Tập tin

BUỔI 15

Làm việc với tập tin

1. Mục tiêu

- Hiểu được cơ bản các nguyên lý cơ bản làm việc với tập tin.
- Áp dụng các kỹ năng lập trình cơ bản với tập tin trên C++.
- Mở rộng kỹ thuật thao tác với tập tin trên C++.

2. Các thuật ngữ

- Input
- Output
- Stream
- Path
- File
- Folder

3. Nội dung


Tổng quan thao tác với tập tin
Mở và đóng tập tin
Đọc và ghi tập tin

3.1 Tổng quan thao tác với tập tin

- **Tại sao phải sử dụng tập tin ?**
  - Thông thường: nhập dữ liệu – biến … từ bàn phím ➔ thao tác ➔ xuất ra màn hình. Dữ liệu được lưu trữ trên RAM (bộ nhớ lưu trữ tạm thời).
  - Ưu điểm: xử lý trên RAM có tốc độ cao do tốc độ truyền dữ liệu cao.
  - Khuyết điểm: RAM giá thành đắt – không lưu trữ dài hạn dữ liệu (mất điện sẽ mất dữ liệu…) ➔ không xử lý được bài toán có dữ liệu lớn (Big Data), không lưu trữ các kết quả để lần sau sử dụng (sau khi tắt chương trình…)
  - Khắc phục: dữ liệu được lưu trữ trên ổ cứng (HDD, SSD…) để có thể xử lý dữ liệu lớn và tái sử dụng dữ liệu. Dữ liệu được tổ chức thành các tập tin để lưu trữ trên ổ cứng.
- **Khái niệm về tập tin**
  - Tập hợp thông tin (dữ liệu) được tổ chức theo một dạng xác định với tên được định danh
  - Một dãy byte liên tục (dưới góc độ lưu trữ)
  - Được lưu trữ trong thiết bị lưu trữ ngoài: USB, HDD, SSD…
  - Cho phép đọc dữ liệu (thiết bị nhập) và ghi dữ liệu (thiết bị xuất).

3.1 Tổng quan thao tác với tập tin

- **Phân loại**
  - Mục đích sử dụng: quan tâm đến nội dung tập tin sẽ phân loại theo phần mở rộng tập tin (đuôi tập tin): .EXE, .DOCX, .TXT, .PPT …
  - Mục đích lập trình: tự tạo các stream tường minh để kết nối với tập tin xác định nên sẽ phân loại theo cách sử dụng stream.
  - 2 dạng tập tin cơ bản: tập tin dạng văn bản (tương ứng với stream văn bản) và tập tin dạng nhị phân (tương ứng với stream nhị phân).
  - Tập tin văn bản sẽ được giới thiệu chính, tập tin nhị phân sinh viên tự tìm hiểu thêm
- **Stream văn bản**
  - Chỉ chứa các ký tự
  - Tổ chức thành từng dòng, kết thúc bởi ký tự kết thúc dòng \0 hoặc ký tự sang dòng mới \n.
- **Stream nhị phân**
  - Chứa các byte
  - Đọc và ghi dữ liệu chính xác từng byte

3.1 Tổng quan thao tác với tập tin

- **Quy tắc đặt tên**

```
<Tên tập tin> . <Mở rộng tập tin>
```

- **Tên tập tin**
  - Bắt buộc phải có
  - Chiều dài tối đa 128 ký tự
  - Gồm các ký tự từ A đến Z, a đến z, số 0 đến , khoảng trắng, các ký tự @#$%^()!

- **Mở rộng tập tin**
  - Không bắt buộc
  - Thông thường 3 – 4 ký tự (chữ và số)

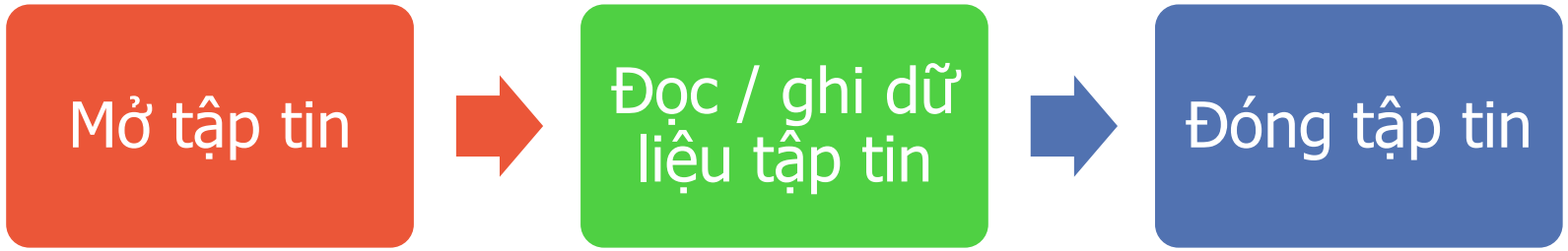
3.1 Tổng quan thao tác với tập tin

- **Đường dẫn**
  - Địa chỉ chỉ đến một tập tin hiện hành trên ổ cứng.
  - Ví dụ: c:\data\list.txt chỉ tập tin list.txt nằm trong ổ cứng C có thư mục con là data
  - Trong chương trình C++, đường dẫn trên được ghi dưới dạng như sau "c:\\data\\list.txt"

- **Vì sao phải viết đường dẫn có thêm \\ như trên ?**
  - Vì dấu '\' là một ký tự biểu diễn nên để biểu diễn thì phải thêm một dấu '\' thành "\\" ở trước để ký hiệu.
  - Nếu nhập đường dẫn từ bàn phím thì không cần thêm dấu '\'.

