



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN: **KĨ THUẬT LẬP TRÌNH**

HƯỚNG DẪN THỰC HÀNH

TUẦN 2

KĨ THUẬT XỬ LÝ TẬP TIN

TP.HCM, ngày 10 tháng 05 năm 2020

MỤC LỤC

1	Giới thiệu	3
2	Đọc từng dòng từ tập tin.....	3
3	Đọc tập tin chứa các phân tử kiểu PHANSO	4
4	Ghi một dòng dữ liệu vào tập tin.....	5
5	Tạo tập tin văn bản từ bàn phím.....	6
6	Thay đổi nội dung một tập tin và chép vào tập tin mới.....	6
7	Đọc và ghi dữ liệu trên cùng một tập tin	7
8	Bài tập.....	8

1 Giới thiệu

Tập tin là một trong những đối tượng ta thường xuyên gặp khi lập trình. Chúng ta có thể có nhu cầu đọc nội dung của một tập tin văn bản nào đó hoặc ghi một nội dung ra tập tin để lưu trữ sau khi tính toán. Trong phần này chúng ta sẽ tìm hiểu các thức xử lý bao gồm đọc, ghi nội dung tập tin thông qua các ví dụ cụ thể.

2 Đọc từng dòng từ tập tin

Trong ví dụ này ta sẽ có một tập tin văn bản, yêu cầu cần phải xuất ra màn hình nội dung từng dòng văn bản này. Nói một cách đơn giản là ta lấy nội dung từ tập tin (từ ổ cứng) xử lý xuất ra màn hình.

Dòng	
1	<code>#include <fstream></code>
2	<code>#include <iostream></code>
3	<code>using namespace std;</code>
4	
5	<code>void TransferLines(istream& inDev, ostream& outDev){</code>
6	<code> const int maxSize = 255; char buf[maxSize + 1];</code>
7	<code> while(inDev.getline(buf, maxSize)){</code>
8	<code> outDev << buf << endl;</code>
9	<code> }</code>
10	<code>}</code>
11	
12	<code>void TransferData(istream& inDev, ostream& outDev){</code>
13	<code> char data;</code>
14	<code> while(inDev.get(data)){ outDev << data;}</code>
15	<code>}</code>
16	
17	<code>void main(){</code>
18	<code> ifstream fs("input.inp");</code>
19	<code> if(!fs){ cout << "Khong the mo file" << endl;}</code>
20	<code> TransferLines(fs, cout); // Có thể thế bằng TransferData</code>
21	<code>}</code>

Ta sẽ giải thích từ hàm main: trong hàm này ta khai báo một biến tên là fs kiểu ifstream. Lưu ý, ta sẽ đính kèm thêm tên tập tin (trong trường hợp này là input.inp, để đơn giản ta tạo tập tin này với nội dung ngẫu nhiên và lưu trong thư mục Debug của Project). Sau đó ta kiểm tra việc mở tập tin có thành công hay không bằng câu lệnh if. Cuối cùng ta sử dụng hàm TransferLines với hai tham số có kiểu lần lượt là istream và ostream. Tham số

đầu tiên của ta là fs, mặc dù kiểu của fs là fstream, tuy nhiên kiểu istream tương thích với kiểu fstream nên ta hoàn toàn có thể gán biến fs vào vị trí này (Có thể hiểu như việc ép kiểu). Còn biến cout có kiểu là ostream và là một biến mà hệ thống **đã có sẵn**, ta chỉ việc sử dụng.

Trong hàm TransferLines ta sẽ khai báo một bộ đệm buf chứa 256 kí tự (mảng kiểu char). Sau đó ta dùng vòng lặp while để thực hiện việc lấy từng dòng từ tập tin (biến inDev đại diện, inDev thực chất là biến fs) và xuất ra màn hình (biến outDev đại diện, outDev thực chất là biến cout). Ta thấy hàm getline sẽ lần lượt lấy từng dòng trong tập tin và cho vào biến buf, sau đó hàm này trả ra chính đối tượng fs. Khi không còn dòng nào trong tập tin thì fs = NULL và thoát ra khỏi vòng lặp.

Hàm TransferData sinh viên tự nghiên cứu ý nghĩa.

