

# 二進位檔案和BMP



# 還記得檔案要如何存取嗎？

- fopen -- 打開檔案

```
FILE *fp;  
fp = fopen("filename", "...");
```

- fscanf

```
fscanf(fp, "...", "...");
```

- fprintf

```
fprintf(fp, "...", "...");
```

不過BMP不方便用fscanf, fprintf存取，因為他是二進位檔案

# 二進位檔案存取

## 打開一個二進位檔案

一樣，是用 `fopen()`，但要注意的是開啟檔案格式是`rb`

- 讀取

```
FILE *fp = fopen("test.bmp", "rb");
```

- 寫入

```
FILE *fp = fopen("test.bmp", "wb");
```

# 二進位檔案存取

## 讀取 **fread**

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

參數說明：

- void\* ptr

就像scanf()，也需要丟一個變數的位址過去，以承接讀取結果

- size\_t size

由於讀取檔案是以二進位方式，因此需要打包成什麼型態的大小

- size\_t count

讀取幾個這個型態

- FILE\* stream

檔案指標

# 二進位檔案存取

## 跳躍 **fseek**

有時在二進位檔案裡面跳來跳去讀取是很重要的@@

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

參數說明：

- FILE\* stream

檔案指標

- offset

偏移量

- origin

要如何偏移呢@@？

# 二進位檔案存取

## 跳躍 **fseek**

### **offset, origin**

origin 總共有三種：

1. SEEK\_SET -> 檔案開頭
2. SEEK\_CUR -> 當前讀取位置
3. SEEK\_END -> 檔案結尾

例子：

```
fseek(fp, 2, SEEK_SET);
```

把目前讀寫游標指到距離開頭兩個**bytes**的地方。

# 二進位檔案存取

## 寫入 **fwrite**

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

參數說明：

- void\* ptr

丟一個東西過去給他寫

- size\_t size

打包 型態的大小

- count

有幾個

- FILE\* stream

檔案指標



# 練習一下

- 把兩個int寫進一個binary檔案 + 讀取

寫入：

```
int a1 = 1, a2 = 2;  
fwrite(&a1, sizeof(int), 1, fp);  
fwrite(&a2, sizeof(int), 1, fp);
```

讀取：

```
int a1, a2;  
fread(&a1, sizeof(int), 1, fp);  
fread(&a2, sizeof(int), 1, fp);
```

# 練習一下

- 把一個int和一個int陣列寫進binary檔案＋讀取
  - 首先，先讀入一個int，代表接下來會有幾個正整數，然後再把剩下的正整數全部讀入加總。

讀取：

```
int cnt;  
int arr[100];  
fread(&cnt, sizeof(int), 1, fp);  
fread(arr, sizeof(int), cnt, fp);
```

寫入：

```
int cnt = ?;  
int arr[100] = ....;  
fwrite(&cnt, sizeof(int), 1, fp);  
fwrite(arr, sizeof(int), cnt, fp);
```

# 使用者定義型別

問：那可不可以寫入自行定義的型別？

可以的

```
struct pack{  
    int weight;  
    int count;  
}Mypack;  
Mypack.weight = 10;  
Mypack.count = 20;  
fwrite(&Mypack, sizeof(pack), 1, fp);
```

相當於....

```
int weight = 10;  
int count = 20;  
fwrite(&weight, sizeof(int), 1, fp);  
fwrite(&count, sizeof(int), 1, fp);
```

要注意**struct padding**

# 記得要關閉檔案

```
fclose(fp);
```

# 再回到BMP

來讀取前面部份東西吧！

可以先試著讀取一個BMP前面的幾個位元。

先備：

1. 要如何在printf裡面將參數用十六進位弄出來

- %X 顯示十六進位
- %2X 十六進位，兩碼
- %02X 十六進位，兩碼，不足兩碼補零

2. fread的回傳值

回傳成功讀取的區塊數量

# 再回到BMP

來讀取前面部份東西吧！

```
FILE* fp = fopen("test1.bmp", "rb");  
unsigned char buff;  
int cnt = 0, line = 0;  
while(fread(&buff, sizeof(char), 1, fp) == 1) {  
    cnt += 1;  
    printf("%02X", buff);  
    if(cnt >= 8) {  
        cnt = 0;  
        puts("");  
        line += 1;  
        if(line > 10) break;  
    }  
}
```

# 再回到**BMP**

來讀取前面部份東西吧！

```
42 4D F6 D4 01 00 00 00
00 00 36 00 00 00 28 00
00 00 C8 00 00 00 C8 00
00 00 01 00 18 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
FF 00 00 FF 00 00 FF 00
00 FF 00 00 FF 00 00 FF
...
```

# 再回到BMP

## 檔案格式

BMP檔案分成四區塊：

- 標頭
- 挾帶訊息
- 調色盤  
不重要(X)
- 點陣圖資料

以下大小單位為byte。



# 再回到BMP

## 檔案格式

### 標頭

偏移	名稱	大小	註解
0000h	ID	2	應該要是BM
0002h	檔案大小	4	整個點陣圖檔案大小
0006h	Reserved	4	保留欄位
000Ah	資料偏移	4	點陣圖資料開始之前的偏移量(通常是54)

# 再回到BMP

## 檔案格式

### 挾帶訊息

偏移	名稱	大小	註解
000Eh	大小	4	挾帶訊息大小，設40
0012h	Width	4	點陣圖的寬度，以像素（pixel）為單位
0016h	Height	4	點陣圖的高度，以像素（pixel）為單位
001Ah	Planes	2	圖層數，必須是1
001Ch	BPP	2	每個像素的位元數(RGB = 24)
001Eh	Comp	4	壓縮方式(不用管，設零就好:P)
0022h	Bitmap	4	點陣圖資料的大小（單位：byte）

# 再回到BMP

檔案格式

挾帶訊息

偏移	名稱	大小	註解
0026h	H-Reso	4	水平解析度(不用管，設零:P)
002Ah	V-Reso	4	垂直解析度(不用管，設零:P)
002Eh	Used Colors	4	調色盤顏色數(不用管，設零:P)
0032h	Important Colors	4	重要的顏色數(不用管，設零:P)

# 再回到**BMP**

檔案格式

點陣圖資料

- 每掃一行就是一條在圖片上的顏色

`b g r b g r b g r b g r .....`

# 再回到**BMP**

檔案格式

點陣圖資料

- 每行必須是四的倍數(補成四)

每行的位元數 =  $((3 \times \text{width} + 3) / 4) \times 4$

# 再回到BMP

檔案格式

點陣圖資料



# 參考資料

- [點陣圖（Bitmap）檔案格式](#)