Ngôn ngữ lập trình C++

Chương 3. Kiểu dữ liệu Tệp

3.1 Khái niệm

- Tệp là tập hợp các byte dữ liệu sắp xếp tuần tự theo một trật tự nhất định lưu trữ trên **bộ nhớ ngoài**.
- Mỗi tệp có ***tên*** và được định vị trên thiết bị nhớ ngoài (thông qua đường dẫn đến tệp).
- Lý do dùng tệp:
  - Tái sử dụng, có thể sao lưu giữa các thiết bị nhớ ngoài khác nhau.
  - Lưu giữ những dữ liệu lớn, những dữ liệu không thường xuyên có được.
- Phân loại tệp trong C++:
  - Tệp nhị phân (binary)
  - Tệp văn bản (text)

Tệp nhị phân

- Lưu trữ dữ liệu nguyên thủy dưới dạng mã.
- Ví dụ: short year=2021; Khi chương trình được dịch, biến year sẽ được lưu dưới dạng 2 Byte:

  00000111 11100101
- Công cụ xem tệp dưới dạng nhị phân: Ví dụ Hex editor

Tệp văn bản

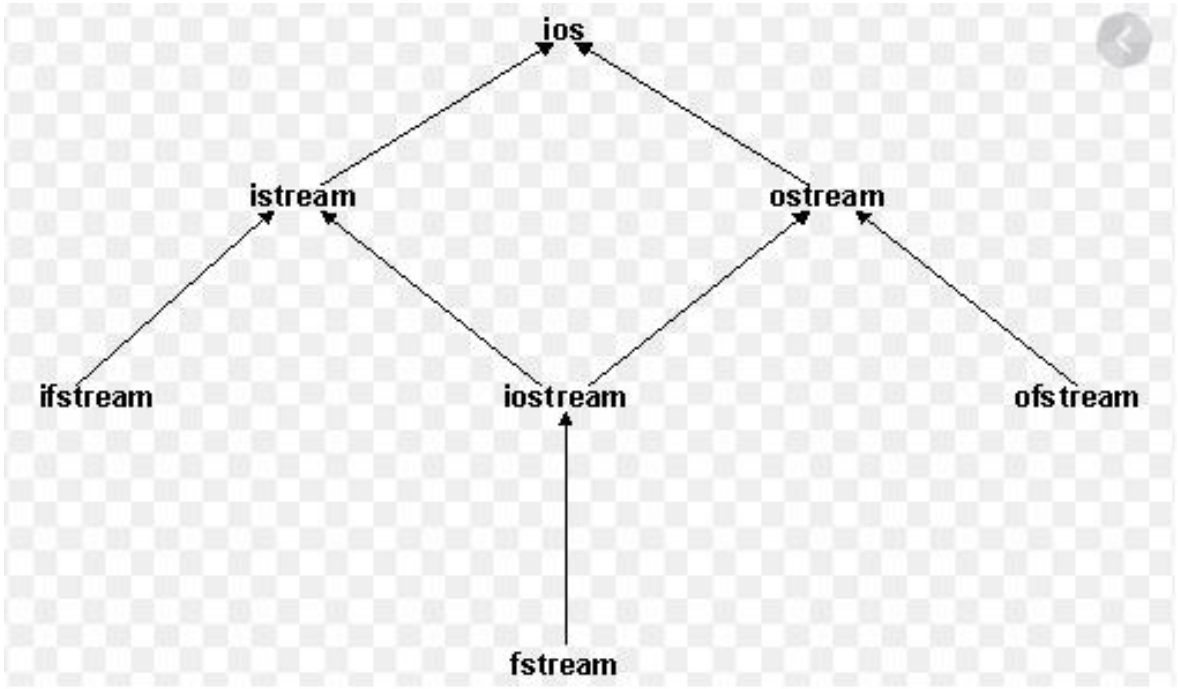
- Tệp văn bản lưu trữ dữ liệu dưới dạng ký tự với các giá trị trong bảng ASCII và ở định dạng có thể đọc được.
- Ví dụ: short year=2021; Khi tệp nguồn chương trình được lưu, giá trị “2021” sẽ chiếm 4 Byte, số được coi như ký tự, mỗi số chiếm 1 Byte.
- Có thể đọc/viết tệp văn bản bằng các chương trình soạn thảo văn bản, ví dụ Notepad.
- Khác với tệp nhị phân, tệp văn bản có định dạng theo các dòng.
- Môn học sẽ tập trung vào xử lý tệp văn bản.

Các thao tác làm việc với tệp

- Mở tệp (để đọc/ghi/thêm dữ liệu…)
- Đọc/Ghi tệp
- Đóng tệp

3.2 Tệp trong C++

- Liên kết tệp với dòng dữ liệu xuất nhập:



- Lớp ifstream kế thừa từ istream, thực hiện các thao tác nhập file,
- Lớp ofstream kế thừa từ ostream, thực hiện các thao tác xuất file,
- Lớp fstream kế thừa từ iostream, thực hiện các thao tác nhập/xuất file.

