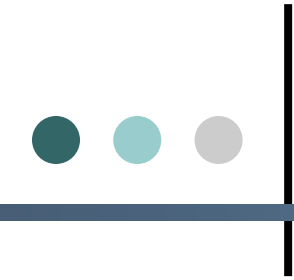

CHƯƠNG 4

NHẬP/XUẤT FILE TRONG C++

Nội dung

1. Tổng quan về nhập/xuất theo luồng trong C++
2. Các kiểu file trong C++
3. Luồng fstream và các tham số.
4. Nhập/Xuất file văn bản:
   - 4.1. Khai báo file văn bản.
   - 4.2. Nhập dữ liệu vào file.
   - 4.3. Ghi dữ liệu ra file.
5. Nhập/Xuất file nhị phân:
   - 5.1. Khai báo file nhị phân
   - 5.2. Đọc dữ liệu từ file
   - 5.3. Ghi dữ liệu ra file
5. Nhập/Xuất mảng cấu trúc và áp dụng

Tổng quan về nhập/xuất theo luồng trong C++

- Các lớp chuẩn chứa dữ liệu và các phương thức dùng cho nhập/xuất dữ liệu gọi là stream (luồng).
- Các lớp dùng với các thiết bị nhập xuất cơ bản (keyboard, monitor):
  - Lớp ios là lớp cơ sở về stream. Lớp ios chứa các thuộc tính để định dạng việc nhập/xuất và kiểm tra lỗi.
  - Các lớp istream và ostream kế thừa từ ios, bổ sung thêm các toán tử <<, >> và các hàm get, getline, read, write, …
  - Lớp iostream là tổng hợp của istream và ostream.
- Đối tượng của các lớp trên được gọi là các luồng dữ liệu.

Tổng quan về nhập/xuất theo luồng trong C++

- Các lớp dùng với các thiết bị khác như máy in, ổ đĩa (file), …:
  - ifstream: được sử dụng trong tạo file và để ghi thông tin vào file.
  - ofstream: được sử dụng để đọc thông tin từ file.
  - fstream: có các khả năng của cả ofstream và ifstream.

Tổng quan về nhập/xuất theo luồng trong C++

- Toán tử <<, >>: dùng với cin và cout
- Các hàm nhập ký tự và chuỗi ký tự:
  - Nhập ký tự:
    - cin.get(): Hàm trả về một ký tự

    Ví dụ: char ch;
    ch = cin.get();
    //nếu nhập A↵, ch nhận 'A', trong cin còn ↵.
    - cin.get(ch): Hàm nhập ký tự cho ch và lại một tham chiếu tới cin.

    Ví dụ: char c, d;
    cin.get(c).get(d);
    //nếu nhập AB↵ thì c nhận 'A', b nhận 'B', cin còn ↵

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Nhập xâu ký tự:
    - cin.get(s, n, fchar): Hàm nhập cho s dãy ký tự từ cin.
      - ✓ Dãy được tính từ ký tự đầu tiên trong cin cho đến khi đã đủ n – 1 ký tự hoặc gặp ký tự kết thúc fchar.
      - ✓ Tự động gán ký tự kết thúc xâu (‘\0’) vào s sau khi nhập xong.
      - ✓ Tham số fchar có thể bỏ qua. Khi đó, fchar được hiểu là ký tự xuống dòng (↵).
      - ✓ Các lệnh có thể viết nối nhau như sau: **cin.get(s1,n1).get(s2.n2)**;

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Nhập xâu ký tự:
    - cin.getline(s, n, fchar): Hàm nhập cho s dãy ký tự từ cin. Hoạt động tương tự cin.get(s, n, fchar). Điểm khác là s sẽ xóa ký tự xuống dòng (↵) khỏi bộ đệm.
    - cin.ignore(n): Đọc và loại bỏ n ký tự còn trong bộ đệm
  - Định dạng:
    - cout.width(n): Cho phép các giá trị in ra màn hình với độ rộng n.
    - cout.fill(ch): ch là ký tự chèn vào khoảng rộng hơn giá trị cần in.
    - cout.precision(n): làm tròn n ký số phần thập phân

