

# KỸ THUẬT LẬP TRÌNH

## Chương 7. Tập Tin

[thuong.bui@ut.edu.vn](mailto:thuong.bui@ut.edu.vn)



## 7.1 Tổng quan về tệp tin

Tệp tin (tiếng Anh là File) - là hình thức lưu trữ dữ liệu phổ biến trên bộ nhớ phụ, gồm 2 loại:

- File văn bản (text file).
- File nhị phân (binary file).

### File văn bản

Chỉ lưu trữ thuần túy văn bản, trong đó các kí tự được biểu diễn bằng mã ASCII của nó, người dùng có thể đọc được.

#### Tính chất:

- Dễ truy xuất và xử lý
- Độ bảo mật kém
- Tốc độ truy xuất kém
- Kích thước lớn

### File nhị phân

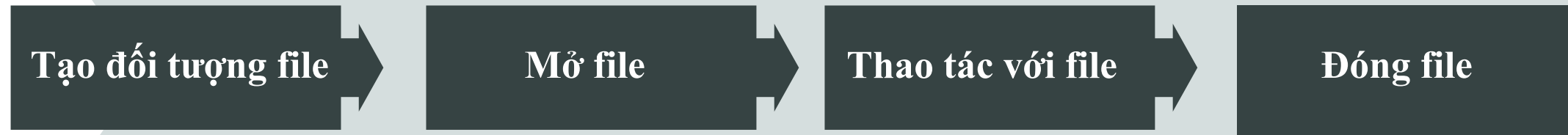
Là file chứa dữ liệu mà có ít nhất một vài chuỗi bit không thể biểu diễn ở dạng văn bản tron. Do đó file này chỉ có máy đọc được, còn người không đọc được.

#### Tính chất:

- Truy xuất và xử lý phức tạp
- Tốc độ truy xuất nhanh
- Tính bảo mật cao hơn

## 7.1 Tổng quan về tệp tin

### Quy trình làm việc với file



## 7.2 Khai báo kiểu tệp tin

Trong C++, để làm việc với các hàm xử lý file cần khai báo đầu chương trình: `#include <fstream>`

Trong thư viện fstream thì ta có 3 lớp stream cơ bản sau :

- **ifstream:** Dùng cho file nhập vào. Loại này chỉ có thể được dùng để đọc dữ liệu từ file vào bộ nhớ mà thôi.
- **ofstream:** Dùng cho file xuất ra. Loại này thì có thể dùng để tạo ra files và chép dữ liệu vào chúng.
- **fstream:** Đây là kênh file.(File stream). Loại này thì có thể vừa tạo file, vừa ghi dữ liệu vào file và đọc dữ liệu từ file vào luôn.

## 7.2 Khai báo kiểu tệp tin

- Để định nghĩa một đối tượng file ta chọn các cách sau:

**fstream** *tên biến file*;

**ifstream** *tên biến file*;

**ofstream** *tên biến file*;

- Mở file:

*tên biến file*.**open**(*tên file*, chế độ mở file);

- Vừa khai báo vừa mở file:

**fstream/ofstream/ifstream** *tên biến file*(*tên file*, chế độ mở);

- Đóng file:

*tên biến file*.**close**();

## 7.2 Khai báo kiểu tệp tin

### Chế độ mở file:

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

### Mặc định:

ofstream	ios::out
ifstream	ios::in
fstream	ios::in   ios::out

*Chú ý: Mở một file đồng thời ở nhiều chế độ khác nhau, ta dùng toán tử hoặc “|”.*

## 7.2 Khai báo kiểu tệp tin

### Mở file văn bản

- Mở một file dưới chế độ văn bản:  
**fstream** *Tên biến file*(*Tên file*, ios::text);
- Khi đó các thao tác đọc, ghi trên biến file được thực hiện theo đơn vị là các từ, được phân cách bởi dấu trống (space bar) hoặc dấu xuống dòng (enter).

