

# File và các thao tác trên tập tin

Nguyễn Đức Thắng

# File

- Một file trên máy tính là một tài nguyên dùng để lưu trữ thông tin lâu dài, sử dụng cho các chương trình máy tính.
- Mở file và đóng file trong C:

```
const char *filePath = "C:/Users/ADMIN/Desktop/my_document.txt";
FILE *file;

file = fopen(filePath, "r");
if (!file)
    std::cout << "Can not open this file" << std::endl;
else
    std::cout << "File is opened" << std::endl;

fclose(file);
```

# Các chế độ mở file

Mode	Description
r	Opens an existing text file for reading purpose.
w	Opens a text file for writing. If it does not exist, then a new file is created. Here your program will start writing content from the beginning of the file.
a	Opens a text file for writing in appending mode. If it does not exist, then a new file is created. Here your program will start appending content in the existing file content.
r+	Opens a text file for both reading and writing.
w+	Opens a text file for both reading and writing. It first truncates the file to zero length if it exists, otherwise creates a file if it does not exist.
a+	Opens a text file for both reading and writing. It creates the file if it does not exist. The reading will start from the beginning but writing can only be appended.

# Ghi file

- Sử dụng các hàm: **fputc** (ghi ký tự vào file), **fputs** (ghi chuỗi vào file), **fprintf** (ghi file theo định dạng của C).

```
int fputc(int c, FILE *f);
```

```
int fputs(const char *str, FILE *f);
```

```
int fprintf(FILE *f, const char *format, ...);
```



# Đọc file

- Tương tự như ghi file, với đọc file ta có các hàm: **fgetc**, **fgets**, **fscanf**

```
int fgetc(FILE *f);
```

```
char* fgets(char *buf, int n, FILE *f);
```

```
int fscanf(FILE *f, const char *format, ...);
```



# Ví dụ

```
#include <iostream>
#include <string>
#include <cstdio>

void writeToFile(FILE *file)
{
    for (int i = 1; i <= 5; i++)
        fprintf(file, "This is an example line %d\n", i);
}

void readFromFile(FILE *file)
{
    char str[255];
    while (fgets(str, 255, file) != NULL)
    {
        std::cout << str;
    }
}
```

```
int main()
{
    const char *filePath = "C:/Users/ADMIN/Desktop/my_document.txt";
    FILE *file;

    file = fopen(filePath, "w+");
    if (!file)
        std::cout << "Can not open this file" << std::endl;
    else
        std::cout << "File is opened" << std::endl;

    writeToFile(file);
    readFromFile(file);

    fclose(file);

    return 0;
}
```

# Thay đổi vị trí con trỏ file (cstdio)

```
int fseek(FILE *f, long int offset, int origin);
```

Trong đó:

- f là con trỏ trỏ đến đối tượng FILE đang mở.
- offset là số bytes được cộng thêm tính từ vị trí origin.
- origin là địa điểm đặt con trỏ trong file:

Constant	Description
SEEK_SET	Beginning of file
SEEK_CUR	Current position of the file pointer
SEEK_END	End of file

# Xác định kích thước file (cstdio)

```
long int ftell(FILE *f);
```

```
__int64 size_of_file(FILE *file)
{
    fseek(file, 0, SEEK_END);
    __int64 size = ftell(file);
    fseek(file, 0, SEEK_SET);

    return size;
}
```





# Đọc/ghi file nhị phân

```
size_t fwrite(const void *ptr, size_t size, size_t count, FILE *f);
```

```
size_t fread(void *ptr, size_t size, size_t count, FILE *f);
```

```
void writeToFile(FILE *file)
{
    char *s = "Hello everyone!";
    fwrite(s, strlen(s), 1, file);
}

void readFromFile(FILE *file)
{
    void *ptr = operator new(255); //allocate 255 bytes on Heap
    fread(ptr, 255, 1, file);
    (static_cast<char *>(ptr))[255] = '\0';
    std::cout << static_cast<char *>(ptr) << std::endl;
}
```

