

Chương 3- Con trỏ và tệp tin (phần 2)

Bộ môn Kỹ thuật hệ thống và Mạng máy tính

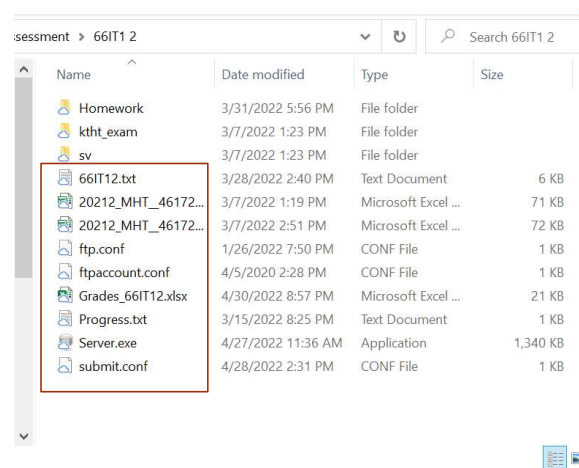
1

Nội dung

- 3.3) Tệp tin (File)
- 3.4) Cách thao tác đọc dữ liệu từ tệp tin văn bản
- 3.5) Cách thao tác tạo tệp văn bản và ghi dữ liệu

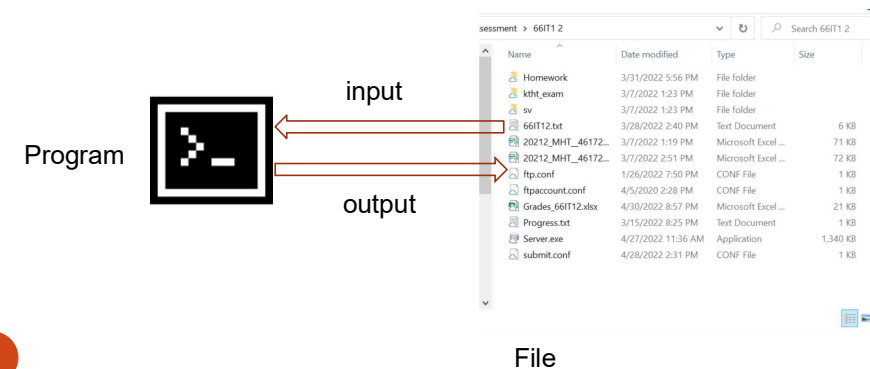
3.3 Tệp tin (File)

- Tệp tin là



Tệp tin (File)

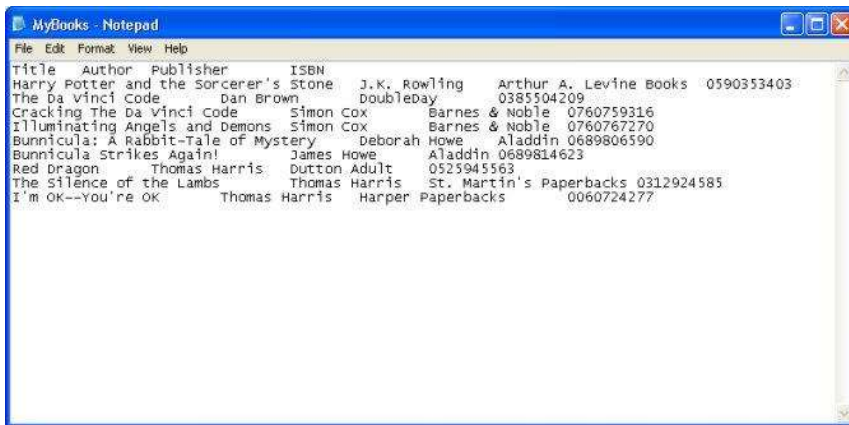
- Chương trình cần thao tác với tệp tin để:
 - Xử lý dữ liệu



4

Tập tin dạng văn bản

Tập văn bản có dữ liệu dạng ký tự có thể đọc bằng chương trình Notepad hoặc Notepad++



Các
dòng
văn
bản

- Tập văn bản có thể chia thành các dòng
- Kết thúc mỗi dòng là kí tự ngắt dòng '\n'

