

HƯỚNG DẪN THỰC HÀNH

THAO TÁC VỚI TẬP TIN

1 Tập tin và các thao tác cơ bản

1.1 Sơ lược tập tin

- Là hình thức lưu trữ phổ biến trên bộ nhớ phụ.
- Gồm 2 loại: tập tin văn bản và tập tin nhị phân.

Tập tin văn bản	Tập tin nhị phân
<ul style="list-style-type: none"> ✓ Dữ liệu được lưu dưới dạng chuỗi các kí tự ✓ Người dùng có thể đọc được. <p>VD (C++): Lưu số nguyên 12345</p> <p>=> lưu chuỗi gồm 5 kí tự liên tiếp nhau: 12345, tốn 5 bytes.</p>	<ul style="list-style-type: none"> ✓ Dữ liệu lưu dưới dạng chuỗi bit 0/1. ✓ Thường người dùng không đọc được. <p>VD (C++): Lưu số nguyên 12345, biết số nguyên có kích thước 2 bytes và</p> $(12345)_{10} = (00110000\ 00111001)_2$ <p>=> lưu chuỗi bit nhị phân tương ứng ở trên, tốn 2 bytes.</p> <p><i>(SV nhận xét trong trường hợp lưu chuỗi kí tự dạng nhị phân?)</i></p>
<ul style="list-style-type: none"> ✓ Tính chất <ul style="list-style-type: none"> ○ Dễ truy xuất và xử lý ○ Độ bảo mật kém ○ Tốc độ truy xuất kém ○ Kích thước lớn ✓ Các dạng thường gặp <ul style="list-style-type: none"> ○ Tập tin cấu hình: *.INI, *.CFG ○ *.TXT, *.HTML, *.XML 	<ul style="list-style-type: none"> ✓ Tính chất <ul style="list-style-type: none"> ○ Truy xuất và xử lý phức tạp ○ Tốc độ truy xuất nhanh hơn ○ Tính bảo mật cao ✓ Các dạng thường gặp <ul style="list-style-type: none"> ○ *.DOC, *.XLS, *.PDF, *.PPT, ... ○ *.JPG, *.PCX, *.BMP, *.TIF, ... ○ *.MP3, *.WAV, *.AVI, ...

1.2 Các thao tác cơ bản trên tập tin

- Ngôn ngữ sử dụng: C++.
- Đối với C++, tài liệu sử dụng các lớp hỗ trợ trong bộ thư viện STL.

Khai báo sử dụng

C++
<pre>#include <fstream> using namespace std; int _tmain(int argc, _TCHAR* argv[]) { // Khai báo sử dụng đối tượng //để thao tác với tập tin fstream f; // ... return 0; }</pre> <p><i>Lưu ý: <fstream> gồm các phương thức đọc và ghi tập tin. Ngoài ra, SV có thể sử dụng</i> <i>○ <ifstream> chỉ đọc tập tin ○ <ofstream> chỉ ghi tập tin.</i></p>

Mở tập tin

C++
Mô tả
<pre>void open(const char* tenFile, ios_base::openmode tuyChonMo)</pre> <p>Trong đó:</p> <ul style="list-style-type: none"> ✓ tenFile: đường dẫn đến tập tin cần thao tác ✓ tuyChonMo: cho biết thao tác tùy chọn, bao gồm các giá trị sau: <ul style="list-style-type: none"> ○ ios_base::in: mở để đọc ○ ios_base::out: mở để ghi ○ ios_base::app: ghi vào cuối ○ ios_base::ate: con trỏ đọc nằm ở cuối tập tin khi mở ○ ios_base::binary: xem tập tin ở dạng nhị phân. Mặc định là văn bản.
Ví dụ

```
int _tmain(int argc, _TCHAR* argv[])
{
    // Khai báo sử dụng đối tượng //để thao tác tập tin
    fstream f;

    // Mở tập tin input.txt dạng //binary để đọc
    f.open("input.txt", ios_base::in | ios_base::binary);
    // ...
    return 0;
}
```

Đóng tập tin

C++
Mô tả
void close()
Ví dụ
<pre>int _tmain(int argc, _TCHAR* argv[]) { // Khai báo sử dụng đối tượng //để thao tác tập tin fstream f; // Mở tập tin input.txt dạng //binary để đọc f.open("input.txt", ios_base::in ios_base::binary); // Thực hiện thao tác //.... // Đóng tập tin f.close(); return 0; }</pre>

Ghi tập tin

C++
Mô tả
<p>✓ <u>Tập tin văn bản</u></p> <ul style="list-style-type: none"> ○ Sử dụng toán tử "<<".

