KỸ THUẬT LẬP TRÌNH C/C++

Chương 9:
ĐỌC VÀ GHI FILE TRONG C++

1. Đọc và ghi file trong C++

File được dùng để lưu dữ liệu cố định trong một thiết bị lưu trữ trên máy tính. Có rất nhiều kiểu file khác nhau nhằm lưu dữ các kiểu dữ liệu và thông tin khác nhau, ví dụ như file text, file Excel, file Json, file XML, file CSV, …

Trong C++ cung cấp một thư viện chứa các phương thức để xử lý file. Khi xử lý file trong chương trình C++ chúng ta có thể đọc dữ liệu trong file và ghi dữ liệu vào file. Để thực hiện xử lý này thì trong C++ hỗ trợ thư viện fstream gồm 3 lớp:

- ifstream (input file stream): được dùng để đọc dữ liệu từ file đã tồn tại.
- ofstream (output file stream): được dùng để tạo file mới và ghi dữ liệu vào file.
- fstream: (file stream): bao gồm cả tính năng của ofstream và ifstream, nó vừa có khả tạo file, ghi dữ liệu vào file và đọc dữ liệu từ file.

1. Đọc và ghi file trong C++

- Đọc ghi file thường có các chế độ (mode) định dạng đi kèm như sau:

ios::in : mở một file để đọc (mode mặc định của ifstream).

ios::out : mở một file để ghi (mode mặc định của ofstream).

ios::binary : dùng để mở file nhị phân.

ios::ate : dùng để đặt vị trí con trỏ ở cuối file khi ta mở file.

ios::app : nếu file đã tồn tại, thêm dữ liệu vào cuối file (append) .

ios::trunc : nếu file đã tồn tại, nội dung của nó sẽ được cắt bỏ (truncate) trước khi mở file.

- Để thực hiện tiến trình xử lý đọc ghi file trong C++ thì cần phải khai báo hai thư viện ở đầu chương trình: <iostream> và <fstream>

```
#include <iostream>
#include <fstream>
```

1. Đọc và ghi file trong C++

1. Mở một file trong C++

Một file phải được mở trước khi có thể đọc dữ liệu từ nó hoặc ghi dữ liệu vào nó. Đối tượng ofstream hoặc đối tượng fstream có thể được sử dụng để mở một file với mục đích ghi file hoặc đối tượng ifstream được sử dụng để mở file chỉ với mục đích đọc file.

Để mở một file chúng ta sử dụng hàm open(), đây là một thành viên của các đối tượng fstream, ifstream và ofstream trong C++.

Cú pháp của hàm open() như sau:

<tên đối tượng>.open("tên file");

Ví dụ: Mở file myinfo.txt với các mode mặc định.

```
ofstream  writefile;
writefile.open("myinfo.txt");
```

1. Đọc và ghi file trong C++

1. Mở một file trong C++ (tt)

Bên cạnh đó, chúng ta có thể kết hợp hai hoặc nhiều mode, và sử dụng dấu " | " để phân cách giữa các mode. Cú pháp như sau:

<tên đối tượng>.open("tên file" , ios::mode | ios::mode);

Ví dụ: Mở myinfo.txt file trong chế độ cho phép đọc và ghi file (trường hợp file này đã tồn tại):

```
ofstream rwfile;
rwfile.open("myinfo.txt", ios::out | ios::in);
```

- Hoặc:

```
fstream rwfile;
rwfile.open("myinfo.txt", ios::out | ios::in );
```

1. Đọc và ghi file trong C++

1. Mở một file trong C++ (tt)

Để kiểm tra file được mở có tồn tại hay không thì chúng ta sử dụng phương thức is_open(), phương thức này trả về true nếu file đang tồn tại, ngược lại thì trả về false.

Ví dụ: Mở myinfo.txt file trong chế độ cho phép đọc và ghi file, nếu file này đang tồn tại thì hiển thị thông báo "Mo file thanh cong", ngược lại thì hiển thị thông báo "Mo file that bai".

```
...
ofstream rwfile;
rwfile.open("myinfo.txt", ios::out | ios::in);
if (rwfile.is_open())
    cout << "Mo file thanh cong" << endl;
else
    cout << "Mo file that bai" << endl;
```

1. Đọc và ghi file trong C++

2. Đóng file trong C++

Khi một chương trình C++ kết thúc, nó tự động đóng tất cả Stream, giải phóng tất cả bộ nhớ đã cấp phát và đóng tất cả các file đã mở. Tuy nhiên cần phải đóng tất cả các file đã mở trước khi kết thúc chương trình. Để làm việc này chúng ta sử dụng hàm close(), hàm này là một thành viên của các đối tượng fstream, ifstream và ofstream.