Tập dạng nhị phân

Tập nhị phân có dữ liệu là khối bit nhị phân, khi mở bằng chương trình như HexEditor thì có dạng sau:

- Tập gồm các khối 16 byte
- Mỗi byte biểu diễn với dạng số Hexa (hệ 16)

3.4 Thao tác đọc dữ liệu từ 1 tập tin văn bản

- Điều kiện ban đầu:
- Cần khai báo thư viện **fstream** trong chương trình:
#include <fstream>

Các bước thực hiện

- **Bước 1:** Khai báo

```
ifstream f; // f là biến tệp
```

- **Bước 2:** Mở tệp bằng

```
f.open("input.txt"); // input.txt là tên tệp
```

- **Chú ý:** Sẽ không mở được

- **Bước 3:** Kiểm tra kết quả việc mở tệp

```
if(f.good()==true) // mở thành công
{
    // thực hiện các bước tiếp theo
}
else // mở không thành công
{
    // kết thúc
}
```

9

Bước 4: Đọc dữ liệu từ tệp vào chương trình

- Nếu dữ liệu là kí tự, số nguyên, số thực thì

```
ifstream f;
// mở tệp
f.open("matran_input.txt")
if(f.good()==true) {
    int r,c;
    f >> r >> c;
    ...
}
```

10

- Nếu dữ liệu là chuỗi kí tự thì sử dụng hàm

```
string s;
getline(f, s);
```

- Trường hợp trên, hàm getline sẽ đọc dữ liệu từ tệp f và

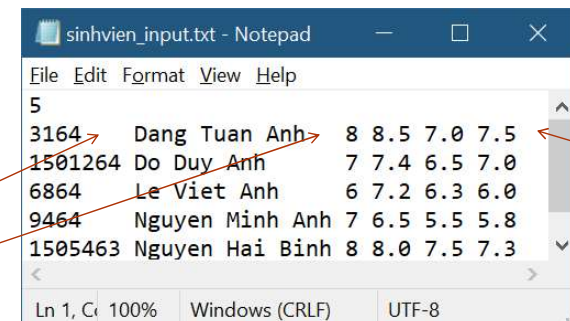
11

Sử dụng hàm getline

- Đọc dữ liệu chuỗi kí tự

```
getline(f, s, '\t'); // ngắt chuỗi khi gặp TAB
getline(f, s, ','); // ngắt chuỗi khi gặp ,
```

Dấu TAB



Dấu xuống
dòng

12

Sau khi đã đọc xong dữ liệu từ tệp

- Bước 5: Đóng tệp bằng

```
f.close();
```

- Sau khi đóng tệp,

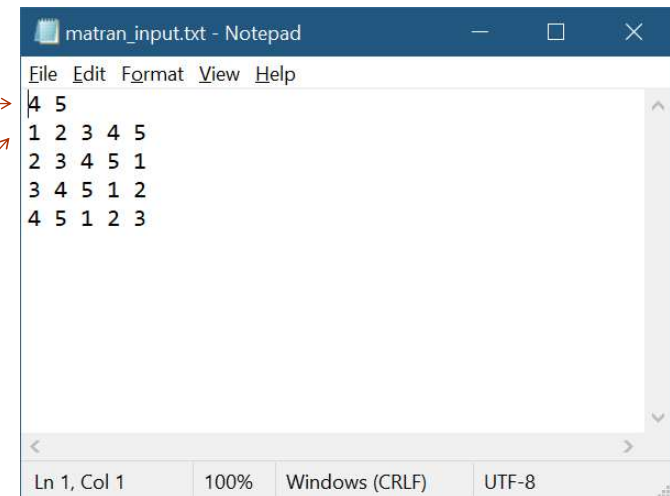
13

Chương trình minh họa file1.cpp

- Đọc dữ liệu từ tệp `matran_input.txt` chứa dữ liệu của 1 ma trận như sau

Dòng 1 là số hàng và cột

Từ dòng 2 là các phần tử của ma trận ghi theo hàng và cột, mỗi phần tử cách nhau 1 dấu trống



