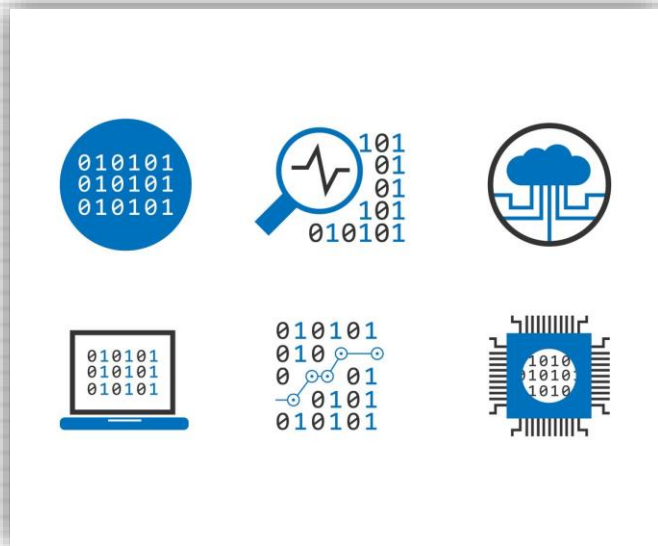


# KỸ THUẬT LẬP TRÌNH

## CHƯƠNG 2: TẬP TIN NHỊ PHÂN (BINARY FILE)



GV: Phạm Nguyễn Sơn Tùng

Email: [pnstung@fit.hcmus.edu.vn](mailto:pnstung@fit.hcmus.edu.vn)

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

## Tập tin văn bản

Là tập tin chỉ lưu trữ thuần túy văn bản, trong đó các kí tự, người dùng có thể đọc được.

- Dễ truy xuất và xử lí.
- Độ bảo mật kém.
- Tốc độ truy xuất kém.

Các tập tin văn bản có đuôi mở rộng: \*.INI, \*.CFG, \*.TXT, \*.HTML, \*.XML

## Tập tin nhị phân

Là tập tin chứa dữ liệu là chuỗi byte, không thể biểu diễn ở dạng văn bản bình thường, do đó tập tin này cần có phần mềm chuyên dụng để đọc, người dùng không đọc được.

- Truy xuất và xử lí phức tạp hơn.
- Tốc độ truy xuất nhanh.
- Tính bảo mật cao hơn.

Các tập tin nhị phân có đuôi mở rộng: \*.DOC, \*.XLS, \*.PDF, \*.PPT, \*.JPG, \*.PCX, \*.BMP, \*.TIF, \*.MP3, \*.WAV, \*.AVI,...

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

Tạo một tập tin nhị phân đơn giản.

```
#include <fstream>
#include <iostream>
using namespace std;
int main()
{
    fstream output;
    output.open("binary.bin", ios::binary | ios::out);
    int p = 5;
    output.write((char*) &p, sizeof(p));
    output.close();
}
```

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

Đọc một tập tin vừa tạo.

```
#include <fstream>
#include <iostream>
using namespace std;
int main()
{
    fstream input;
    input.open("data.bin", ios::binary | ios::in);
    int p = 0;
    input.read((char*) &p, sizeof(p));
    cout << p;
    input.close();
}
```

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

Cùng một lượng dữ liệu tập tin nhị phân có kích thước bé hơn tập tin văn bản.

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
#pragma warning (disable : 4996)

struct SinhVien
{
    string HoTen;
    string MSSV;
    int Tuoi;
};
```

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

```
SinhVien NhapThongTin(string hoten, string mssv, int tuoi)
{
    SinhVien temp;
    temp.HoTen = hoten;
    temp.MSSV = mssv;
    temp.Tuoi = tuoi;
    return temp;
}

void GhiTapTin(fstream& output, SinhVien sv)
{
    output.write((char*)&sv, sizeof(SinhVien));
}
```

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

```
int main()
{
    SinhVien sv1 = NhapThongTin("Nguyen Van A", "2021200", 18);
    SinhVien sv2 = NhapThongTin("Tran Van B", "1821200", 20);
    SinhVien sv3 = NhapThongTin("Ly Quang C", "1921200", 19);
    SinhVien sv4 = NhapThongTin("Ngo Hong D", "2021200", 18);
    fstream output;
    output.open("data.dat", ios_base::binary | ios_base::out);
    GhiTapTin(output, sv1);
    GhiTapTin(output, sv2);
    GhiTapTin(output, sv3);
    GhiTapTin(output, sv4);
    output.close();
}
```

# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

Đọc dữ liệu đã ghi.