- Ngoài ra, còn phương thức khác như **put** (SV tự tìm hiểu)

✓ Tập tin nhị phân

```
ostream& write(const char* s, streamsize n)
```

>> Ghi một block dữ liệu **n** bytes vào tập tin. Trong đó,

- **s**: con trỏ đến block dữ liệu cần ghi
- **n**: kích thước của block dữ liệu

Ví dụ

```

// Tập tin văn bản
int _tmain(int argc, _TCHAR* argv[])
{
    // Khai báo sử dụng đối tượng //để thao tác tập tin
    fstream f;

    // Mở tập tin input.txt dạng //văn bản để đọc
    f.open("input.txt", ios_base::out);

    // Ghi 1 số nguyên và một chuỗi //kí tự vào tập tin
    int n=10;
    char line[256] = "Hom qua toi di hoc";

    f << n << endl;
    f << line;

    // Đóng tập tin
    f.close();

    return 0;
}

// Tập tin nhị phân
int _tmain(int argc, _TCHAR* argv[])
{
    // Khai báo sử dụng đối tượng //để thao tác tập tin
    fstream f;

    // Mở tập tin input.txt dạng //binary để ghi
    f.open("input.txt", ios_base::out | ios_base::binary);
    int n=10;
    char line[256] = "Hom qua toi di hoc";
    //Ghi số nguyên n, sizeof(int) //dùng để lấy số byte lưu 1 số nguyên
    f.write((const char*)&n, sizeof(int));

    // Ghi chuỗi line, hàm strlen //dùng để lấy chiều dài của chuỗi.
    f.write(line, strlen(line));
    // Đóng tập tin
    f.close();
    return 0;
}

```

Lưu ý: Khi thao tác với tập tin nhị phân, C++ cho phép ghi một hoặc nhiều mẫu tin (record). Mẫu tin có kiểu dữ liệu được định nghĩa bằng struct, bao gồm nhiều trường dữ liệu khác nhau. Xét ví dụ sau:

CDiem là kiểu dữ liệu biểu diễn cho điểm trong không gian 2 chiều Oxy, định nghĩa như sau:

```
typedef struct CDiem
{
    int x;
    int y;
};
```

Dưới đây là ví dụ để ghi một hoặc nhiều điểm vào tập tin nhị phân output.txt;

```
int _tmain(int argc, _TCHAR* argv[])
{
    // Tạo điểm A(1, 3)
    CDiem A;
    A.x = 1;
    A.y = 3;

    // Tạo 5 điểm có tung độ gấp đôi hoành độ
    int n = 5;
    CDiem* B = new CDiem[n];
    for(int i = 0; i < n; i++)
    {
        B[i].x = i;
        B[i].y = i * 2;
    }

    // Ghi tập tin dạng nhị phân
    fstream f;
    f.open("output.txt", ios_base::out | ios_base::binary);
    // Ghi điểm A vào tập tin, sizeof(CDiem) để lấy kích thước 1 điểm.
    f.write((const char*)&A, sizeof(CDiem));
    // Ghi 5 điểm B
    f.write((const char*)B, n * sizeof(CDiem));

    f.close();

    // Giải phóng vùng nhớ cho con trỏ B
    delete B;
    return 0;
}
```

Đọc tập tin

C++
Mô tả
<p>✓ <u>Tập tin văn bản</u></p> <p>✓ ○ Sử dụng toán tử “>>”.</p> <p>○ Dừng đọc khi gặp khoảng trắng hoặc không có kí tự nào.</p> <p>○ Ngoài ra, còn một số phương thức khác như: get, getline (tự tìm hiểu).</p> <p>✓ <u>Tập tin nhị phân</u></p> <p><code>istream& read(char* s, streamsize n)</code></p> <p>>> Đọc một block dữ liệu có kích thước n bytes từ tập tin. Trong đó,</p> <ul style="list-style-type: none"> ○ s : con trỏ tới vùng nhớ lưu dữ liệu đọc được ○ n: số lượng bytes cần đọc.
Ví dụ
<pre>//Tập tin văn bản int _tmain(int argc, _TCHAR* argv[]) { // Khai báo sử dụng đối tượng //để thao tác tập tin fstream f; // Mở tập tin input.txt dạng //text để đọc f.open("input.txt", ios_base::in); // Đọc dữ liệu từ tập tin, gồm //1 số nguyên và một chuỗi kí tự liên nhau. int n; char line[256]; f >> n; f >> line; //Đóng tập tin f.close(); return 0; }</pre>

```
// Tập tin nhị phân
int _tmain(int argc, _TCHAR* argv[])
{
    // Khai báo sử dụng đối tượng //để thao tác tập tin   fstream f;
    // Mở tập tin input.txt dạng //binary để đọc
    f.open("input.txt", ios_base::in | ios_base::binary);
    // Đọc dữ liệu từ tập tin, gồm //1 số nguyên và một chuỗi kí tự
    int n=0;
    char line[50];

    f.read((char*)&n, 4);
    f.read(line, 50);

    // Đóng tập tin
    f.close();

    return 0;
}
```

