

File operations

5 - Thao tác với tệp

<https://github.com/tqlong/advprogram>

Nội dung

- **Nhập liệu từ tệp văn bản**
- **Xử lý lỗi với tệp**
- **Kỹ thuật**
 - Giới thiệu các thư viện
<fstream>, <vector>, <algorithm>
 - Xử lý lỗi, xử lý ngoại lệ đơn giản với **<exception>**

Nhập liệu từ tệp (file)

- Hangman hiện thời sử dụng danh sách từ cố định
 - Không cho phép đổi từ vựng (ví dụ: chọn lĩnh vực)
 - Mã nguồn chương trình chứa danh sách từ
 - Phải dịch lại chương trình nếu thay đổi từ
- Giải pháp: *Tách mã nguồn và dữ liệu*
 - Dữ liệu lưu ở tệp
 - Chương trình có mã lệnh đọc tệp, đưa dữ liệu vào bộ nhớ (biến)

Thư viện fstream

- Thư viện trong STL làm việc với file
 - <http://www.cplusplus.com/reference/fstream/fstream/>
- Làm việc với file
 - Phổ biến trong các phần mềm
 - Phức tạp, tỉ mỉ
 - Có nhiều lỗi “không ngờ”
- Học cách sử dụng **<fstream>**
 - Cách nhanh nhất: làm theo bài hướng dẫn (*tutorials*)
 - Ví dụ: <http://www.cplusplus.com/doc/tutorial/files/>

Tạo file, ghi vào file với ofstream

- Biến kiểu **ofstream**

- Đại diện cho một tệp có thể ghi được
- Phương thức **open**: mở file để ghi
- Ghi văn bản giống như dùng **cout**

```
// basic file operations
```

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
int main () {
```

```
    ofstream myfile;
```

```
    myfile.open ("example.txt");
```

```
    myfile << "Writing this to a file.\n";
```

```
    myfile.close();
```

```
    return 0;
```

```
}
```

Sử dụng thư viện fstream

Khai báo biến kiểu ofstream

Mở file example.txt để ghi

Ghi nội dung văn bản vào file

Đóng file lại: giải phóng tài nguyên, ghi xuống đĩa

Tạo file, ghi vào file với ofstream

```
// writing on a text file
#include <iostream>
#include <fstream>
using namespace std;

int main () {
    ofstream myfile ("example.txt");
    if (myfile.is_open())
    {
        myfile << "This is a line.\n";
        myfile << "This is another line.\n";
        myfile.close();
    }
    else cout << "Unable to open file";
    return 0;
}
```

Kiểm tra việc mở tệp có thành công ?

Đọc file với ifstream

```
// reading a text file
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main () {
    string line;
    ifstream myfile ("example.txt");
    if (myfile.is_open())
    {
        while ( getline (myfile,line) )
        {
            cout << line << '\n';
        }
        myfile.close();
    }

    else cout << "Unable to open file";

    return 0;
}
```

Thư viện fstream chứa ifstream

Mở file example.txt đã ghi ở ví dụ trước
Kiểm tra việc mở tệp có thành công ?

Hàm std::getline lưu 1 dòng của tệp vào biến line
và chuyển vị trí đọc xuống dòng tiếp theo
Lặp đến khi std::getline trả về "false" (tức là không
còn gì để đọc, hết tệp)
Đóng tệp, giải phóng tài nguyên hệ thống

Di chuyển vị trí đọc tệp, kích thước tệp

```
// obtaining file size
#include <iostream>
#include <fstream>
using namespace std;

int main () {
    streampos begin,end;
    ifstream myfile ("example.bin", ios::binary);
    begin = myfile.tellg();
    myfile.seekg (0, ios::end);
    end = myfile.tellg();
    myfile.close();
    cout << "size is: " << (end-begin)
         << " bytes.\n";
    return 0;
}
```

Khai báo kiểu lưu trữ vị trí trên stream
Mở tệp ở chế độ nhị phân
Lấy vị trí hiện tại (đầu tệp) vào biến begin
Di chuyển cách vị trí cuối tệp 0 đơn vị (ios::end)
Lấy vị trí hiện tại (cuối tệp) vào biến end
Đóng tệp
Tính kích thước tệp bằng hiệu của end và begin

Đọc từ vựng Hangman từ tệp

Từ vựng của Hangman được lưu trong một tệp văn bản:

- Tập nằm trong thư mục “data” cùng với chương trình
- Mỗi từ trên một dòng
 - Số dòng (số từ) chưa biết trước

→ Cần kiểu dữ liệu lưu trữ số lượng từ “tùy ý”

Thư viện vector

- Cho phép lưu trữ dãy giá trị cùng kiểu
 - Truy xuất giống như mảng tĩnh
 - Ví dụ: **x[i]**
- Cho phép thay đổi kích thước (số phần tử)
 - Có thể coi như mảng “động”
 - Không cần tự lập trình xin cấp phát bộ nhớ
- Nhiều tiện ích thao tác với mảng
 - Thêm, chèn, xóa, sửa
 - Kết hợp với **<algorithm>**: tìm kiếm, sắp xếp ...

<http://www.cplusplus.com/reference/vector/vector/>

Thư viện vector

Chèn vào cuối vector