Ví dụ: Sau khi ghi dữ liệu vào file myinfo.txt xong thì đóng file này lại.

- Mở file myinfo.txt ở mode ghi file:

```
ofstream writefile;
writefile.open("myinfo.txt", ios::out);
```

- Đóng file đã mở:

```
writefile.close();
```

1. Đọc và ghi file trong C++

3. Ghi file trong C++

Trong C++ để ghi dữ liệu vào một file chúng ta sử dụng toán tử chèn luồng (<<) kết hợp với tên đối tượng ofstream hoặc fstream.

Ví dụ: Ghi chèn thêm dữ liệu từ biến data vào trong file myinfo.txt đã tồn tại.

- Mở file myinfo.txt trong chế độ ghi dữ liệu vào cuối file:

```
ofstream writefile;
writefile.open("myinfo.txt", ios::app); //Mở file trong chế độ ios::app
```

- Ghi dữ liệu từ biến data vào file myinfo.txt:

```
writefile << data << endl;
```

Lưu ý: Trong chương trình cần phải khai báo trước biến data để lưu dữ liệu như sau:

```
string data;  // Khai báo biến data có kiểu dữ liệu là string.
```

1. Đọc và ghi file trong C++

3. Ghi file trong C++ (tt)

Ví dụ: Ghi thông tin sinh viên gồm: Họ tên, năm sinh, MSSV vào file.

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

int main ()
{
   string data;
   ofstream writefile;
   writefile.open("myinfo.txt", ios::out);
   cout << "Ghi du lieu vao file!" << endl;

   cout << "Nhap ten sinh vien: ";
   getline(cin, data);
   writefile << data << endl;

   cout << "Nhap nam sinh: ";
   cin >> data;
   writefile << data << endl;

    cout << "Nhap MSSV: ";
    cin >> data;
    writefile << data << endl;

    writefile.close();

   return 0;
}
```

1. Đọc và ghi file trong C++

4. Đọc một File trong C++

Trong C++ để đọc dữ liệu từ file bằng cách sử dụng toán tử trích luồng (>>) kết hợp với tên đối tượng ifstream.

Ví dụ: Đọc dữ liệu từ trong file myinfo.txt.

- Mở file myinfo.txt trong chế độ đọc file:

```
ifstream readfile;
readfile.open("myinfo.txt", );
```

- Đọc và in một dòng dữ liệu trong file ra màn hình:

```
getline(readfile, data); //đọc 1 dòng dữ liệu trong file vào biến data
cout << data << endl; //in dòng dữ liệu trong biến data ra màn hình
```

Lưu ý: Trong chương trình cần phải khai báo biến data để chứa dữ liệu đọc ra từ file:

```
string data;  // Khai báo biến data có kiểu dữ liệu là string.
```

1. Đọc và ghi file trong C++

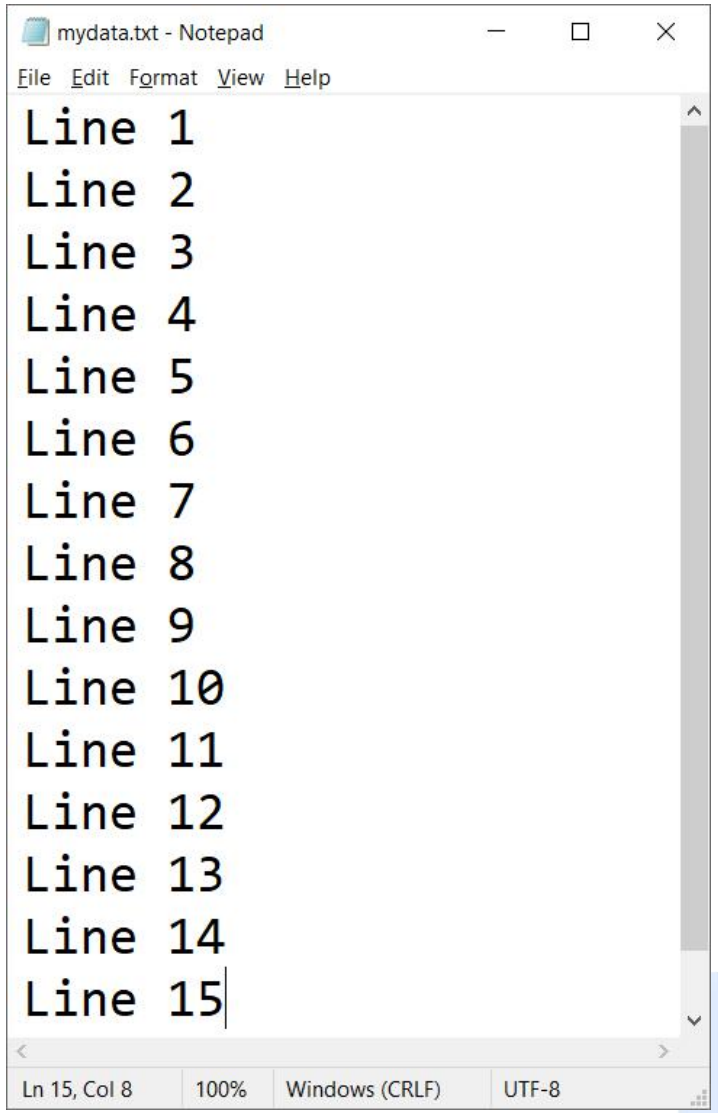
4. Đọc một File trong C++ (tt)

Trường hợp đọc và in ra nhiều dòng dữ liệu trong file.

Ví dụ: File mydata.txt có chứa 15 dòng dữ liệu, viết chương trình đọc 10 dòng dữ liệu trong file này và in ra màn hình.

Gợi ý: sử dụng vòng lặp for như sau:

```
for(int i=0; i <=9 ; i++){
    getline(readfile, data);
    cout << data << endl;
}
```



1. Đọc và ghi file trong C++

4. Đọc một File trong C++ (tt)

Trường hợp muốn đọc và in ra tất cả các dòng trong file, thì chúng ta cần phải kiểm tra xem con trỏ đã trỏ đến cuối file hay chưa, nếu chưa thì tiếp tục đọc, ngược lại thì kết thúc. Để làm được điều này, chúng ta sử dụng phương thức eof(), phương thức này trả về true khi con trỏ đã trỏ tới cuối file và trả về false khi con trỏ chưa ở cuối file.

Ví dụ: Đọc và in ra tất cả các dòng dữ liệu trong file mydata.txt.

Gởi ý: sử dụng vòng lặp while như sau:

```
while(!readfile.eof())  //khi chưa tới cuối file thì tiếp tục đọc dòng dữ liệu tiếp theo.
{
    getline (readfile, data); //đọc một dòng dữ liệu trong file
    cout << data << endl; //in dòng dữ liệu đã đọc ra màn hình
 }
```

1. Đọc và ghi file trong C++

4. Đọc một File trong C++ (tt)

Ví dụ:

```
#include <fstream>
#include <iostream>
#include <string.h>

using namespace std;

int main ()
{
   string data;

   ifstream readfile;
   readfile.open("myinfo.txt", ios::in );

   cout << "Doc du lieu co trong file!" << endl;

  //Đọc tất cả các dòng nội dung trong file
    while(!readfile.eof())
    {
        getline (readfile, data);
        cout << data << endl;
    }

   readfile.close();

   return 0;
}
```

2. Con trỏ vị trí file

Để xác định vị trí của con trỏ vị trí file (file-position pointer) chúng ta sử dụng hàm seekg().

Cú pháp:

seekg(seekbyte, ios_base::seekdir);

Trong đó:

seekbyte: số byte cần dịch (luôn là một số nguyên, mỗi byte là một kí tự, khoảng trắng xem là một ký tự). Nếu là số dương thì dịch từ trái sang phải, còn nếu là số âm thì sẽ dịch từ phải sang trái.

seekdir: vị trí bắt đầu dịch, có 3 vị trí bắt đầu gồm:

+ beg (mặc định): bắt đầu ở vị trí đầu file.
+ cur: bắt đầu ở vị trí hiện tại.
+ end: bắt đầu ở vị trí cuối file.

Lưu ý: Dữ liệu được lấy ngay tại con trỏ vị trí file đến khi gặp khoảng trắng thì dừng lại.

2. Con trỏ vị trí file

Ví dụ: Cho file mydata.txt có chứa dữ liệu gồm các số nguyên được cách nhau khoảng trắng:

18 27 36 45 54. Viết chương trình thực hiện yêu cầu như sau:

- Di chuyển con trỏ vị trí file qua phải 3 byte bắt đầu từ đầu file, lấy dữ liệu và in ra màn hình.
- Di chuyển con trỏ vị trí file qua phải 4 byte bắt đầu từ vị trí hiện tại, lấy dữ liệu và in ra màn hình.
- Di chuyển con trỏ vị trí file qua trái 2 byte bắt đầu từ cuối file, lấy dữ liệu và in ra màn hình.

Kết quả chạy chương trình ?