Lưu ý: Tương tự như ghi, C++ cho phép đọc một hoặc nhiều mẫu tin (record). Xét ví dụ đọc tập tin chứa các điểm đã ghi trong ví dụ “Ghi tập tin”, ta có thể viết chương trình đọc như sau:

```
int _tmain(int argc, _TCHAR* argv[])
{
    // Khai báo điểm A
    CDiem A;

    // Khai báo mảng B, có thể chứa 5 điểm
    int n = 5;
    CDiem* B = new CDiem[n];

    // Mở tập tin dạng binary
    fstream f;
    f.open("output.txt", ios_base::in | ios_base::binary);
    // Đọc điểm A.
    f.read((char*)&A, sizeof(CDiem));
    // Đọc danh sách 5 điểm B.
    f.read((char*)B, n * sizeof(CDiem));

    // Hiển thị kết quả ra màn hình
    cout << "Diem A(" << A.x << ", " << A.y << ")" << endl;
    cout << "Danh sach Diem" << endl;
    for(int i = 0; i < n; i++)
    {
```



```

        cout << "(" << B[i].x << ", " << B[i].y << "), ";
    }

    f.close();

    // Giải phóng vùng nhớ cho con trỏ B
    delete B;
    return 0;
}

```

1.3 Một số phương thức khác

Với ngôn ngữ C++

Phương thức	Mô tả
is_open()	Kiểm tra tập tin có đang mở hay không. Trả về true nếu có
tellg()	Lấy vị trí kế tiếp sẽ được đọc trong tập tin. Trả về 1 nếu lỗi
tellp()	Lấy vị trí mà dữ liệu kế tiếp sẽ được ghi vào tập tin. Trả về -1 nếu lỗi
istream& seekg(streampos viTri) istream& seekg(streamoff: offset, ios_base::seek_dir tuyChon)	Di chuyển con trỏ đọc đến vị trí mong muốn. Di chuyển con trỏ đọc đến vị trí cách vị trí <tuyChon> một khoảng là offset byte, có thể nhận một trong các giá trị sau: <ul style="list-style-type: none"> ○ ios_base::beg : bắt đầu tập tin ○ ios_base::cur : vị trí hiện hành ○ ios_base::end : cuối tập tin
seekp	Di chuyển con trỏ ghi đến vị trí mong muốn
peek	Đọc kí tự kế tiếp nhưng không di chuyển con trỏ đọc tập tin
flush	Đồng bộ vùng đệm với tập tin, nhằm đảm bảo các nội dung trong vùng đệm được ghi vào tập tin.
eof	Kiểm tra con trỏ đọc/ghi đến cuối tập tin hay chưa.

2 Các thuật toán trên tập tin nhị phân

2.1 Tính kích thước

C++

```
int _tmain(int argc, _TCHAR* argv[])
{
    // Khai báo sử dụng đối tượng //để thao tác tập tin
    fstream f;

    // Mở tập tin input.txt dạng //binary để ghi
    f.open("input.txt", ios_base::in | ios_base::binary);
    int length;
    // Di chuyển đến vị trí cuối
    f.seekg (0, ios::end);

    // Lấy vị trí của con trỏ tập tin
    length = f.tellg();
    // Trở về vị trí ban đầu
    f.seekg (0, ios::beg);

    // Đóng tập tin
    f.close();

    return 0;
}
```

2.2 Cắt, ghép

Kích thước tập tin quá lớn có thể gây trở ngại trong việc chia sẻ như upload/download... Việc chia tập tin thành n tập tin nhỏ hơn khi chia sẻ và ghép trở lại khi sử dụng là một giải pháp. Chương trình cắt, ghép tập tin có thể thực hiện như sau:

✓ Cắt nhỏ tập tin:

- Tính kích thước tập tin ban đầu.
- Xác định kích thước cho n tập tin nhỏ hơn
- Tạo n tập tin mới và ghi từng block nội dung vào các tập tin đó. Tên các tập tin nên đặt theo qui tắc để thuận tiện trong việc ghép về sau. Ví dụ: <ten_file>.001, <ten_file>.002....
- Đóng tất cả các tập tin lại.