### Mở file nhị phân

- Mở một file dưới chế độ nhị phân:  
**fstream** *Tên biến file*(*Tên file*, ios::binary);
- Khi đó các thao tác đọc, ghi trên biến file được thực hiện theo đơn vị byte theo kích thước các bản ghi (cấu trúc) được ghi trong file.

## 7.3 Các thao tác truy xuất tệp tin

**Xuất dữ liệu ra file:** Để ghi dữ liệu ra file ta sử dụng toán tử xuất << như sau:

`DataFile << chuỗi_hoặc_biểu_thức_cần_ghi;`

Trong đó DataFile là tên đối tượng gắn với File mà ta cần ghi dữ liệu vào.

**Ví dụ:** Tạo và ghi dữ liệu vào file *teptin.txt* cùng thư mục với file .cpp trong project

```
#include<iostream>
#include<fstream>
using namespace std ;
int main(){
    int x=2025;
    ofstream FILE; //Tao doi tuong FILE de ghi du lieu vao file
    FILE.open( "teptin.txt"); //Gan file teptin.txt vao doi tuong file FILE
    FILE<< "Ky thuat lap trinh "<<x;
    FILE.close(); //Dong file hay giai phong file teptin.txt khoi doi tuong file FILE
}
```



## 7.3 Các thao tác truy xuất tệp tin

**Đọc file sử dụng toán tử >>** : Toán tử này sẽ đọc nội dung file từ vị trí con trỏ hiện tại đến khoảng trống đầu tiên.

**Ví dụ:** Đọc nội dung file info.ini

```
int main(){
    char data[55]; //Dung de luu du lieu tu file
    ifstream teptin;
    teptin.open( "info.ini");
    teptin>>data;
    cout<< "Tu dau tien cua file: "<<data;
    teptin.close();
    teptin.open( "info.ini");
    cout<< "\nNoi dung cua file: \n";
    while(teptin>>data)
        cout<<data<< " ";
    teptin.close();
    cout<<endl;
    system("pause");
}
```

Nội dung file info.ini

Ky thuat lap trinh  
Phan tep tin

Kết quả chạy chương trình

Tu dau tien cua file: Ky  
Noi dung cua file:  
Ky thuat lap trinh Phan tep tin

## 7.3 Các thao tác truy xuất tệp tin

**Đọc file bằng hàm getline:** Hàm này đọc dữ liệu trên 1 dòng.

**Ví dụ:** Đọc nội dung file info.ini

```
#include<iostream>
#include<fstream>
using namespace std ;
int main(){
    char data[99];
    fstream teptin("info.ini", ios::in);
    cout<< "Noi dung file la:\n";
    while(teptin.getline(data, 99))
        cout<<data<<endl;
    teptin.close();
    cout<<endl;
    system("pause");
}
```

Nội dung file info.ini

Ky thuat lap trinh  
Phan tep tin

Kết quả chạy chương trình

Noi dung file la:  
Ky thuat lap trinh  
Phan tep tin

## 7.4 Các hàm tệp tin

### Hàm xóa và đổi tên file

- Để xóa tệp tin ta dùng hàm `remove` như sau: ***remove*** (“đường dẫn tệp tin”);
- Để đổi tên tệp tin ta dùng hàm `rename` như sau: ***rename*** (“tên tệp tin cũ “, “tên tệp tin mới “);

**Ví dụ:** Đổi tên file **info.ini** ở ví dụ trên, sau đó xóa file khỏi ổ đĩa

```
int main(){
    char name[20]; int flag;
    cout<< "Ten file moi ban muon doi la: "; cin.getline(name, 20);
    rename( "info.ini", name);
    cout<< "Doi ten file thanh cong"<<endl;
    cout<< "Ban co muon xoa file "<<name<< " khoi o dia khong ? 0/1: " ;
    cin>>flag;
    if(flag==1) {
        remove(name);
        cout<< "Xoa file thanh cong";
    }
    system("pause");
}
```

