

CSC10001

HƯỚNG DẪN XỬ LÝ TẬP TIN VỚI C++

FIT-HCMUS

Contents

1	Các kiểu dữ liệu giúp xử lý tập tin thông dụng	2
1.1	fstream	2
1.2	ifstream	2
1.3	ofstream	3
2	Một số hàm được sử dụng để xử lý tập tin	3
2.1	open()	3
2.2	close()	4
2.3	is_open()	4
2.4	eof()	4
2.5	getline()	4
2.6	Toán tử extraction (»)	5
2.7	Toán tử insertion («)	6
3	Ví dụ mẫu	6
4	Một số lưu ý	8

1 Các kiểu dữ liệu giúp xử lý tập tin thông dụng

1.1 fstream

- Kiểu dữ liệu cho phép mở tập tin để **đọc, ghi hoặc ghép nối**,... tập tin.
- Ví dụ sử dụng:

```
#include <fstream>    // std::fstream

int main ()
{
    std::fstream fs;

    // Mo tap tin "test.txt" cho phép doc hoac ghi hoac chinh sua (append)
    fs.open("test.txt", std::fstream::in | std::fstream::out | std::fstream::app);

    // Ghi dong chu Hello World vao tap tin
    fs << "Hello World";

    // Dong tap tin
    fs.close();

    return 0;
}
```

- Tham khảo thêm: [fstream C++](#)

1.2 ifstream

- Kiểu dữ liệu cho phép mở tập tin để **đọc**.
- Ví dụ sử dụng:

```
#include <iostream> // std::cout
#include <fstream> // std::ifstream
#include <string> // std::getline

int main()
{
    std::ifstream ifs;
    // Mo tap tin "test.txt" de doc
    // Noi dung tap tin "test.txt" chi bao gom 1 dong Hello World
    ifs.open("test.txt");

    std::string s = "";
    std::getline(ifs, s); // Doc dong du lieu Hello World vao bien s
}
```

```
std::cout << s;    // Man hình console xuất hiện dòng Hello World

// Đóng tập tin
ifs.close();

return 0;
}
```

- Tham khảo thêm: [ifstream C++](#)

1.3 ofstream

- Kiểu dữ liệu chỉ cho phép mở tập tin để ghi hoặc chỉnh sửa.
- Ví dụ sử dụng:

```
#include <fstream>    // std::ofstream

int main ()
{
    std::ofstream ofs;

    // Mở tập tin "test.txt" cho phép ghi hoặc chỉnh sửa (ghép nối - append)
    ofs.open ("test.txt", std::ofstream::out | std::ofstream::app);

    // Ghi dòng Hello World vào tập tin
    ofs << "Hello World";

    // Đóng tập tin
    ofs.close();

    return 0;
}
```

- Tham khảo thêm: [ofstream C++](#)

2 Một số hàm được sử dụng để xử lý tập tin

2.1 open()

- Dùng để mở tập tin.
- Đối với kiểu dữ liệu khác nhau, ta sẽ có cách gọi hàm khác nhau (chủ yếu ở tham số thứ 2). Xem lại phần 1.

2.2 close()

- Dùng để đóng tập tin.
- Ví dụ:

```
f.close();
```

2.3 is_open()

- Dùng để kiểm tra xem liệu tập tin có được mở.
- Trả về **true** nếu phần mở được tập tin, ngược lại trả về **false**.
- Ví dụ:

```
if (!ifs.is_open())
{
    cout << "Khong mo duoc file!";
    return 0;
}
```

2.4 eof()

- Toán tử kiểm tra liệu đã đọc đến cuối tập tin chưa.
- Trả về **true** nếu đã đọc đến cuối tập tin, ngược lại trả về **false**.
- Ví dụ:

```
while (!ifs.eof())    // Lap khi chua doc den cuoi tap tin
{
    getline(ifs, s);
}
```

2.5 getline()

- Dùng để đọc dữ liệu trong tập tin.
- Để sử dụng hàm này, chúng ta cần **include <string>**.
- Có các cách sử dụng sau đây, giả sử dòng dữ liệu cần đọc là "Ten;Tuoi;NamSinh\n":

1. Đọc hết một dòng trong tập tin.

Ví dụ: `getline(ifs, s);`

Sau khi gọi câu lệnh trên, toàn bộ dữ liệu của một dòng trong tập tin sẽ được đọc vào **s**, đồng thời loại bỏ ký tự xuống dòng (**\n**).

→ **s = "Ten;Tuoi;NamSinh"**.

2. Đọc đến khi gặp dấu phân cách (delemeter).

Ví dụ: `getline(ifs, s, ';');`

Sau khi gọi câu lệnh trên, `s` sẽ nhận thông tin từ đầu dòng đến khi gặp dấu phân cách `';`'.

→ `s = "Ten"`.