3 Đọc tập tin chứa các phân tử kiểu PHANSO

Giả sử chúng ta có tập tin chứa các phân tử kiểu PHANSO. Dòng đầu tiên của tập tin chứa số lượng phân tử. Các dòng tiếp theo mỗi dòng tương ứng với một PHANSO, trong đó tử số và mẫu số cách nhau bởi khoảng trắng. Ta sẽ thực hiện đọc nội dung tập tin này vào một mảng PHANSO để xử lý.

Dòng	
1	<code>#include <fstream></code>
2	<code>#include <iostream></code>
3	<code>using namespace std;</code>
4	
5	<code>struct PHANSO{</code>
6	<code>int tu, mau;</code>
7	<code>};</code>
8	
9	<code>istream& operator>>(istream& inDev, PHANSO& p){</code>
10	<code>inDev >> p.tu >> p.mau;</code>
11	<code>return inDev;</code>
12	<code>}</code>
13	
14	<code>PHANSO* NhapMangPhanSo(istream& inDev, int& n){</code>
15	<code>PHANSO* arr = NULL; inDev >> n;</code>
16	<code>if(inDev.fail() n <= 0){return arr;}</code>
17	<code>arr = new PHANSO[n];</code>
18	<code>if(arr != NULL){</code>
19	<code>for(int i = 0 ; i < n ; i++){</code>

20	inDev >> arr[i];
21	if(inDev.fail()){
22	delete[] arr; arr = NULL; break;
23	}
24	}
25	}
26	return arr;
27	}
28	
29	void main(){
30	ifstream fs("data.inp");
31	if(fs.fail()){cout << "Khong the mo tap tin" << endl;}
32	int n;
33	PHANSO* arr = NhapMangPhanSo(fs, n);
34	}

4 Ghi một dòng dữ liệu vào tập tin

Trong ví dụ này ta có nhu cầu ghi một chuỗi dữ liệu (từ RAM) vào tập tin (ổ cứng)

Dòng	
0	#include <fstream>
1	#include <iostream>
2	using namespace std;
3	
4	void main(){
5	ofstream os("hello.txt", ios::app); //mở tập tin
6	if(!os){cout << "Khong the mo tap tin" << endl;}
7	char* str = "Hello World";
8	while(*str){
9	os.put(*str);
10	str++;
11	}
12	}

Dòng đầu tiên ta khai báo biến os kiểu ofstream với tham số đính kèm là tên tập tin và chế độ ghi là gì. Chế độ ios::app là mở tập tin và chèn vào cuối tập tin, nếu tập tin chưa tồn tại sẽ tạo mới. Các dòng còn lại đáng chú ý chỉ có dòng 9. Ta dùng hàm put thuộc đối tượng os kiểu ofstream để thực hiện lấy ký tự vào tập tin.

5 Tạo tập tin văn bản từ bàn phím

Ta có nhu cầu tạo ứng dụng cho người dùng gõ nội dung từ bàn phím, sau đó lưu vào tập tin (Giống như lệnh COPY CON trong DOS). Lưu ý: người dùng sẽ kết thúc việc nhập bằng tổ hợp phím Ctrl + Z, sau đó nhấn Enter.

Dòng	
0	<code>#include <fstream></code>
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	
4	<code>void main(){</code>
5	<code> ofstream os("data.txt");</code>
6	<code> if(!os){cout << "Khong the mo tap tin" << endl;}</code>
7	<code> TransferData(cin, os);</code>
8	<code>}</code>

Rõ ràng dù ta có nhu cầu mới nhưng có thể tận dụng lại hàm TransferData với một chút biến đổi tham số truyền vào. Đầu vào của hàm ta dùng cin, chính là biến đại diện cho bàn phím và đầu ra là tập tin đang mở.

6 Thay đổi nội dung một tập tin và chép vào tập tin mới

Trong ví dụ này, ta có một tập tin chứa MỘT phần tử phân số với tử số và mẫu số cách nhau bằng khoảng trắng. Ta sẽ đọc nội dung này vào biến PHANSO, sau đó biến đổi về dạng chuỗi "TuSo/MauSo" và ghi chuỗi này vào một tập tin khác.