14

```
ifstream f; // bien tep f
f.open(filename); // mo tep
if(f.good()==true) { // mo file thanh cong
    int r,c;
    f >> r >> c; // doc so hang r va so cot c tu file

    // Cap phat dong 1 doi tuong Matrix
    Matrix* mp = new Matrix(r,c);

    // bat dau doc cac phan tu ma tran
    for(int i=0;i<r;i++) {
        for(int j=0;j<c;j++) {
            int t;
            f >> t; // doc phan tu (i,j) tu file va luu vao t
            mp->setElement(i,j,t);
        }
    }

    f.close();
}
```

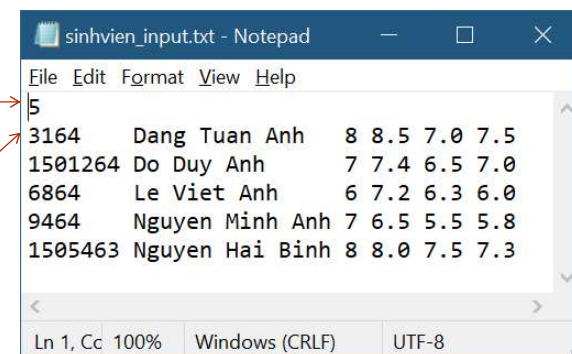
15

Chương trình minh họa file2.cpp

- Tệp `sinhvien_input.txt` chứa các dữ liệu điểm môn C++ của 1 nhóm sinh viên

Dòng 1 là số sinh viên

Từ dòng 2, mỗi dòng lưu thông tin của 1 sinh viên gồm mã sv, họ tên, điểm chuyên cần, điểm bài tập, điểm giữa kỳ và điểm thi



16

Định dạng dữ liệu trên mỗi dòng
<MSSV>**TAB**<Họ tên>**TAB**<Chuyên cần> <Bài tập> <Giữa kỳ> <Thi> **ENDL**

```

ifstream f;
f.open(filename);
if(f.good()==true) {
    f >> soSV; // doc so sinh vien tu file
    f.ignore(); // bo? qua dau xuong dong ENDL
    SinhvienCPP* mp = new SinhvienCPP[soSV]; // cap phat dong 1 mang
    for(int i=0; i<soSV; i++)
    {
        string s;    float c,b,g,t;
        getline(f,s,'\t'); // doc MSSV
        mp[i].setMSSV(s);

        getline(f,s,'\t'); // doc Hoten
        mp[i].setHoten(s);

        f >> c >> b >> g >> t; // doc cac diem so
        mp[i].setChuyencan(c);
        mp[i].setBaitap(b);
        mp[i].setGiuaky(g);
        mp[i].setThi(t);
        f.ignore(); // bo? qua dau xuong dong ENDL
    }
    f.close();
}

```

17

3.5) Thao tác tạo 1 tệp văn bản mới trên đĩa và ghi dữ liệu

- Điều kiện ban đầu:
- Cần khai báo thư viện **fstream** trong chương trình:
#include <fstream>

18

Các bước thực hiện

- **Bước 1:** Khai báo 1 biến đối tượng

```
ofstream f;
```

- **Bước 2:** Tạo tệp mới bằng hàm thành phần **open** theo cú pháp:

```
f.open("matran_output.txt");
```

- **Chú ý:**

19

- **Bước 3:** Kiểm tra kết quả việc tạo tệp mới

```

if(f.good()==true) // tạo tệp thành công
{
    // thực hiện các bước tiếp theo
}
else // tạo tệp không thành công
{
    // kết thúc
}

```

20

Bước 4: Ghi dữ liệu từ chương trình vào tệp

- Với tất cả kiểu dữ liệu

```
f << mt.getElement(i,j) << " ";  
f << sv.getHoten() << "\\t";  
f << endl;
```

- Có thể kết hợp với

```
f << fixed << setprecision(1);  
f << setw(6) << sv.getDQT() << "\\t";
```