```
// push_back
#include <iostream>
#include <vector>
using namespace std;

int main ()
{
    vector<int> myvector;
    int myint;

    cout << "Please enter some integers (enter 0 to end):\n";

    while (cin >> myint) {
        myvector.push_back (myint);
    }

    cout << "myvector stores " << int(myvector.size()) << " numbers.\n";

    return 0;
}
```

```
// push_back
```

Sử dụng thư viện vector

Khai báo myvector là vector các số nguyên

Lặp đến khi không còn dữ liệu mới
Phương thức push_back: Thêm myint vào cuối myvector

In số phần tử của myvector

Thư viện vector

Truy xuất các phần tử trong vector

```
vector<int> myvector (10); // 10 zero-initialized ints
```

```
// assign some values:  
unsigned sz = myvector.size();  
for (unsigned i=0; i<sz; i++)  
    myvector.at(i)=i;
```

```
cout << "myvector contains:";  
for (unsigned i=0; i<sz; i++)  
    cout << ' ' << myvector.at(i);  
cout << '\n';
```

```
// reverse vector using operator[]:  
for (unsigned i=0; i<sz/2; i++)  
{  
    int temp;  
    temp = myvector[sz-1-i];  
    myvector[sz-1-i]=myvector[i];  
    myvector[i]=temp;  
}
```

```
cout << "myvector contains:";  
for (unsigned i=0; i<sz; i++)  
    cout << ' ' << myvector[i];  
cout << '\n';
```

Khai báo vector có 10 phần tử

Lưu kích thước vector

Gán giá trị tại vị trí thứ i (tính từ 0) qua phương thức at

In giá trị tại vị trí thứ i qua phương thức at

Sử dụng toán tử [] truy xuất và gán giá trị phần tử của vector giống như mảng tĩnh

In giá trị tại vị trí thứ i qua phương thức at

Đọc từ vựng Hangman từ tệp

Duyệt từng từ của tệp, đưa vào vector

```
vector<string> readWordListFromFile(const string& filePath)
{
    vector<string> wordList;
    string word;
    ifstream wordFile(filePath);
    if (wordFile.is_open()) {
        while (wordFile >> word) {
            wordList.push_back(word);
        }
        wordFile.close();
        return wordList;
    } else {
        cout << "Error: Unable to open vocabulary file "
              << filePath << endl;
        return wordList;
    }
}
```

Tham số của hàm là đường dẫn đến tệp chứa từ, đầu ra của hàm là vector các string
Khai báo biến chứa các từ sẽ đọc

Mở tệp có đường dẫn trong filePath
Kiểm tra tệp có mở thành công
Đọc từng từ (giống cin) đến khi không đọc được nữa, dùng phương thức push_back đưa từ vừa đọc vào wordList
Đóng tệp
Trả về wordList

Báo lỗi khi mở tệp

Thay thế hàm chooseWord()

Chọn từ tiếng Anh từ wordList

```
string chooseWordFromList(const vector<string>& wordList)
{
    if (wordList.empty()) return "";

    const int NUMBER_OF_WORDS = wordList.size();
    return wordList[generateRandomNumber(0, NUMBER_OF_WORDS)];
}
```

Tham số là tham chiếu hằng wordList
Khi truyền vector nên dùng tham chiếu
hằng nếu có thể (không thay đổi)
Kiểm tra nếu wordList có rỗng

Lấy số phần tử của mảng (lớn hơn 0)
Thao tác còn lại giống hệt hàm
chooseWord()

Sửa hàm initialize()

Khởi tạo wordList