## 7.4 Các hàm tệp tin

### Hàm đọc và ghi kí tự

- Hàm ghi kí tự vào file **put()**: `Đối_tượng_file.put(kí_tự);`
- Hàm đọc kí tự từ file **get()**: `Đối_tượng_file.get(kí_tự);`

**Ví dụ:** Ghi nội dung vào file `vidu.doc` và đọc nội dung của nó.

```
int main(){
    char kt;
    fstream File;    File.open("vidu.doc",ios::out);
    cout<<"Nhap noi dung cho file, de ket thuc an dau ^\n";
    cin.get(kt);
    while(kt != '^') {
        File.put(kt); cin.get(kt);
    }
    File.close();
    cout<<"Ghi du lieu thanh cong\n";
    cout<<"Noi dung cua file vidu.doc la:\n";
    File.open("vidu.doc",ios::in);
    while(File.get(kt))    cout<<kt;
    cout<<endl; system("pause");
}
```

```
Nhap noi dung cho file, de ket thuc an dau ^
Cong cha nhu nui Thai Son
Nghia me nhu nuoc trong nguon chay ra
^
Ghi du lieu thanh cong
Noi dung cua file vidu.doc la:
Cong cha nhu nui Thai Son
Nghia me nhu nuoc trong nguon chay ra
```

## 7.4 Các hàm tệp tin

### Hàm đọc và ghi file ở dạng nhị phân

Đối với file văn bản mỗi byte được xem là một kí tự. Kí tự có mã 26 được xem là kí tự kết thúc file.

Đối với file nhị phân, thông tin lưu trong file được xem như 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.

Một file luôn ngầm định dưới dạng văn bản, do vậy để chỉ định file là nhị phân ta cần sử dụng cờ ios::binary.

#### Hàm ghi file ở dạng nhị phân write():

**fileObject.write(address, size);**

- *fileObject* - Đối tượng file.
- *address* – Địa chỉ đầu tiên của 1 vùng nhớ được ghi vào file.
- *size* – Số lượng byte của vùng nhớ mà nó sẽ được ghi.

#### Hàm đọc file dạng nhị phân read()

**fileObject.read(address, size);**

- *fileObject* - Đối tượng file.
- *address* – Là địa chỉ đầu tiên mà vùng nhớ mà dữ liệu được đọc vào được lưu.
- *size* – Số lượng byte trong bộ nhớ được đọc vào từ file.

## 7.4 Các hàm tệp tin

Chuyển kiểu khi dùng write(), read()

- Tham số thứ nhất là con trỏ kiểu char trỏ đến vùng dữ liệu cần ghi vào file. Vì con trỏ bắt buộc có kiểu char nên khi muốn ghi dữ liệu có kiểu khác vào file, ta dùng hàm chuyển kiểu:

**reinterpret\_cast <char \*>(Dữ liệu);**

Hoặc **(char \*)(địa chỉ);**

- Tham số thứ hai là kích cỡ dữ liệu được ghi vào file. Kích cỡ này được tính theo byte, nên thông thường ta dùng toán tử:

**sizeof(Kiểu dữ liệu);**

### Chú ý:

- Phải sử dụng chế độ đọc/ghi file nhị phân khi muốn đọc, ghi các dữ liệu có cấu trúc (struct) vào file.
- Khi đọc/ghi dữ liệu kiểu cấu trúc, để toán tử sizeof() thực hiện chính xác thì các thành viên của cấu trúc không được là kiểu con trỏ.