# Đọc/ghi struct

```
struct Employee
{
    __int32 ID;
    char name[50];
};

Employee emps[3] =
{
    { 1, "Le Tran Dat" },
    { 2, "Ngo Doan Tuan" },
    { 3, "Le Dinh Huy" }
};

void writeToFile(FILE *file)
{
    for (int i = 0; i < 3; i++)
    {
        fwrite(&emps[i], sizeof(Employee), 1, file);
    }
}

void readFromFile(FILE *file)
{
    Employee emp;
    for (int i = 0; i < 3; i++)
    {
        fread(&emp, sizeof(Employee), 1, file);

        std::cout << emp.ID << std::endl;
        std::cout << emp.name << std::endl;
        std::cout << "=====" << std::endl;
    }
}
```

# Luyện tập



# Đọc file trong C++

- Các thao tác file C dùng con trỏ file ở thư viện `cstdin`, trong C++, ta dùng stream ở thư viện `fstream`.

- File: 1 2 3 4 5

- File: 1 2 3 4 5 a b c

```
#include <iostream>
#include <fstream>

int main()
{
    std::ifstream fileInput("C:/Users/ADMIN/Desktop/my_document.txt");

    if (fileInput.fail())
    {
        std::cout << "Failed to open this file!" << std::endl;
        return -1;
    }
    while (!fileInput.eof())
    {
        int n;
        fileInput >> n;
        std::cout << n << " ";
    }
    std::cout << std::endl;

    fileInput.close();

    return 0;
}
```

# Khắc phục lỗi

```
std::ifstream fileInput("C:/Users/ADMIN/Desktop/my_document.txt");

if (fileInput.fail())
    std::cout << "Failed to open this file!" << std::endl;

while (!fileInput.eof())
{
    int n;
    if (fileInput >> n)
        std::cout << n << " ";
    else
        return -2;
}
std::cout << std::endl;

fileInput.close();

system("pause");
return 0;
```

# Đọc tất cả ký tự trong file

- Đọc toàn bộ dữ liệu trong file và in ra màn hình, chúng ta có thể đọc từng ký tự trong file sử dụng biến kiểu char thay vì số nguyên.

```
while (!fileInput.eof())
{
    char c;
    if(fileInput >> c)
        std::cout << c << " ";
}
std::cout << std::endl;
```

# Lệnh đọc file khác

```
This is line 1  
This is line 2  
This is line 3  
This is line 4  
This is line 5
```

```
while (!fileInput.eof())  
{  
    char temp[255];  
    fileInput.getline(temp, 255);  
    std::string line = temp;  
    std::cout << line << std::endl;  
}
```

- Tìm hiểu thêm: <http://www.cplusplus.com/reference/fstream/ifstream/>

# Các chế độ đọc file C++

Ios file mode	Meaning
<code>ate</code>	Seeks to the end of the file before reading/writing
<code>binary</code>	Opens the file in binary mode (instead of text mode)
<code>in</code>	Opens the file in read mode (default for <code>ifstream</code> )



# Ví dụ

```
std::fstream fileInput("C:/Users/ADMIN/Desktop/my_document.txt", std::ios::in);

if (fileInput.fail())
    std::cout << "Failed to open this file!" << std::endl;

while (!fileInput.eof())
{
    char temp[255];
    fileInput.getline(temp, 255);
    std::string line = temp;
    std::cout << line << std::endl;
}
std::cout << std::endl;

fileInput.close();
```

```
std::fstream fileInput("C:/Users/ADMIN/Desktop/my_document.txt", std::ios::in | std::ios::binary);
```

# Sử dụng 1 input stream cho nhiều file

```
std::ifstream fileInput("C:/Users/ADMIN/Desktop/my_document.txt");
```

```
//.....
```

```
fileInput.close();
```

```
fileInput.open("C:/Users/ADMIN/Desktop/my_document2.txt");
```

```
//.....
```

```
fileInput.close();
```



# Ghi dữ liệu vào file trong C++

```
void writeDataToFile(std::string file)
{
    std::ofstream fileOutput(file);

    if (fileOutput.fail())
    {
        std::cout << "Cannot open file at " << file << std::endl;
        return;
    }

    fileOutput << "Hello world!" << std::endl;
    fileOutput << "I'm Nguyen Duc Thang" << std::endl;
    for (int i = 1; i <= 10; i++)
    {
        fileOutput << i << std::endl;
    }
}
```