3.1 Tổng quan thao tác với tập tin

- Thao tác chính với tập tin: đọc tập tin và ghi tập tin
- Thao tác cơ bản:



- Tương ứng nhập / xuất từ màn hình, đọc / ghi tập tin cũng tương tự

cin >>	Input	ifstream >>
cout <<	Output	ofstream <<
#include "iostream"	Library	#include "fstream"

3.2 Mở và đóng tập tin

- Mở tập tin cho đọc dữ liệu từ tập tin

  ifstream tên_biến(đường dẫn tập tin)

- Kiểm tra tập tin đang được sử dụng hay có tồn tại hay không

  If(A != NULL)…

- Đóng tập tin

  A.close()

- Ví dụ: mở tập tin "test" – kiểm tra tập tin có sử dụng được không – đóng tập tin

```
ifstream out("c:\\test.txt");
if(!out){
        cout << "Khong mo duoc file." << endl;
        return 1;
}
...
out.close();  /// Dong tap tin
```

3.2 Mở và đóng tập tin

- Mở tập tin cho đọc dữ liệu

```
ifstream tên_biến(đường dẫn tập tin)
```

- Mở tập tin cho ghi dữ liệu vào cuối tập tin (ghi tích hợp cuối tập tin)

```
ofstream tên_biến(đường dẫn tập tin, ios::app)
```

- Kiểm tra tập tin và đóng tập tin tương tự phần trước

3.3 Ghi và đọc tập tin

- Ghi dữ liệu vào tập tin sử dụng ofstream << tương tự xuất ra màn hình (cout <<)

```
ofstream A(tên tập tin)
A << [dữ liệu]
```

- Ghi dữ liệu tích hợp vào cuối tập tin

```
ofstream A(tên tập tin, ios::app)
A << [dữ liệu]
```

- Ví dụ để ghi một tập tin có tên là "test" ở thư mục cùng file chạy .exe với nội dung như sau

10 123.23

HelloCplusplus.

3.3 Đọc và ghi tập tin

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
	ofstream out(“C:\\test.txt"); /// Mo tap tin de ghi
	if(!out)    {/// Kiem tra tap tin dang su dung hay chua
		cout << "Khong mo duoc file." << endl;
		return 1;
	}
	out << 10 << "\t" << 123.23 << endl; /// Ghi du lieu vao tap tin
	out << "Hello C plus plus.";
	out.close();  /// Dong tap tin
	return 0;
}
```

3.3 Đọc và ghi tập tin

- Ghi dạng tích hợp vào cuối tập tin (*append*). Mở tập tin "test" ghi vào cuối tập tin

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
        ofstream out("test", ios::app); /// Mo tap tin test de ghi tich hop [ios::app]
        if(!out)        {
                cout << "Khong mo duoc file.\n";
                return 1;
        }
        out << "Append" << endl;
        out.close();
        return 0;
}
```

3.3 Đọc và ghi tập tin

- Đọc tập tin sử dụng ifstream và toán tử >>. Tương tự như dùng cin >>

```
ifstream A(tên tập tin)
A >> [dữ liệu]
```

- Ví dụ: đọc dữ liệu từ tập tin "test"

```
char ch;      int i; float f;      char str[80];
ifstream in("C:\\test.txt");
if(!in){
        cout << "Khong mo duoc file.\n";
        return 1;
}
/// Gan du lieu vao bien
in >> i;
 in >> f;
 in >> ch;
 in >> str;
/// Xuat cac bien ra man hinh
cout << i << " " << f << " " << ch << endl;
cout << str << endl;
in.close();
```

3.3 Đọc và ghi tập tin

- Hàm kiểm tra cuối tập tin: hàm Boolean **eof()**
- Đọc tập tin bằng cách duyệt từng ký tự. Sử dụng hàm get(…)
- Ví dụ: đọc tập tin "test" bằng cách duyệt theo từng ký tự

```
char ch;
ifstream in("test");
if(!in){
        cout << "Khong mo duoc file.\n";
        return 1;
}
while(!in.eof()){ /// Kiem tra cuoi tap tin hay chua
        in.get(ch);
        cout << ch;
}
in.close();
out.close();
```