21

Sau khi đã ghi xong dữ liệu vào tệp

- Bước 5: Đóng tệp bằng

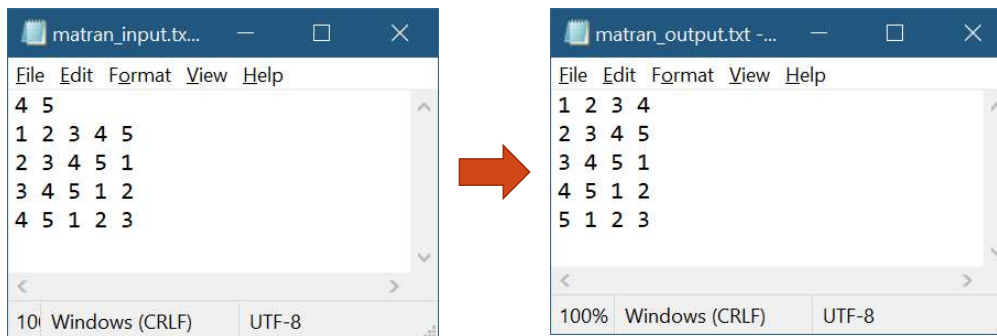
```
f.close();
```

- Sau khi đóng tệp thì

22

Chương trình minh họa file3.cpp

- Ghi ma trận chuyển vị ra tệp matran_output.txt



23

```
// Ham ghi ma tran len file  
void outputMatrixToFile(const Matrix& mt,const string& filename)  
{  
    ofstream f; // bien tep f  
    f.open(filename); // mo tep  
    if(f.good()==true) // mo file thanh cong  
    {  
        for(int i=0; i<mt.getRow(); i++)  
        {  
            for(int j=0; j<mt.getCol(); j++)  
            {  
                // ghi phan tu ma tran ra file f  
                f << mt.getElement(i,j) << " ";  
            }  
            f << endl;  
        }  
        f.close();  
    }  
}  
...  
Matrix m2 = ~m1; // m2 la chuyen vi cua m1  
outputMatrixToFile(m2,"matran_output.txt");
```

24

Chương trình minh họa file4.cpp

- Ghi bảng điểm sinh viên ra tệp sinhvien_output.txt

sinhvien_input.txt - Notepad

	STT	MSSV	Ho ten	DQT	DKT	DHP
5	3164	Dang Tuan Anh	8	8.5	7.0	7.5
	1501264	Do Duy Anh	7	7.4	6.5	7.0
	6864	Le Viet Anh	6	7.2	6.3	6.0
	9464	Nguyen Minh Anh	7	6.5	5.5	5.8
	1505463	Nguyen Hai Binh	8	8.0	7.5	7.3

sinhvien_output.txt - Notepad

	STT	MSSV	Ho ten	DQT	DKT	DHP
1	3164	Dang Tuan Anh	7.8	7.4	7.5	
2	1501264	Do Duy Anh	7.0	6.9	6.9	
3	6864	Le Viet Anh	6.4	6.1	6.2	
4	9464	Nguyen Minh Anh	6.4	5.7	5.9	
5	1505463	Nguyen Hai Binh	7.8	7.4	7.5	

Ln 1, Col 1 100% Windows (CRLF) UTF-8

25

```
// ham ghi bang diem sinh vien len file
void outputSinhvienToFile(int soSV, SinhvienCPP* sv, const string&
filename)
{
    ofstream f; // bien tep f
    f.open(filename); // mo tep
    if(f.good()==true) // mo file thanh cong
    {
        f << fixed << setprecision(1);
        f << "STT" << '\t' << "MSSV" << '\t' << "Ho ten" << "\t\t";
        f << "DQT" << '\t' << "DKT" << '\t' << "DHP" << endl;
        for(int i=0;i<soSV;i++)
        {
            f << i+1 << '\t';
            f << sv[i].getMSSV() << '\t';
            f << sv[i].getHoten() << '\t';
            f << sv[i].getDQT() << '\t' << sv[i].getDKT() << '\t';
            f << sv[i].getDHP() << endl;
        }
        f.close();
    }
    ...
    outputSinhvienToFile(n,dssv,"sinhvien_output.txt");
}
```

26