# File output mode

```
std::fstream file("C:/Users/ADMIN/Desktop/my_document.txt", std::ios::out);
```

```
std::fstream file("C:/Users/ADMIN/Desktop/my_document.txt", std::ios::out | std::ios::binary);
```

Ios file mode	Meaning
app	Opens the file in append mode
ate	Seeks to the end of the file before reading/writing
binary	Opens the file in binary mode (instead of text mode)
nocreate	Opens the file only if it already exists
noreplace	Opens the file only if it does not already exist
out	Opens the file in write mode (default for ofstream)
trunc	Erases the file if it already exists

# Sử dụng 1 output stream cho nhiều file

```
void writeDataToFile(std::string file)
{
    std::ofstream fileOutput(file);

    if (fileOutput.fail())
    {
        std::cout << "Cannot open file at " << file << std::endl;
        return;
    }

    fileOutput << "Hello world!" << std::endl;
    fileOutput << "I'm Nguyen Duc Thang" << std::endl;

    fileOutput.close();

    fileOutput.open(file, std::ios::app);

    fileOutput << "Goodbye everyone!" << std::endl;
}
```

# Luyện tập

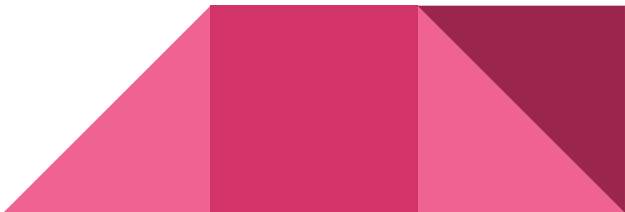


# Stream for string C++

- Ngôn ngữ C++ cung cấp 1 số stream khác sử dụng cho đối tượng string.
- Một số stream cho string C++: **istringstream**, **ostringstream**, **stringstream** trong thư viện **sstream**.
- Có 2 cách để đưa dữ liệu buffer vào đối tượng stringstream: sử dụng << hoặc set value bằng phương thức **str**

```
std::stringstream ss;  
ss << "I'm learning" << " " << "C++ programming language" << std::endl;
```

```
std::stringstream ss;  
ss.str("I'm learning C++ programming language\n");
```



# Stream for string C++

- Phương thức str còn dùng để truy xuất dữ liệu trong stringstream.

```
std::stringstream ss;  
ss << "I'm learning" << " " << "C++ programming language" << std::endl;  
std::cout << ss.str();
```

- Để lấy dữ liệu từ buffer của đối tượng stringstream đưa vào std::string, chúng ta sử dụng toán tử >>

```
std::stringstream os;  
os << "12345 67.89"; // insert a string of numbers into the stream  
  
std::string strValue;  
os >> strValue;  
  
std::string strValue2;  
os >> strValue2;  
  
// print the numbers separated by a dash  
std::cout << strValue << " - " << strValue2 << std::endl;
```



# Chuyển đổi giữa string và số

- Số sang string:

```
std::stringstream ss;  
  
int nValue = 12345;  
double dValue = 67.89;  
ss << nValue << " " << dValue;  
  
std::string strValue1, strValue2;  
ss >> strValue1 >> strValue2;  
  
std::cout << strValue1 << " " << strValue2 << std::endl;
```

# Chuyển đổi giữa string và số

- String sang số:

```
std::stringstream ss;  
ss << "12345 67.89";  
  
int nValue;  
double dValue;  
ss >> nValue >> dValue;  
  
std::cout << nValue << " " << dValue << std::endl;
```

# Clear stringstream để sử dụng lại

- Chúng ta có thể dùng một đối tượng stringstream cho nhiều string khác nhau bằng cách xóa dữ liệu bên trong buffer của nó đi.

```
std::stringstream ss;  
ss.str(""); //erase the buffer  
ss.clear(); //erase and reset error flags
```





Thank you!