```
void initialize(vector<string>& wordList,  
               string& word, string& secretWord,  
               int& incorrectGuess,  
               string& incorrectChars, string& correctChars)  
{  
    wordList =  
        readWordListFromFile("data/Ogden_Picturable_200.txt");  
    word = chooseWordFromList(wordList);  
    secretWord = string(word.length(), '-');  
    incorrectGuess = 0;  
    incorrectChars = "";  
    correctChars = "";  
}
```

Hangman 3.0

https://raw.githubusercontent.com/tqlong/advprogram/master/lec3-hangman/HangMan_3_0.cpp
https://github.com/tqlong/advprogram/raw/master/lec3-hangman/data/Ogden_Picturable_200.txt

Thêm tham số là tham chiếu đến wordList
vì cần đọc từ vựng từ tệp (thay đổi)

Đọc từ vựng từ tệp

Chọn từ trong danh sách từ vừa đọc

Chuẩn hóa dữ liệu

Dữ liệu từ tệp, đặc biệt là dữ liệu tải về từ Internet cần được **chuẩn hóa**

- Đảm bảo chương trình hoạt động với dữ liệu đúng như ý định ban đầu
- Sửa lỗi dữ liệu, loại bỏ dữ liệu “xấu”

Với Hangman, cần chuyển mọi từ về **dạng chữ thường** để phép toán so sánh (**==**, **!=**) hoạt động chính xác

Chuyển từ sang chữ thường

- Có thể đặt ở hàm đọc **readWordFromFile()**
 - Phải chuyển nhiều lần
- Đặt ở hàm **chooseWordFromList()**
 - Duy nhất 1 lần

```
string chooseWordFromList(const vector<string>& wordList)
{
    if (wordList.empty()) return "";
    const int NUMBER_OF_WORDS = wordList.size();
    string word =
        wordList[generateRandomNumber(0, NUMBER_OF_WORDS)];
    return getLowerString(word);
}
```

```
string getLowerString(const string& s)
{
    string res = s;
    int sz = s.size();
    for (int i = 0; i < sz; i++)
        res[i] = tolower(s[i]);
    return res;
}
```

Giới thiệu thư viện algorithm

```
string getLowerString(const string& s)
{
    string res = s;
    int sz = s.size();
    for (int i = 0; i < sz; i++)
        res[i] = tolower(s[i]);
    return res;
}
```



```
#include <algorithm>

string getLowerString(const string& s)
{
    string res = s;
    transform(s.begin(), s.end(), res.begin(), ::tolower);
    return res;
}
```

Duyệt mảng là một
thao tác phổ biến
nhất trong lập trình

Duyệt từ đầu đến cuối của **s**, biến đổi bằng hàm **tolower()**, đặt kết quả lần lượt vào các ký tự tính từ đầu của **res**

<http://stackoverflow.com/questions/313970/how-to-convert-stdstring-to-lower-case>

Con trỏ duyệt (Iterator)

s.begin(), s.end()

trả về các *iterator*

là khái niệm khái quát hóa của chỉ số mảng

- Sẽ học kỹ hơn ở các buổi sau
- <http://www.cplusplus.com/reference/iterator/>

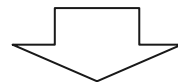
```
#include <algorithm>

string getLowerString(const string& s)
{
    string res = s;
    transform(s.begin(), s.end(), res.begin(), ::tolower);
    return res;
}
```

Chuẩn hóa phán đoán của người chơi

```
char getUserInput()  
{  
    char ch;  
    cin >> ch;  
    return tolower(ch);  
}
```

```
cin >> ch;
```



```
ch = getUserInput();
```

Hangman 3.1

Hangman 3.1

https://raw.githubusercontent.com/tqlong/advprogram/master/lec3-hangman/HangMan_3_1.cpp

- Đọc dữ liệu từ tệp
 - Sử dụng **<fstream>**, **<vector>**
- Lựa chọn phần tử ngẫu nhiên trong **vector**
- Chuẩn hóa từ về dạng chữ thường
 - Duyệt mảng, biến đổi sử dụng **<algorithm>**

Bài tập: Chọn tệp dữ liệu

- Từ tham số dòng lệnh
- Từ lựa chọn của người chơi

Nội dung

- Nhập liệu từ tệp văn bản
- **Xử lý lỗi với tệp**
- Kỹ thuật
 - Thư viện
<fstream>, <vector>, <algorithm>
 - Xử lý lỗi, xử lý ngoại lệ đơn giản với **<exception>**

Xử lý lỗi runtime

- Chương trình luôn chạy đúng khi dữ liệu “hoàn hảo”
- Thực tế, trong Hangman, dữ liệu đọc từ tệp hay gặp phải nhiều lỗi “bất ngờ”
 - Tệp không tồn tại
 - Đĩa lỗi
 - Người dùng nhập ký tự lạ ...
 - Nếu không xử lý, chương trình sẽ cho kết quả bất định (dù không có lỗi logic trong mã nguồn)

Xử lý lỗi runtime

Với C++, xử lý lỗi (còn gọi là xử lý ngoại lệ) được hỗ trợ bởi thư viện **<exception>**

<http://www.cplusplus.com/reference/exception/exception/>

Hướng dẫn: <http://www.cplusplus.com/doc/tutorial/exceptions/>

Lập trình xử lý lỗi (*exception handling*):

- Đoạn mã có thể sinh lỗi ném ngoại lệ bằng lệnh **throw**
- Đoạn mã xử lý lỗi bắt ngoại lệ bằng khối lệnh **try ... catch ...**

Xử lý lỗi cho readWordFromList()