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Định dạng:
    - cout.width(n): Cho phép các giá trị in ra màn hình với độ rộng n.
    - cout.fill(ch): ch là ký tự chèn vào khoảng rộng hơn giá trị cần in.
    - cout.precision(n): làm tròn n ký số phần thập phân.
  - Các cờ định dạng:
    - Bật/tắt cờ định dạng:
      - cout.setf(danh_sách_cờ): Bật cờ
      - cout.unsetf(danh_sách_cờ): Tắt cờ
        - ✓ Các cờ trong danh sách cách nhau bởi dấu "|"

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Các cờ:
    - Canh lề:
      - ios::left: canh trái (ký tự độn nằm sau)
      - ios::right: canh phải (ký tự độn nằm trước)
      - ios::internal: giống canh phải. Tuy nhiên, dấu của giá trị sẽ in trước, tiếp theo là ký tự độn, rồi đến giá trị số.
    - Định dạng số nguyên:
      - ios::dec: in số nguyên dưới dạng thập phân
      - ios::oct: in số nguyên dưới dạng cơ số 8
      - ios::hex: in số nguyên dưới dạng cơ số 16

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Các cờ:
    - Định dạng số thực:
    - ios::fixed: in số thực dạng dấu phẩy tĩnh
    - ios::scientific: in số thực dạng dấu phẩy động
    - ios::showpoint: in đủ n số lẻ phần thập phân. Nếu tắt thì không in các số 0 vô nghĩa.
    - Định dạng hiển thị:
    - ios::showpos: nếu bật thì sẽ in dấu cộng (+) trước số dương.
    - ios::showbase: nếu bật sẽ in số 0 trước số hệ 8, 0x trước số hệ 16.
    - ios::uppercase: ký tự trong hệ 16 sẽ viết hoa.

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Các bộ và hàm định dạng:
    - Bộ định dạng:
    - ✓ dec: tương tự ios::dec
    - ✓ oct: tương tự ios::oct
    - ✓ hex: tương tự ios::hex.
    - ✓ endl: xuất ký tự xuống dòng.
    - ✓ flush: đẩy toàn bộ dữ liệu ra dòng xuất

Tổng quan về nhập/xuất theo luồng trong C++

- Các hàm nhập ký tự và chuỗi ký tự:
  - Các bộ và hàm định dạng:
    - Hàm định dạng:
      - ✓ setw(n): tương tự cout.width(n).
      - ✓ setprecision(n): tương tự cout.precision(n).
      - ✓ setfill(c): tương tự cout.fill(c).
      - ✓ setiosflag(|): tương tự cout.setf(|)
      - ✓ resetiosflag(|): tương tự cout.unsetf(|)

Các kiểu file trong C++

- Khái niệm về file:
  - File là một tổ chức thông tin đặt trên các thiết bị nhớ ngoài, được truy cập thông qua tên file.
  - Nội dung của file được xác định bởi người tạo ra file.
  - Hệ điều hành quản lý cá thông tin cơ bản của file:
    - ✓ Tên file,
    - ✓ Phần mở rộng của file,
    - ✓ Độ lớn file,
    - ✓ Ngày giờ tạo file,
    - ✓ Thời gian gần nhất truy cập file,
    - ✓ …

Các kiểu file trong C++

- Cơ bản có hai loại file:
  - File văn bản:
    - ✓ Trong file văn bản mỗi byte được xem là một kí tự.
    - ✓ Tuy nhiên nếu 2 byte 10 (LF), 13 (CR) đi liền nhau thì được xem là một kí tự và nó là kí tự xuống dòng.
    - ✓ Như vậy file văn bản là một tập hợp các dòng kí tự với kí tự xuống dòng có mã là 10.
    - ✓ Kí tự có mã 26 được xem là kí tự kết thúc file.
  - File nhị phân:
    - ✓ Dữ liệu trong file được xem là dãy byte bình thường.
    - ✓ Mã kết thúc file được chọn là -1, được định nghĩa là EOF trong stdio.h.
    - ✓ Các thao tác trên file nhị phân thường đọc ghi từng byte một, không quan tâm ý nghĩa của byte.

