

Tập tin nhị phân (tt)

GV. Nguyễn Minh Huy

Nội dung



- Tập tin nhị phân.
- Bài tập BMP.

Nội dung



- **Tập tin nhị phân.**
- **Bài tập BMP.**



■ Đọc ghi cấu trúc:

- Ánh xạ các byte tập tin vào thành phần cấu trúc.
 - ➔ Hiệu quả hơn đọc ghi từng thành phần.
- Thứ tự ánh xạ theo thứ tự khai báo thành phần.

```
struct PhanSo {  
    int tu;  
    int mau;  
};
```

```
void docPS(FILE *f, PhanSo &p) {  
    fread( &p, sizeof(PhanSo), 1, f );  
}  
  
void ghiPS(FILE *f, PhanSo p) {  
    fwrite( &p, sizeof(PhanSo), 1, f );  
}
```

// Đọc 8 bytes vào phân số p.
// 4 bytes đầu vào p.tu.
// 4 bytes sau vào p.mau.

// Ghi 8 bytes phân số p ra tập tin.
// p.tu ra 4 bytes đầu.
// p.mau ra 4 bytes sau.



■ Thư viện <stdint.h>:

- Kích thước số nguyên bao nhiêu bytes?
→ Tùy thuộc hệ máy tính.
- Đọc/ghi nhị phân cần số nguyên kích thước cố định.
→ Thư viện <stdint.h>
- Kiểu số nguyên xác định:
 - Số nguyên 1 byte: int8_t, uint8_t.
 - Số nguyên 2 bytes: int16_t, uint16_t.
 - Số nguyên 4 bytes: int32_t, uint32_t.
 - Số nguyên 8 bytes: int64_t, uint64_t.

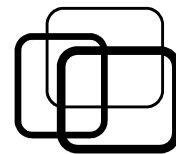
```
#include <stdint.h>

struct PhanSo
{
    int32_t tu;
    int32_t mau;
};
```

Nội dung



- Tập tin nhị phân.
- **Bài tập BMP.**



■ Bài tập 5.1:

Viết chương trình C/C++ cắt ảnh BMP thành các phần bằng nhau dùng tham số dòng lệnh. Mỗi phần ảnh cắt được lưu vào một file BMP.

Cú pháp dòng lệnh:

<tên chương trình> <file Bmp> [-h <số phần cắt dọc>] [-w <số phần cắt ngang>]

Ví dụ: chương trình tên cutbmp.exe

- Cắt 3 phần theo chiều cao (lưu vào 3 ảnh BMP):

cutbmp.exe d:/images/img1.bmp -h 3

- Cắt 2 phần theo chiều cao, 4 phần theo chiều dọc (lưu vào 8 ảnh BMP):

cutbmp.exe d:/images/img1.bmp -h 2 -w 4