2.6 Toán tử extraction (»)

- Dùng để đọc dữ liệu từ tập tin.
- Toán tử này được sử dụng tương tự cách sử dụng `std::cin >>`.
- Đọc đến khi gặp dấu cách `' '`, dấu tab `'\t'`, hoặc dấu xuống dòng `'\n'`...
- Ví dụ đọc mảng 1 chiều từ tập tin `"in.txt"` có nội dung như sau:

```
5
1 3 5 7 -1
```

Chỉ cần sử dụng toán tử extraction mà không cần sử dụng `getline`.

```
...
int arr[100];
int n = 0;
ifstream ifs("in.txt");           // Mở file để đọc

ifs >> n;                          // Đọc số lượng phần tử của mảng
for (int i = 0; i < n; i++)
    ifs >> arr[i];                // Đọc từng phần tử

ifs.close();
...
```

- Trong trường hợp nội dung tập tin không có số lượng phần tử đi kèm mà chỉ có nội dung mảng (1 3 5 7 -1), ta có thể sử dụng vòng lặp `while` để đọc nội dung tập tin.

```
...
int arr[100];
int n = 0;
ifstream ifs("in.txt");

while (ifs >> arr[n])
    n++;

ifs.close();
...
```

Lưu ý: Không gộp `n++` vào `ifs >> arr[n]`. Khi nội dung tập tin có dấu cách hoặc dấu xuống dòng ở cuối → việc gộp sẽ làm tăng `n` → phần tử cuối cùng là rác.

2.7 Toán tử insertion («)

- Dùng để ghi dữ liệu xuống tập tin.
- Toán tử này được sử dụng tương tự cách sử dụng `std::cout << .`
- Ví dụ bình phương từng phần tử trong mảng đọc được phía trên và ghi xuống tập tin "out.txt".

```
...
ofstream ofs("out.txt");           // Mo file de ghi

ofs << n;                           // Ghi so luong phan tu cua mang
for (int i = 0; i < n; i++)
    ofs << arr[i] * arr[i] << " "; // Ghi tung phan tu

ofs.close();
...
```

3 Ví dụ mẫu

Đề bài: Đọc và lưu trữ dữ liệu trong tập tin "data.txt" có nội dung như sau:

```
Ten;Toan;Van
Hoa;9;7
Loan;8;8
Hung;7;8
Thanh;10;9.5
```

Sau đó, ghi xuống tập tin "result.txt" Tên và điểm tổng 2 môn của học sinh, có định dạng như sau:

```
Ten;Tong
Hoa;16
Loan;16
Hung;15
Thanh;19.5
```

Bài giải:

```
#include <iostream>
#include <fstream> // std::ifstream, std::ofstream
#include <string> // std::getline, std::stof
#include <vector>

using namespace std;

struct Student
{
```

```

    string name;
    float math, literature;
};

int main()
{
    // PHAN 1: DOC TAP TIN
    // Mo tap tin de doc
    ifstream ifs;
    ifs.open("data.txt");

    // Thoat chuong trinh neu khong mo duoc tap tin
    if (!ifs.is_open())
    {
        cout << "Khong mo duoc file!";
        return 0;
    }

    // Doc bo dong "Ten;Toan;Van" trong tap tin
    string ignore_line = "";
    getline(ifs, ignore_line);

    // Tao cac bien du lieu de xu ly
    vector<Student> list_student;
    Student temp_student;
    string name = "";
    string math = "";
    string literature = "";

    // Duyet den cuoi khi khong doc duoc
    while (getline(ifs, name, ';'))
    {
        // Noi dung name duoc doc truoc de tranh truoc hop lap thong tin khi co 1 dong trong o cuoi
        // Doc noi dung tu file vao cac bien tam
        getline(ifs, math, ';');
        getline(ifs, literature, '\n');

        // Gan du lieu tu cac bien tam vao struct Student
        temp_student.name = name;
        temp_student.math = stof(math);           // Chuyen diem toan tu string thanh float
        temp_student.literature = stof(literature); // chuyen diem van tu string thanh float

        // Day hoc sinh doc duoc vao vector cac hoc sinh
        list_student.push_back(temp_student);
    }
}

```

```

// Dong tap tin
ifs.close();

//-----
// PHAN 2: GHI THONG TIN XUONG TAP TIN
// Mo tap tin de ghi
ofstream ofs;
ofs.open("result.txt");

// Thoat chuong trinh neu khong mo duoc tap tin
if (!ofs.is_open())
{
    cout << "Khong mo duoc file!";
    return 0;
}

// Ghi thong tin xuong tap tin
ofs << "Ten;Tong\n";
for (int i = 0; i < list_student.size(); i++)
{
    ofs << list_student[i].name << ";";
    ofs << list_student[i].math + list_student[i].literature << "\n";
}

// Dong tap tin
ofs.close();

return 0;
}

```

4 Một số lưu ý

1. Khi mở tập tin, **BẮT BUỘC** phải đóng tập tin.
2. Khi mở tập tin để ghi (`std::ofstream::out`) có 2 trường hợp xảy ra:
 - (a) Tập tin không tồn tại → tạo tập tin → ghi nội dung mới vào tập tin.
 - (b) Tập tin đã tồn tại → xóa toàn bộ nội dung đã có từ trước trong tập tin → ghi nội dung mới vào tập tin.
3. Muốn mở tập tin để chỉnh sửa, hoặc thêm thông tin vào sau tập tin → sử dụng `std::ofstream::app`.

HẾT