایل بخوانیم. مثلاً فرض کنید محتویات فایل ما به صورت زیر باشد:

```
1500 2600 45 141
sadjad
```

حال کد زیر بر روی این فایل اجرا می‌کنیم:

```
int a, b, c, d;
string s;
ifstream fin("file.txt");

fin >> a >> b;
fin >> c;
fin >> d;
fin >> s;
```

بعد از اجرای این کد مقادیر ۱۵۰۰، ۲۶۰۰، ۴۵ و ۱۴۱ در a، b، c و d قرار می‌گیرند و "sadjad" در داخل رشته s خواهد بود. در واقع fin مشابه cin عمل کرده است و همانطور که cin ورودی را از کنسول می‌خواند، fin از فایل ما ورودی را خوانده است.

همچنین می‌توانیم ورودی را به صورت خط به خط (به جای کاراکتر به کاراکتر) بخوانیم. قطعه کد زیر این مطلب را نشان می‌دهد:

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

int main()
{
    ifstream fin("file.txt");
    string line;

    while(!fin.eof())
    {
        getline(fin, line);
        cout << line;
    }

    fin.close();

    return 0;
}
```

دستور getline که متعلق به string است به این ترتیب عمل می‌کند که یک خط را از فایل می‌خواند و در یک رشته (در اینجا مثلاً line) می‌ریزد. به این ترتیب می‌توانیم یک فایل را خط به خط بخوانیم!

کنترل وجود فایل

همان‌طور که گفتیم، خواندن از فایلی که وجود ندارد، معنی ندارد! برای اینکه کنترل کنیم فایل با نامی که می‌خواهیم وجود دارد می‌توانیم بنویسیم:

```
ifstream fin("file.txt");  
  
if(!fin)  
    cout << "File doesn't exist!" << endl;
```

باز کردن فایل

به جای نوشتن

```
ifstream fin("file.txt");
```

می‌توانیم بنویسیم:

```
ifstream fin;  
fin.open("file.txt");
```

دستور open در واقع فایل موردنظر ما را برای خواندن باز می‌کند.

توضیحات بیشتر در مورد نوشتن در فایل

اگر فایلی که قصد نوشتن در آن را داریم وجود نداشته باشد، به طور خودکار ایجاد می‌شود. اما اگر این فایل وجود داشته باشد، محتویات آن از بین می‌رود! در واقع، هنگام نوشتن، محتویات جدید به طور کامل جایگزین محتویات قبلی می‌شود. برای اینکه در صورت وجود فایل، مطالبی که می‌خواهیم بنویسیم در انتهای فایل و در ادامه اطلاعات قبلی آن بیایند باید بنویسیم:

```
ofstream fout("file.txt", ios::app);
```

ios::app باعث می‌شود تا در صورت وجود فایل، اطلاعات جدید در انتهای آن Append شوند.

مطالعه بیشتر

1. <http://www.cplusplus.com/reference/>
2. <http://www.cpp-home.com/archives/67.htm>

موفق باشید!