Các kiểu file trong C++

- Các thao tác trên file: Cơ bản gồm:
  - ✓ Mở file
  - ✓ Đọc file
  - ✓ Đóng file,
  - ✓ Xóa file,
  - ✓ …

Các kiểu file trong C++

- Các phương thức đọc ghi ký tự:
  - gut(c); //ghi ký tự ra file
  - get(c); //đọc ký tự từ file
- Các phương thức đọc ghi dãy ký tự
  - write(char *buf, int n); //ghi n ký tự trong buf ra dòng xuất
  - read(char *buf, int n); //nhập n ký tự từ buf ra dòng nhập
  - gcount(); //cho biết số ký tự read đọc được

Các kiểu file trong C++

❖ Ví dụ

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <ctype.h>

using namespace std;
```

```
int main() {
    fstream fnguon("DSSV", ios::in | ios::binary);
    fstream fdich("Data_2", ios::out | ios::binary);
    char ch;
    while (!fnguon.eof()) {
        fnguon.get(ch);
        fdich.put(ch);
    }
    fnguon.close();
    fdich.close();
    getch();
    return 0;
}
```

Các kiểu file trong C++

❖ Ví dụ

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <fstream>

using namespace std;

void ghi(char*,  char*);
```

```
int main(){
        char kitu[] = {'a','b'};
        ghi("D:\\File_b.dat",  kitu);
        system("pause");
        return 0;
}
void ghi(char *name, char kitu[]){
        ofstream ghi(name, ios::binary);
        for(int i=0; i<2; i++)
            ghi.write(&kitu[i], sizeof(kitu[i]));
        cout << "Da hoan thanh" << endl;
        ghi.close();
}
```

Các kiểu file trong C++

❖ Ví dụ

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <conio.h>
```

```
int main()
{
   fstream fnguon("DATA1", ios::in | ios::binary);
   fstream fdich("DATA2", ios::out | ios::binary);
   char buf[2000] ;
   int n = 2000;
   while (n) {
      fnguon.read(buf, 2000);
      n = fnguon.gcount();
      fdich.write(buf, n);
   }
   fnguon.close();
   fdich.close();
}
```

Các kiểu file trong C++

- Tạo đối tượng file:

```
ifstream <tên_biến> //file chỉ đọc
ofstream <tên_biến> //file để ghi
```

  - Ví dụ 1:

```
ifstream fi_1 //file chỉ đọc
ofstream fo_1 //file để ghi
fstream f //file vừa đọc vừa ghi
f.open("BaiGiang"); //mở file BaiGiang và gắn với f
```

  - Ví dụ 2:

```
ifstream fi("BaiGiang") //mở file BaiGiang gắn với đối
                        //tượng fi để đọc
ofstream fo("BaiGiang") // mở file BaiGiang gắn với đối
                        //tượng fo để ghi
```

Các kiểu file trong C++

- Các chế độ quy định cách thức làm việc với file:
  - ios::binary: Quan niệm file theo kiểu nhị phân. Ngầm định là kiểu văn bản
  - ios::in: file để đọc
  - ios::out: file để ghi. Nếu file đã có trên đĩa thì nội dung sẽ bị ghi đè.
  - ios::app: bổ sung vào cuối file.
  - ios::trunc: xóa nội dung file đã có
  - ios::nocreate: không làm gì nếu file chưa có.
  - ios::replace: không làm gì nếu file đã có.

Các kiểu file trong C++

- Đóng file và giải phóng đối tượng:
  - Để đóng file được đại diện bởi f, sử dụng phương thức close như sau:

    f.close();
  - Sau khi đóng file (và giải phóng mối liên kết giữa đối tượng và file) có thể dùng đối tượng để gắn và làm việc với file khác bằng phương thức open như trên.

Các kiểu file trong C++

- Đóng file và giải phóng đối tượng:
- Ví dụ:

```
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
void main() {
        ofstream f; // khai báo đối tượng f
        int x;
        f.open("SoNguyen"); // mở file SoNguyen và gắn với f
        for (int i = 1; i<=10; i++) {
                cin >> x;
                f << x << ' ';
        }
        f.close();
}
```

Các kiểu file trong C++

- Kiểm tra sự tồn tại của file, kiểm tra hết file:
  - Việc mở một file chưa có để đọc sẽ gây ra lỗi và làm dừng chương trình.
  - Khi xảy ra lỗi mở file, giá trị trả lại của phương thức bad là một số khác 0. Dùng phương thức này để kiểm tra sự tồ tại của một file trên đĩa.

```
ifstream f("Bai_tap");
if (f.bad()) {
        cout << "file Bai_tap chua co";
        exit(1);
}
```

  - Khi đọc hoặc ghi, con trỏ file sẽ chuyển dần về cuối file. Khi con trỏ ở cuối file, phương thức eof() sẽ trả lại giá trị khác không.

Luồng fstream và các tham số

Stt	Tên phương thức	Ý nghĩa
1	fi.open(“tên_file”)	Mở file đã tồn tại để đọc.
2	fi.open(“tên_file”)	Tạo file mới để ghi.
3	fi.open(“tên_file”, ios::in, ios::out)	Mở file hoặc tạo mới file để đọc hoặc ghi.
4	fi.getline(line, n);	Đọc một dòng từ file
5	fi>>tên-biến;	Nhận giá trị cho biến từ file
6	fi.read (buff, n)	Đọc n byte vào buff từ file
7	fi>>tên-biến	Ghi nội dung của biến vào file
8	fi.write(buff, n)	Ghi n byte từ buff vào file
9	fi.close()	Đóng file
10	fi.eof()	Kiểm tra cuối file
11	fi.fail()	Kiểm tra lỗi đọc hoặc ghi file
12	fi.seekg(n, ios::<const>)	Di chuyển vị trí thứ n trong file để đọc
13	fi.seekp(n, ios::<const>)	Di chuyển vị trí thứ n trong file để ghi

Nhập/Xuất file văn bản

- #include <fstream>
- Sử dụng ifstream (file chỉ đọc), ofstream (chỉ ghi), fstream (đọc/ghi)
- Đọc/ghi dữ liệu dùng các toán tử >> và << tương tự như với vào/ra chuẩn
- Mở file:

```
ifstream f1("ten_file", ios::in | ios::binary);
ofstream f2;
f2.open("ten_file", ios::out | ios::trunc);
```

  - Các mode:

app	**Nhảy con trỏ tới cuối file khi ghi**	**trunc**	**Xoá nội dung cũ khi mở**
ate	**Con trỏ tới cuối file**	**binary**	**File nhị phân**
in	**Cho phép đọc**	**out**	**Cho phép ghi**

Nhập/Xuất file văn bản

- Đóng file:
  - f.close();
  - Có thể để đóng file tự động trong destructor khi các đối tượng bị huỷ
- Chú ý khi dùng fstream để dùng cả đọc và ghi: trước khi chuyển từ việc đọc sang ghi hoặc ngược lại, phải dùng hàm seekg/seekp(...)

Nhập/Xuất file văn bản

- Ví dụ: Đọc số từ bàn phím và ghi vào file

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

using namespace std;
```

```
void main()
{
    ofstream f; // khai báo (tạo) đối tượng f
    int x;
    f.open("DaySo"); // mở file DaySo và gắn với f
    for (int i = 1; i<=10; i++) {
        cin >> x;
        f << x << ' ';
    }
    f.close();
}
```

Nhập/Xuất file văn bản

❖ Ví dụ: Đọc file

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

using namespace std;
```

```
int main()  {
    ifstream textfile("text.txt");
    string s1; //Lưu một dòng vào file
    stringstream ss;
    //truyền data vào buffer qua stream ss
    ss << textfile.rdbuf();
    s1 = ss.str(); //assign to s1 string
    cout << s1.data() << endl;
    return 0;
}
```

Nhập/Xuất file nhị phân

- Mở file: thêm cờ ios::binary
- Đọc dữ liệu:

```
file.read(char* buffer, int size);
file.gcount(); // số byte đã được đọc
```

- Ghi dữ liệu:

```
file.write(char *buffer, int size);
```

- Kiểm tra lỗi đọc/ghi:

```
file.read/write(...)
if (!file) {...}
if (!file.read/write(...)) {...}
```

Nhập/Xuất file nhị phân

- Di chuyển con trỏ file: C++ phân biệt con trỏ đọc và con trỏ ghi:
  - ✓ Di chuyển con trỏ đọc file:

    file.seekg(int pos, ios::beg/cur/end)
  - ✓ Vị trí con trỏ đọc hiện tại:

    file.tellg()
  - ✓ Di chuyển con trỏ ghi file:

    file.seekp(int pos, ios::beg/cur/end)
  - ✓ Vị trí con trỏ ghi hiện tại:

    file.tellp()

Nhập/Xuất file nhị phân

- Ví dụ: copy file

```
bool copy_file(const char *src, const char *dst)
{
    ifstream fs(src, ios::in | ios::binary);
    ofstream fd(dst, ios::out | ios::binary| ios::trunc);
    if (!fs || !fd)
        return false;
    char buf[1024];
    while (fs) {
        fs.read(buf, sizeof(buf));
        fd.write(buf, fs.gcount());
    }
    return true;
}
```

Nhập/Xuất mảng cấu trúc và áp dụng

- Viết chương trình sau nhập danh sách sinh viên, ghi vào file 1, đọc ra mảng, sắp xếp theo tuổi và in ra file 2. Dòng đầu tiên trong file ghi số sinh viên, các dòng tiếp theo ghi thông tin của sinh viên gồm họ tên với độ rộng 24 kí tự, tuổi với độ rộng 4 kí tự và điểm với độ rộng 8 kí tự.

Nhập/Xuất mảng cấu trúc và áp dụng

```
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <ctype.h>

struct Sv {
    char *hoten;
    int tuoi;
    double diem;
};
```

```
using namespace std;
void nhap();
void ghifile(char *fname);
void docfile(char *fname);
int main()
{
    nhapSV();
    ghifile("D:\\DSSV.tes");
    docfile("D:\\DSSV.tes");
    cout << "Da hoan thanh";
    getch();
    return 0;
}
```

Nhập/Xuất mảng cấu trúc và áp dụng

```
void nhapSV()
{
    cout << "\nSo sinh vien: ";
    cin >> sosv;
    int n = sosv;
    sv = new Sinhvien[n+1]; // Bỏ phần tử thứ 0
    for (int i = 1; i <= n; i++)
    {
        cout << "\nNhap sinh vien thu: " << i << endl;
        cout << "\nHo ten: "; cin.ignore();
        cin.getline(sv[i].hoten,3);
        cout << "\nTuoi: "; cin >> sv[i].tuoi;
        cout << "\nDiem: "; cin >> sv[i].diem;
    }
}
```

Nhập/Xuất mảng cấu trúc và áp dụng

```
void ghifile(char fname)
{
      ofstream f(fname) ;
      f << sosv;
      f << setprecision(1) << setiosflags(ios::showpoint);
      for (int i=0; i<sosv; i++) {
            f << endl<<setw(24) << sv[i].hoten<<setw(4)<< sv[i].tuoi;
            f << setw(8) << sv[i].diem;
      }
      f.close();
}
```

Nhập/Xuất mảng cấu trúc và áp dụng

```
void docfile(char fname)
{
	ifstream f(fname);
	f >> sosv;
	for (int i=0; i<sosv; i++)
	{
		f.getline(sv[i].hoten, 25);
		f >> sv[i].tuoi >> sv[i].diem;
	}
	f.close();
}
```

Case study

- Ứng dụng ghi dữ liệu của bài tập nhóm ra file:
  - File văn bản
  - File nhị phân