## 7.4 Các hàm tệp tin

**Ví dụ 7.4.1:** Minh họa về ghi và đọc file ở dạng nhị phân.

```
int main(){
    char data[]= "Ky thuat lap trinh", data2[88],name[22];
    fstream file;
    cout<< "Nhap vao ten file: ";    cin.getline(name,22);
    file.open(name, ios::out | ios::binary);
    file.write(data, sizeof(data)); //Ham sizeof tra ve kich thuoc cua chuoai data
    file.close();
    file.open(name, ios::in | ios::binary);
    file.read(data2, sizeof(data2));
    cout << "Noi dung da ghi vao file "<<name<< " la:\n"<<data2;
    file.close();
    cout<<endl; system("pause");
}
```

Kết quả chạy thử chương trình

```
Nhap vao ten file: test01
Noi dung da ghi vao file test01 la:
Ky thuat lap trinh
```

### Ví dụ 7.4.2: Minh họa về ghi và đọc file ở dạng nhị phân với dữ liệu có cấu trúc.

```
#include<iostream>
#include<fstream>
using namespace std ;
struct product{
    int id;
    char name[20];
    float price;
};
int main(){
    fstream f("test.txt",ios::in|ios::out|ios::binary);
    int n; cout<<"Nhap so luong san pham: "; cin>>n;
    product p[n];
    for(int i=0;i<n;i++){
        cout<<"Nhap thong tin san pham thu "<<i+1<<endl;
        cout<<"Nhap id: "; cin>>p[i].id;
        cout<<"Nhap ten: "; cin.ignore(); cin.getline(p[i].name,20);
        cout<<"Nhap gia: "; cin>>p[i].price;
        f.write(reinterpret_cast<char*>(&p[i]),sizeof(product));
    }
    cout<<"thong tin cac san pham:"<<endl;
    cout<<"STT\tID\tTen\tGia"<<endl;
    for(int i=0;i<n;i++){
        f.read((char*)&p[i],sizeof(product));
        cout<<i+1<<"\t"; cout<<p[i].id<<"\t";
        cout<<p[i].name<<"\t"<<p[i].price<<endl;
    }
    f.close(); system("pause");
}
```

#### Kết quả chạy thử chương trình

```
Nhap so luong san pham: 4
Nhap thong tin san pham thu 1
Nhap id: 55
Nhap ten: Tieu
Nhap gia: 77000
Nhap thong tin san pham thu 2
Nhap id: 97
Nhap ten: Dieu
Nhap gia: 77000
Nhap thong tin san pham thu 3
Nhap id: 16
Nhap ten: Mang
Nhap gia: 33000
Nhap thong tin san pham thu 4
Nhap id: 87
Nhap ten: Khoai
Nhap gia: 32000
thong tin cac san pham:
STT      ID      Ten      Gia
1         55      Tieu     77000
2         97      Dieu     77000
3         16      Mang     33000
4         87      Khoai    32000
```



## 7.4 Các hàm tệp tin

Một số hàm trong thư viện `stdio.h`

Một số hàm xử lý chung

Tên hàm	Chức năng
<b>fopen</b>	Mở tệp tin
<b>fclose</b>	Đóng file
<b>fcloseall</b>	Đóng tất cả file đang mở
<b>fflush</b>	Làm sạch vùng đệm của một tệp tin đang mở
<b>fflushall</b>	Làm sạch vùng đệm của tất cả
<b>remove/unlink</b>	Xóa file
<b>feof</b>	Kiểm tra đã đến cuối file chưa

## 7.4 Các hàm tệp tin

Một số hàm trong thư viện `stdio.h`

Một số hàm xử lý tệp tin văn bản

Tên hàm	Chức năng
<b>fprintf</b>	Ghi giá trị dạng text lên tệp tin
<b>fscanf</b>	Đọc giá trị dạng text từ tệp tin
<b>putc/fputc</b>	Ghi lên tệp tin một kí tự (sử dụng tệp tin văn bản hoặc nhị phân có khác biệt )
<b>getc/fgetc</b>	Đọc từ tệp tin một kí tự (sử dụng tệp tin văn bản hoặc nhị phân có khác biệt)
<b>fputs</b>	Ghi một chuỗi vào tệp tin
<b>fgets</b>	Đọc một chuỗi từ tệp tin