3.3 Đọc và ghi tập tin

- Đọc tập tin theo từng dòng. Đây là cách đọc tập tin phổ biến có thể áp dụng rộng rãi cho nhiều dạng thao tác.
- Đọc tập tin theo từng dòng sử dụng hàm getline(…) và vòng lặp duyệt tới cuối tập tin.

```
Ifstream A(…)
Getline(A, chuỗi dòng của tập tin)
```

- Ví dụ: đọc tập tin "test" theo từng dòng

```
ifstream in("test");
if(!in){
        cout << "Khong mo duoc file.\n";
        return 1;
}
/// vong lap xac dinh tung dong tren tap tin
for(string str; getline(in, str);) /// getline #include "string"
        cout << str << endl;
in.close();
```

4. Bài tập minh họa

- Tạo tập tin tên "trung binh" có 3 cột giá trị cách nhau khoảng TAB

46 56 12
12 34 56
45 78 90

- Tính giá trị trung bình {X, Y, Z} (làm tròn 2 chữ số sau dấu thập phân). Sau đó, ghi tích hợp giá trị trung bình lại trong tập tin. Kết quả mong muốn của nội dung tập tin "trung binh" như sau:

46 56 12
12 34 56
45 78 90
34.33 56 52.66

Các bước giải quyết (bài tập minh họa)

- Ghi tập tin với ban đầu
- Tính giá trị trung bình (cộng tích lũy chia cho số lần tích lũy)
  - Đọc từng dòng trên tập tin
  - Mỗi dòng:
    - Phân tách chuỗi tương ứng với các cột
    - Chuyển đổi từ kiểu string sang kiểu int
    - Cộng dồn và đếm số lần tích lũy
  - Tính giá trị trung bình, làm tròn số
  - Ghi tích hợp vào tập tin.

Tạo tập tin (bài tập minh họa)

```
#include "iostream"
#include "fstream"
using namespace std;
int main(){
        ofstream out(“c:\\abc\\trung binh.txt");
        if(!out)        {
                cout << "Khong mo duoc file.\n";
                return 1;
        }
        out << 46 << "\t" << 56 << "\t" << 12 << endl;
        out << 12 << "\t" << 34 << "\t" << 56 << endl;
        out << 45 << "\t" << 78 << "\t" << 90 << endl;
        out.close();
        return 0;
}
```

Tính giá trị trung bình (bài tập minh họa)

```
ifstream in("trung binh");

float s1 = 0, s2 = 0, s3 = 0; /// Dung de tinh tong va trung binh
int step = 0, t;
for(string str; getline(in, str);)       {
        istringstream iss(str); /// su dung input string stream
        step = 0;
        do{
                string sub;
                iss >> sub; /// tuong tu cin: truyen vao bien sub gia tri duoc auto delimiter.
                if(sub != ""){
                        t = atoi(sub.c_str()); /// chuyen doi string sang int.
                        if(step == 0)           s1 += t;
                        else if(step == 1)      s2 += t;
                        else if(step == 2)      s3 += t;
                        ++step;
                }
        } while (iss);
}
s1 /= step; s2 /= step; s3 /= step;
in.close();
```

Ghi tích hợp vào tập tin (bài tập minh họa)

```
/// Lam tron 2 chu so
float f = 100;
s1 = (int)(s1 * 100) / f;
s2 = (int)(s2 * 100) / f;
s3 = (int)(s3 * 100) / f;

/// Ghi tich hop
ofstream out("trung binh", ios::app);
if(!out)       {
        cout << "Khong mo duoc file.\n";
        return 1;
}
out << endl;
out << s1 << "\t" << s2 << "\t" << s3 << endl;
out.close();
```

5. Bài tập bắt buộc

1. Nhập một dãy số nguyên và ghi thành file TEXT trên đĩa với tên file là “Integer.txt” theo yêu cầu sau : Dòng đầu ghi dòng chữ: “Begin Day”, dòng thứ hai ghi số phần tử của dãy, các phần tử của dãy được ghi ở các dòng tiếp theo, mỗi dòng gồm 20 phần tử (dòng cuối cùng có thể ít hơn 20 phần tử), cuối cùng ghi thêm 1 dòng chữ: “End Day”. Viết hàm đọc dữ liệu của dãy từ file và tìm phần tử lớn nhất của dãy.
2. Nhập một ma trận vuông và ghi thành file TEXT trên đĩa với tên file là “matran.txt” theo yêu cầu sau : Dòng đầu ghi cấp của ma trận, các dòng tiếp theo là mỗi dòng của ma trận. Hai số kề nhau được lưu bởi khoảng trắng. Sau đó đọc file “matran.txt” để tìm phần tử lớn nhất, phần tử nhỏ nhất trong ma trận, và xuất ra dòng có tổng số các phần tử là lớn nhất, …
3. Nhập một danh sách sinh viên (mỗi sinh viên có mã, họ tên, năm sinh, điểm trung bình) và ghi lên đĩa thành một tập tin. Viết hàm đọc file và xuất ra danh sách, nhập thêm dữ liệu cho mẫu tin của một sinh viên vào cuối tập tin, tìm kiếm sinh viên theo mã, …