Dòng	
0	<code>#include <fstream></code>
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	
4	<code>//...sử dụng lại operator >> và struct PHANSO của ví dụ trên</code>
5	
6	<code>ostream& operator<< (ostream& outDev, PHANSO& ps){</code>
7	<code> if(ps.mau < 0) {ps.tu = -ps.tu; ps.mau = -ps.mau;}</code>
8	<code> if(ps.tu == 0 ps.mau == 1) outDev << ps.tu;</code>
9	<code> else outDev << ps.tu << "/" << ps.mau;</code>
10	<code> return outDev;</code>
11	<code>}</code>
12	
13	<code>void TransferObjects(istream& inDev, ostream& outDev){</code>
14	<code> PHANSO p;</code>

15	<code>while(inDev >> p){ outDev << p;}</code>
16	<code>}</code>
17	
18	<code>void main(){</code>
19	<code>ifstream ifile("input.inp");</code>
20	<code>if(!ifile){cout << "Khong the tim thay" << endl;}</code>
21	<code>ofstream ofile("output.inp");</code>
22	<code>if(!ofile){cout << "Khong the mo tap tin" << endl;}</code>
23	<code>TransferObjects(ifile, ofile);</code>
24	<code>}</code>

7 Đọc và ghi dữ liệu trên cùng một tập tin

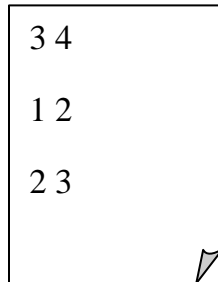
Việc thay thế một phần nội dung trong văn bản là công việc thường thấy (chức năng replace trong Word). Giả sử ta có tập tin nào đó, giờ ta muốn thay thế kí tự khoảng trắng thành kí tự tab. Ta sẽ xử lý điều này trên một tập tin.

Dòng	
0	<code>#include <fstream></code>
1	<code>#include <iostream></code>
2	<code>#include <cstdlib></code>
3	<code>using namespace std;</code>
4	
18	<code>void main(){</code>
19	<code>fstream fs("hello.txt", ios::in ios::out);</code>
20	<code>if(!ifile){cout << "Khong the tim thay" << endl;}</code>
23	<code>char ch;</code>
24	<code>while(fs.get(ch)){</code>
25	<code>if(ch == ' '){</code>
26	<code>int pos = fs.tellg();</code>
27	<code>pos--;</code>
28	<code>fs.seekp(pos);</code>
29	<code>fs.put('\t');</code>
30	<code>fs.seekg(pos + 1);</code>
31	<code>}</code>
32	<code>}</code>
33	<code>}</code>

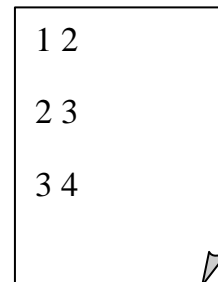
Lưu ý dòng 26, ta sẽ ghi nhận vị trí con trỏ ĐỌC tại pos, sau đó ta lùi lại một vị trí tại dòng 27 và thực hiện di chuyển con trỏ GHI tại vị trí này và chèn kí tự '\t', sau đó ta di chuyển con trỏ ĐỌC lại vị trí pos + 1 (chính là vị trí trước đó ta đã ghi nhận).

8 Bài tập

- 1) Xây dựng hàm ReplaceChar với yêu cầu là thay thế một kí tự cũ nào đó bằng một kí tự mới trong văn bản
- 2) Xây dựng hàm sao chép hai tập tin, với đầu vào của hàm là chuỗi tên của hai tập tin
- 3) Ví dụ xét tập tin PHANSO sau



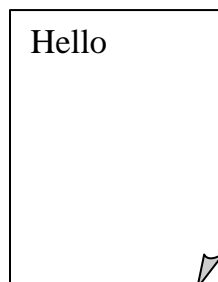
3 4
1 2
2 3



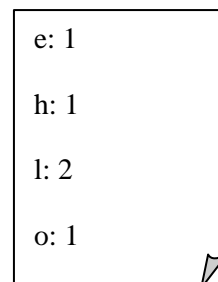
1 2
2 3
3 4

Viết hàm đọc tập tin danh sách PHANSO này, sau đó thực hiện sắp xếp và ghi ra một tập tin khác theo thứ tự tăng dần

- 4) Viết hàm thống kê số lần xuất hiện của các kí tự từ một tập tin văn bản, sau đó ghi số lần xuất hiện ra một tập tin khác.



Hello



e: 1
h: 1
l: 2
o: 1

Sinh viên tự thiết kế hàm main minh họa **các yêu cầu** bên trên **MỘT CÁCH RÕ RÀNG**.