## 7.4 Các hàm tệp tin

Một số hàm trong thư viện `stdio.h`

Một số hàm xử lý tệp tin nhị phân

Tên hàm	Chức năng
<b>putw</b>	Ghi một số nguyên lên tệp tin
<b>getw</b>	Đọc một số nguyên từ tệp tin
<b>fwrite</b>	Ghi các mẫu tin (có cấu trúc định trước) lên tệp tin
<b>fread</b>	Đọc các mẫu tin (có cấu trúc định trước) từ tệp tin
<b>fseek</b>	Di chuyển con trỏ đến vị trí mong muốn
<b>ftell</b>	Cho biết vị trí hiện tại của con trỏ

## 7.5 Con trỏ tệp tin

Các phương thức sau cho phép làm việc trên đối tượng của dòng xuất (**ofstream**).

- **Đối\_tượng.seekp(n);** Di chuyển con trỏ đến byte thứ n (các byte được tính từ 0)
- **Đối\_tượng.seekp(n, vị trí xuất phát);** Di chuyển đi n byte (có thể âm hoặc dương) từ vị trí xuất phát. Vị trí xuất phát gồm:
  - ios::beg : từ đầu file
  - ios::end : từ cuối file
  - ios::cur : từ vị trí hiện tại của con trỏ.
- **Đối\_tượng.tellp(n);** Cho biết vị trí hiện tại của con trỏ.

Để làm việc với dòng nhập (ifstream) thì tên các phương thức trên được thay tương ứng bởi các tên : **seekg** và **tellg**.

Đối với các dòng nhập lẫn xuất có thể sử dụng được cả 6 phương thức trên.

## 7.5 Con trỏ tệp tin

**Ví dụ:** Đếm ký tự trong file

```
#include<iostream>
#include<fstream>
using namespace std ;
int main(){
    ofstream f;
    f.open( "demkitu.txt");
    f<< "Ky thuat lap trinh\nSu dung ngon ngu C++";
    f.seekp(0, ios::end); //Di chuyển con trỏ tới cuối file
    cout << "So ki tu co trong file la: " << f.tellp();
    f.close();
    cout<<endl; system("pause");
}
```

Kết quả chạy thử chương trình


So ki tu co trong file la: 40

## 7.5 Con tr  t p tin

**V  dụ:** Thay th  n i dung c a file

```
#include<iostream>
#include<fstream>
using namespace std ;
int main(){
    ofstream f("thaythe.txt");
    f<<"Noi dung 1\nNoi dung 2\n";
    f.seekp(0, ios::beg);
    f<< "Noi dung 3";
    f.close();
    cout<<endl;
    system("pause");
}
```

N i dung trong file thaythe.txt

 thaythe.txt  
Noi dung 3  
Noi dung 2

# BÀI TẬP CHƯƠNG 7

**Bài 1.** Nhập một dãy số gồm  $n$  phần tử số nguyên rồi lưu vào tệp tin 1.txt

- a) Sắp xếp các số trong tệp 1.txt và lưu kết quả vào tệp 2.txt
- b) Xóa tệp 1.txt và đổi tên tệp 2.txt thành sort.txt
- c) Thêm vào cuối tệp sort.txt số 2025

**Bài 2.** Tạo file tho.doc để ghi dữ liệu 1 bài thơ

- a) Hãy đọc nội dung bài thơ trong file trên
- b) Câu thơ nào có nhiều kí tự nhất

**Bài 3.** Viết chương trình để tổ chức và quản lý file sinh viên (Họ tên, ngày sinh, giới tính, điểm các môn) với các chức năng : Nhập, xem, xóa, sửa.

**Bài 4.** Sử dụng file viết chương trình quản lý người dùng với các chức năng

- Đăng kí tài khoản
- Đăng nhập (nếu chưa có tài khoản thì cho phép đăng kí mới)
- Nếu đăng nhập thành công: Người dùng có thể đổi mật khẩu mới (cần nhập mật khẩu cũ để xác nhận đổi).