```
vector<string> readWordListFromFile(const string& filePath)
{
    vector<string> wordList;
    string word;
    ifstream wordFile(filePath);
    if (wordFile.is_open()) {
        while (wordFile >> word) {
            wordList.push_back(word);
        }
        wordFile.close();
        return wordList;
    } else {
        cout << "Error: Unable to open vocabulary file "
              << filePath << endl;
        return wordList;
    }
}
```

Hiện giờ xử lý lỗi không mở được file bằng cách trả về một vector rỗng

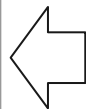
- Có thể nhầm với trường hợp file rỗng không có từ nào

Xử lý lỗi cho readWordFromList()

```
#include <stdexcept>
...
vector<string> readWordListFromFile(const string& filePath)
{
    vector<string> wordList;
    string word;
    ifstream wordFile(filePath);
    if (wordFile.is_open()) {
        while (wordFile >> word) {
            wordList.push_back(word);
        }
        wordFile.close();
        return wordList;
    } else {
        throw domain_error("Error: Unable to open "
                           "vocabulary file " + filePath);
    }
}
```



Sử dụng thư viện stdexcept

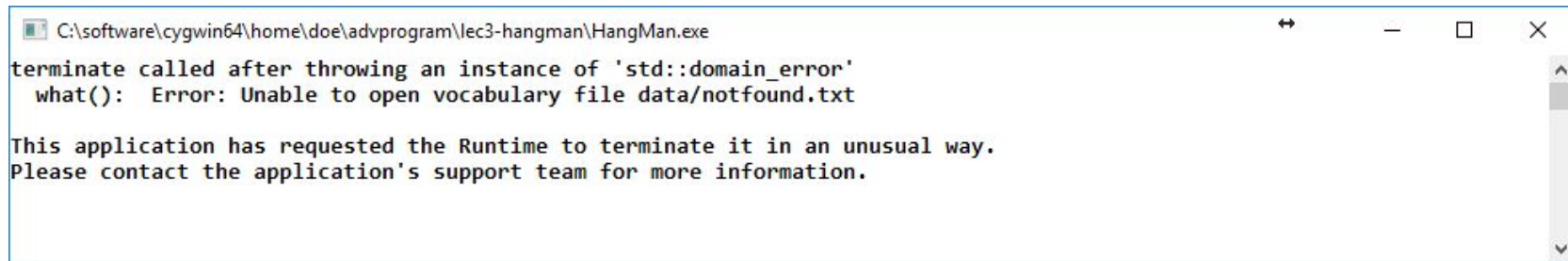


Ném ngoại lệ domain_error
được định nghĩa trong
stdexcept
Không cần return kết quả
nữa

Xử lý lỗi cho readWordFromList()

Thử đổi tên file thành một file không có trên đĩa

- Chương trình báo lỗi **domain_error**



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\software\cygwin64\home\doe\advprogram\lec3-hangman\HangMan.exe. The window contains the following text: terminate called after throwing an instance of 'std::domain_error', what(): Error: Unable to open vocabulary file data/notfound.txt. Below this, a standard Windows error message is displayed: This application has requested the Runtime to terminate it in an unusual way. Please contact the application's support team for more information.

```
C:\software\cygwin64\home\doe\advprogram\lec3-hangman\HangMan.exe
terminate called after throwing an instance of 'std::domain_error'
what(): Error: Unable to open vocabulary file data/notfound.txt

This application has requested the Runtime to terminate it in an unusual way.
Please contact the application's support team for more information.
```

Bắt ngoại lệ trong initialize()

```
void initialize(vector<string>& wordList, string& word,
               string& secretWord, int& incorrectGuess,
               string& incorrectChars, string& correctChars)
{
    try {
        wordList = readWordListFromFile("data/notfound.txt");
    } catch (domain_error e) {
        cout << "Exception occurs: " << e.what() << endl;
        exit(1);
    }
    word = chooseWordFromList(wordList);
    secretWord = string(word.length(), '-');
    incorrectGuess = 0;
    incorrectChars = "";
    correctChars = "";
}
```

Hangman 3.2

https://github.com/tqlong/advprogram/raw/master/lec3-hangman/HangMan_3_2.cpp

exit(1)



```
C:\software\cygwin64\home\doe\advprogram\lec3-hangman\HangMan.exe
Exception occurs: Error: Unable to open vocabulary file data/notfound.txt
Process returned 1 (0x1)   execution time : 0.010 s
Press any key to continue.
```