A. Khai báo

- Cần khai báo thư viện fstream:

  #include<fstream>

- Khai báo biến tệp:

ifstream ten_bientep1; //khai báo biến tệp để đọc

ofstream ten_bientep2; // khai báo biến tệp để ghi

fstream ten_bientep3; // khai báo biến tệp để đọc và ghi

B. Mở tệp

- **Cú pháp:**

  ten_bientep.open(“đường dẫn\\tên_tệp”, chế độ);

- Có thể viết:

  // mặc định mở để đọc:

  ifstream ten_bientep(“đường dẫn\\tên_tệp”);

  fstream ten_bientep(“đường dẫn\\tên_tệp”);

  // mặc định mở tệp để ghi:

  ofstream ten_bientep(“đường dẫn\\tên_tệp”);

- Nếu mở tệp với mục đích khác thì bổ sung thêm tham số ***chế độ***, ví dụ:

  fstream ten_bientep(“đường dẫn\\tên_tệp”, **chế độ**)

- Nếu không có “đường dẫn”: Tệp ở thư mục hiện hành.

B. Mở tệp

- **Chế độ:**

ios :: in	Mở một tệp tin để đọc
ios :: out	Mở một tệp tin có sẵn để ghi
ios :: app	Mở một tệp tin có sẵn để thêm dữ liệu vào cuối tệp.
ios :: ate	Mở tệp tin và đặt con trỏ tệp tin vào cuối tệp.
ios :: trunc	Nếu tệp tin đã có sẵn thì dữ liệu của nó sẽ bị mất.
ios :: nocreate	Mở một tệp tin, tệp tin này bắt buộc phải tồn tại
ios :: noreplace	Chỉ mở têp tin khi tệp tin chưa tồn tại.
ios :: binary	Mở một tệp tin ở chế độ nhị phân
ios :: text	Mở một tệp tin ở chế độ văn bản.

- Có thể mở tệp với đồng thời nhiều chế độ
- ***Ví dụ:***

```
fstream data;
data.open("D\\Cpp\\Document.txt", ios::in | ios::text);
```

Hoặc `fstream data ("D\\Cpp\\Document.txt", ios::in | ios::text);`

B. Mở tệp

- Nếu không mở được tệp $\Rightarrow$ Nên báo lỗi bằng phương thức ***fail()***
  - Hàm trả về **true** nếu đối tượng ifstream, ofstream, fstream liên kết đến file không thành công, false nếu liên kết thành công.
- Hoặc báo lỗi mở tệp bằng phương thức ***is_open()***
  - Hàm trả về true nếu mở tệp thành công, false nếu mở tệp không thành công.
- Ví dụ:

```
fstream data;
data.open(“D:\\Cpp\\Document.txt”, ios::in |ios::text);
if (data.fail()) //hoặc if ( ! data.is_open())
   { cout << “Khong mo đuoc file!" << endl;
     return -1; } // Thoát khỏi chương trình.
```

C. Đọc tệp văn bản

- Sử dụng toán tử: “ >> ” tương tự như cin >> nhưng thay vì “cin” là **ten_bientep**.
- Ví dụ:

```
int n;
data >> n;
```

- Nếu đọc xâu ký tự (chứa dấu cách) trong tệp văn bản: Dùng hàm

```
getline(ten_bientep, biến_xâu);
```

- Nên làm sạch bộ đệm bằng ten_bientep.ignore() trước khi dùng getline nếu đọc tệp cả số lẫn xâu ký tự.
- Ví dụ:

```
int n;
string xau;
data >> n;
data.ignore();
getline(data, xau);
```

C. Đọc tệp văn bản

- Kiểm tra kết thúc tệp: Sử dụng phương thức/hàm ***eof()***:

ten_bientep.eof()

  - Nếu đã gặp cuối tệp khi đọc, hàm trả về giá trị *true*.
  - Ngược lại, khi chưa đọc đến cuối tệp, hàm trả về giá trị *false*.

- Ví dụ: Đọc dữ liệu đến hết tệp

```
int n;
while (! data.eof())
{
        data >> n;
}
```

D. Ghi tệp văn bản

- Sử dụng toán tử: “ << ” tương tự như cout << nhưng thay vì “cout” là **ten_bientep**.
- Có thể sử dụng mọi định dạng như khi cout << ra màn hình.
- Ví dụ: file_inoutput.cpp
- Thêm dữ liệu vào tệp: Dùng chế độ ios::app
- Ví dụ: file_appstring.cpp

E. Đóng tệp

- Sử dụng phương thức ***close()***
- Ví dụ: data.close();
- Chú ý: Mỗi biến file chỉ có thể liên kết tới 1 tệp cụ thể tại một thời điểm nên trước khi tạo liên kết đến tệp khác cần phải đóng liên kết trước đó.

Tổng kết các bước làm việc với tệp

1. Khai báo đối tượng để đọc hoặc ghi từ tệp

   ifstream f;

   ofstream f;

   fstream f;

2. Mở tệp (đọc / ghi / cả đọc cả ghi)

   f.open("Input.dat", ios::out|ios::in);

   if (f.fail()) // Hoặc if (! f.is_open())

   cout<<"Loi khi mo tep";

   …

3. Đọc / ghi dữ liệu
   - Dùng toán tử ">>" để đọc dữ liệu và toán tử "<<" để ghi dữ liệu vào tệp.
   - Chú ý kiểm tra kết thúc tệp nếu cần thiết (chủ yếu là khi đọc tệp)

4. Đóng tệp

   f.close();