```
int main()
{
    fstream input;
    input.open("data.dat", ios_base::binary | ios_base::in);
    SinhVien sv;
    input.read((char*)& sv, sizeof(SinhVien));
}
```



# GIỚI THIỆU VỀ TẬP TIN NHỊ PHÂN

Tập tin **Bitmap** là tập tin lưu ảnh của hệ điều hành Windows. Đặc điểm nổi bật nhất của định dạng BMP là tập tin hình ảnh thường không được nén bằng bất kỳ thuật toán nào. Khi lưu ảnh, các điểm ảnh được ghi trực tiếp vào tập tin.

Cấu trúc tập tin ảnh BMP bao gồm 4 phần:

- **Bitmap Header** (14 bytes): giúp nhận dạng tập tin bitmap.
- **Bitmap Information** (40 bytes): lưu một số thông tin chi tiết giúp hiển thị ảnh.
- **Color Table** (4 \* NumColors bytes) định nghĩa các màu sẽ được sử dụng trong ảnh.
- **Bitmap Data**: lưu dữ liệu ảnh.

# CÁCH ĐỌC MỘT TẬP TIN BITMAP

**Bitmap Header (14 bytes):** giúp nhận dạng tập tin bitmap.

Header (14 bytes)	Kích thước (Bytes)	Mô tả
Signature	2	Chữ ký
FileSize	4	Kích thước tập tin
Reserved	4	Dùng để lưu các thông tin khác
DataOffset	4	Offset bắt đầu dữ liệu Bitmap

# CÁCH ĐỌC MỘT TẬP TIN BITMAP

**Bitmap Information** (40 bytes): lưu một số thông tin chi tiết giúp hiển thị ảnh.

Information (40 bytes)	Kích thước (Bytes)	Mô tả
Size	4	Kích thước của vùng Info
Width	4	Chiều dài của bitmap
Height	4	Chiều cao của bitmap
Planes	2	Số lượng mặt phẳng màu của bitmap, ảnh bitmap luôn luôn là 1.
BitCount	2	Số lượng bit biểu diễn một pixel

# CÁCH ĐỌC MỘT TẬP TIN BITMAP

**Bitmap Information** (40 bytes): lưu một số thông tin chi tiết giúp hiển thị ảnh.

Compression	4	Phương pháp sử dụng để nén bitmap
ImageSize	4	Kích thước của ảnh
XPixelsPerM	4	Độ phân giải theo chiều ngang (pixel/mét)
YPixelsPerM	4	Độ phân giải theo chiều dọc(pixel/mét)
ColorsUsed	4	Số lượng màu thực sự sử dụng
ColorsImportant	4	Số lượng các màu quan trọng

# CÁCH ĐỌC MỘT TẬP TIN BITMAP

**Color Table** (4 \* NumColors bytes) định nghĩa các màu sẽ được sử dụng trong ảnh.

Color Table (x bytes)	Kích thước (Bytes)	Mô tả
Red	1	Cường độ màu đỏ
Green	1	Cường độ màu xanh lá
Blue	1	Cường độ màu xanh dương
Reserved	1	Dự trữ (không sử dụng)

# CÁCH ĐỌC MỘT TẬP TIN BITMAP

**Bitmap Data:** lưu dữ liệu ảnh.

Bitmap Data	Kích thước (Bytes)	Mô tả
data	Kích thước còn lại ảnh	Dữ liệu pixel