**CÁC THAO TÁC TRÊN TẬP TIN**

1. Mở đóng tập tin.

* Để mở một tập tin dùng hàm (phương thức) open

Fstream myFile;

MyFile.open(filePath,mod);

\_filePath: đường dẫn file

\_mod: chế độ mở file

* Các mod trong C++

\_ios::in:mở file để đọc

\_ios::out: mở file có sẳn để ghi

\_ios::binary: mở file ở chế độ nhị phân.

\_ios::ate: mở file và đặt con trỏ file vào cuối file.

\_ios::app:mở file và ghi dữ liệu vào cuối file. Nếu file không tồn tại thì mở file mới.

\_ios\_trunc:mở file, xóa bỏ nội dung trong file vừa mở

\_ví dụ:

#include<iostream>

#include<fstream>

using namespace std;

void main(){

fstream myFile;

myFile.open("D://C++//include.docx", ios::in | ios::binary);

bool flag1 = myFile.fail();

if (flag1 == true)

cout << "File khong ton tai" << endl;

else

cout << "Mo file thanh cong" << endl;

system("pause");

}

1. Đóng file

* Sau khi thao tác với file xong, chúng sẽ đóng file bằng lệnh

\_f.close()

\_việc đóng file sẽ giúp chúng ta bảo toàn được dữ liệu đang tiến hành lưu dữ và tránh khỏi một số lỗi không đáng có

1. Đọc và ghi file nhị phân

* Nếu mở file mà không ghi rõ chế độ mở file, trình biên dịch sẽ chuyển về chế độ mở file text.
* Nếu muốn mở ở chế độ binary, ta cần phải ghi rõ ràng mod là ios::binary.
* Nếu mở file dưới chế độ binary sẽ phải đọc file dưới dạng binary.
* Sử dụng hàm write và read để đọc và ghi file dưới dạng binary.
* Đọc dữ liệu từ file

\_myFile.read(address,size);

* Address: kiểu dữ liệu char\*, là địa chỉ mà dữ liệu sẽ lưu vào khi đọc lên.
* Size: kiểu dữ liệu int, là số lượng byte trong bộ nhớ được đọc từ file
* Ghi dữ liệu vào file

\_myFile.write(address,size);

* Address: kiểu dữ liệu char\*, là địa chỉ vùng nhớ của dữ liệu được lưu vào file.
* Size: kiểu dữ liệu int, là số byte vùng nhớ mà nó sẽ ghi

Ví dụ: #include<iostream>

#include<fstream>

using namespace std;

void main(){

fstream myFile;

char a = 'a', b;

//Mở file có sẳn để ghi

myFile.open("D://C++//include.dox", ios::out | ios::binary);

cout << "Ghi du lieu vao file: " << endl;

myFile.write(&a, sizeof(a));

myFile.close();

//Mở file để đọc

myFile.open("D://C++//include.dox", ios::in | ios::binary);

cout << "doc du lieu cua bo nho: " << endl;

myFile.read(&b, sizeof(b));

cout << b << " ";

myFile.close();

system("pause");

}

* Nếu chúng ta muốn ghi các kiểu khác vào file nhị phân thì ta phải dùng cú pháo sau đây:

Reinterpret\_cat<dataType>(value)

\_dataType sẽ là kiểu dữ liệu mà chúng ta muốn ép về

\_value sẽ là giá trị mà chúng ta muốn ép

\_ví dụ:

Int x=1;

File.write(reinterpret\_cat<char\*>(&x),sizeof(x));

Ví dụ:

#include<iostream>

#include<fstream>

using namespace std;

void main(){

fstream myFile;

int a[10] = { 1,2,3,4,5,6,7,8,9,10 };

//Mở file có sẳn để ghi

myFile.open("D://C++//include.dox", ios::out, ios::binary);

//ghi dữ liệu từ mảng vào file

myFile.write(reinterpret\_cast<char\*>(a), sizeof(a));

//đóng file

myFile.close();

//mở file để đọc

myFile.open("D://C++//include.dox", ios::in);

int b[10];

//đọc nội dung file vào mảng b

cout << "Doc du lieu vao bo nho: " << endl;

myFile.read(reinterpret\_cast<char\*>(b), sizeof(b));

//xuất dữ liệu

for (int i = 0; i < 10; i++) {

cout << b[i] << " ";

}

myFile.close();

system("pause");

}

1. Đọc và ghi file văn bản

* Đọc dữ liệu từ file

\_f>>data;

* f là tên biến file.
* Data là tên biến lưu dữ liệu. có thể có các kiểu như int, float,long,char…
* Khi dữ liệu được đọc có kiểu khác với biến dữ liệu được lưu trữ.
* Biến lưu trữ dữ liệu kiểu số không đọc vào kiểu ký tự.
* Biến lưu trữ dữ liệu kiểu số được tự động chuyển qua lại giữa các kiểu dữ liệu(float sang int,int sang float) bằng cơ chế chuyển kiểu ngầm định.
* Ghi dữ liệu vào file

\_f<<data;

* f: là tên biến file.
* Data: biến dữ liệu chứa data cần ghi vào file. Có thể có các kiểu dữ liệu như int,float,long,char…
* Thư viện fstream hỗ trợ ghi dữ liệu char,int,long…(nhóm dữ liệu primitive). Tự nhiên lại không hỗ trợ ghi dữ liệu mở rộng(struct…)

\_Ngoài ra chúng ta có thể sử dụng hàm get và hàm getline để lấy thông tin từ file

* Kiểm tra kết thúc file
* Kiểm tra kết thúc file

\_f.eof() sẽ kiểm tra xem đã kết thúc file chưa?

* Ví dụ: đọc dữ liệu từ file
* #include<iostream>
* #include<fstream>
* using namespace std;
* void main(){
* fstream myFile;
* myFile.open("D://C++//comchim.txt", ios::in);
* char a[100];
* while (!myFile.eof()) {
* myFile>> a;
* cout << a << " ";
* }
* myFile.close();
* system("pause");
* }

Ví dụ: Ghi dữ liệu vào File

#include<iostream>

#include<fstream>

using namespace std;

void main(){

fstream myFile;

myFile.open("D://C++//comchim.txt",ios::out);

char a[100] = "xin chao cac mat l";

myFile << a;

myFile.close();

system("pause");

}

1. Các hàm định vị file

* Các hàm đặt vị trí con trỏ:

\_seekg (đặt vị trí đọc cho lớp istream)

\_seekp (đặt vị trí ghi cho ostream)

* Cột mốc (ios::beg mặc định)

\_ios::beg – đầu file

\_ios::cur –vị trí hiện tại

\_ios::end – cuối file

* Các hàm lấy vị trí hiện tại con trỏ

\_tellg và tellp

Ví dụ:

#include<iostream>

#include<fstream>

using namespace std;

void main(){

fstream myFile;

myFile.open("D://C++//comchim.txt",ios::in);

char a[100];

//bắt đầu từ byte số 2 chạy

myFile.seekg(2);

while (!myFile.eof()) {

//từ vị trí hiện tại

myFile.seekg(2, ios::cur);

myFile >> a;

cout << "Gia tri: " << a;

cout << endl;

cout << "vi tri: " << myFile.tellg();

cout << endl;

}

myFile.close();

system("pause");

}