✓ Ghép tập tin.

- Tạo mới tập tin, sẽ chứa nội dung ghép lại
- Lần lượt mở từng tập tin con và ghi nội dung lần lượt vào tập tin mới.

3 Bài tập làm thêm

Bài 1. Viết chương trình phát sinh ngẫu nhiên 10000 số tự nhiên và ghi kết quả xuống 2 tập tin: một ở dạng văn bản và một ở dạng nhị phân. Lưu ý các số cách nhau bởi khoảng trắng. Thực hiện so sánh kích thước của 2 tập tin.

Bài 2. Cho trước một tập tin lưu các số nguyên. Viết chương trình đọc và hiển thị các số nguyên trên màn hình. Để tiện theo dõi, hãy hiển thị các số nguyên này trên nhiều dòng, mỗi dòng gồm 15 số.

Bài 3. Viết chương trình soạn thảo văn bản, cho phép người dùng nhập vào một văn bản. Chương trình sẽ kết thúc nếu người dùng nhập vào chuỗi “end”, đồng thời lưu toàn bộ nội dung của văn bản xuống tập tin có tên là **MSSV.txt**.

Chẳng hạn, người dùng nhập văn bản như sau:

```
Chan troi rat xanh goi nang xon xao <Enter>
Con thuyen rat vui va gio ngot ngao <Enter>
Moi cuoi rat xinh lung linh mau ao <Enter>
end <Enter> ← Kết thúc ở đây.
```

Nội dung tập tin **<MSSV>.txt** sẽ là:

```
Chan troi rat xanh goi nang xon xao
Con thuyen rat vui va gio ngot ngao
Moi cuoi rat xinh lung linh mau ao
```

Bài 4. Viết chương trình đọc nội dung của tập tin đã tạo ở bài c) và hiển thị các kết quả sau lên màn hình.

- Nội dung của tập tin
- Thống kê số lần xuất hiện của các kí tự. Lưu ý không phân biệt chữ hoa và chữ thường.

Bài 5. Định nghĩa cấu trúc cho đối tượng sinh viên gồm các thông tin sau: **<MSSV>**, **<Họ tên>**. Hãy viết chương trình cho phép nhập và tra cứu thông tin của sinh viên như sau:

- Yêu cầu người dùng chọn chức năng: 1 nếu nhập thông tin, 2 nếu muốn tra cứu và 0 nếu thoát.
- **Về việc nhập thông tin:** yêu cầu người dùng nhập MSSV và họ tên của sinh viên. Sau đó ghi thêm sinh viên này vào một tập tin nhị phân có sẵn

- **Về việc tra cứu:** yêu cầu người dùng nhập số thứ tự. Sau đó hiển thị thông tin tương ứng của sinh viên đó ra màn hình. Nếu không tìm thấy thì báo là không tồn tại.

Bài 6. Thực hiện cài đặt chương trình cho phép cắt, ghép tập tin bất kỳ với giải thuật nêu ở trên.

Bài 7. An có một tập tin muốn chia sẻ cho bạn của mình nhưng không muốn người khác đọc được. Vì thế, An quyết định xáo trộn nội dung của tập tin theo một qui tắc nhất định và chỉ nói qui tắc cho người bạn đó biết. Hãy giúp An viết chương trình thực hiện công việc đó và giúp người bạn của An phục hồi lại nội dung ban đầu. Biết rằng qui tắc xáo trộn được định nghĩa như sau:

- Chia nội dung thành các block 4 bytes liên tiếp nhau.
- Sau đó đổi chỗ 2 block liên tiếp cho nhau, nghĩa là block 1 sẽ đổi chỗ cho block 2, block 3 đổi chỗ cho block 4....

Bài 8. Tương tự như yêu cầu bài 6, nhưng thay vì đổi chỗ các block dữ liệu, An sẽ thực hiện như sau:

- Chọn một số n làm key, $n \in [1, 64]$
- Thực hiện chuyển đổi từng kí tự (khác khoảng trắng và tab) trong tập tin thành một kí tự khác cách nó $n - 1$ kí tự trong bảng mã ASCII về phía tay phải. Nếu kí tự cần tìm vượt khỏi giới hạn của bảng mã thì quay về đầu để tìm tiếp.

Ví dụ: cho một văn bản có nội dung như sau:

Hom qua toi di hoc ~.

Chọn $n = 3$ thì tập tin kết quả sau khi chuyển đổi sẽ như sau:

Krp txd wrl gl krf (SOH)1

Trong đó (SOH) nghĩa là start of heading, là kí tự có mã là 1 trong bảng mã ASCII.