

# 程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH11  
檔案處理



# 本章綱要

11-1 簡介

11-2 資料階層

11-3 檔案和串流

11-4 建立一個循序存取檔案

11-5 由循序存取檔案讀出資料

11-6 隨機存取檔案

11-7 建立隨機存取檔案

11-8 隨機地寫入資料到隨機存取檔案

11-9 由隨機存取檔案讀出資料

11-10 範例：交易處理程式

# 11.1 簡介

## ■ 資料檔案

- C程式可以建立、更新和處理資料檔案
- 程式會利用檔案來長期保存大量的資料
  - 儲存在變數和陣列中的資料是暫時的，其存在記憶體內，重新開機後就會不見
  - 而檔案資料是儲存在磁碟、硬碟之中

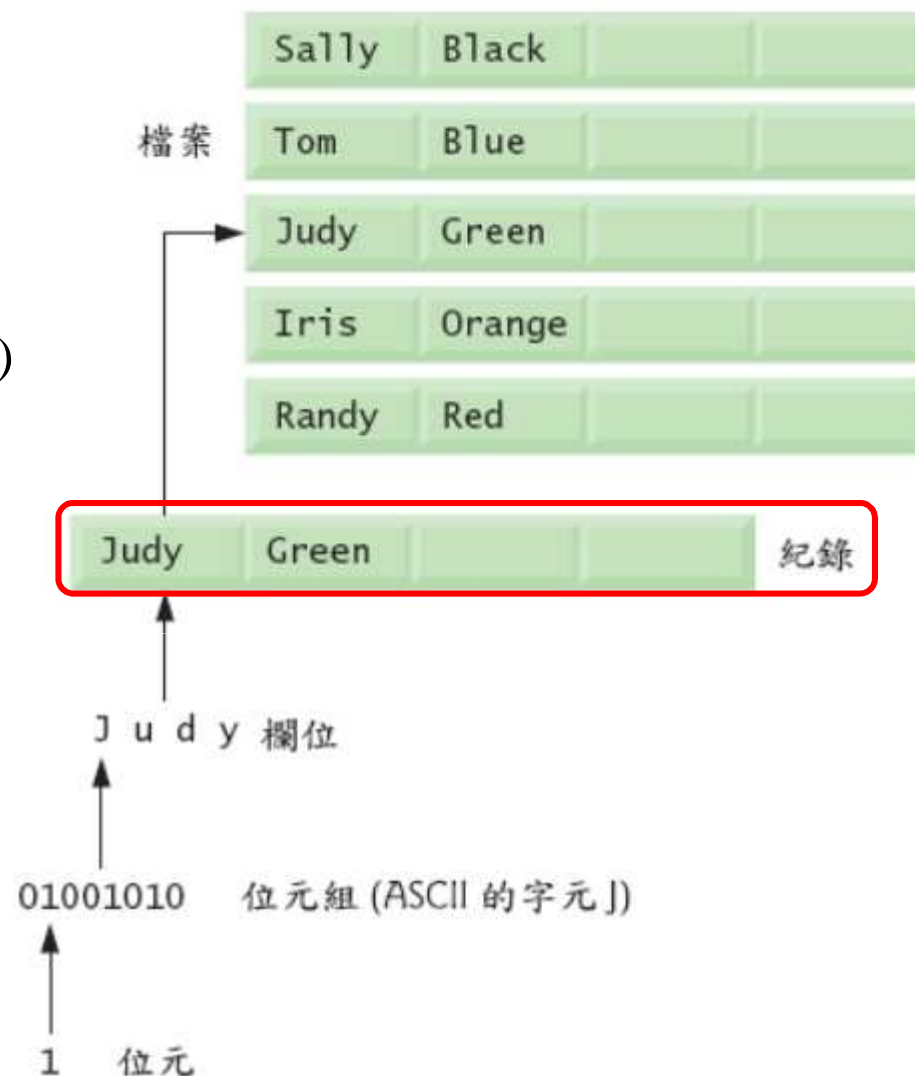
## 11.2 資料階層

### ■ 資料階層

- 資料庫 (一群互相關連的檔案)
- 檔案 (一群相關的紀錄)
- 紀錄 (一群相關的欄位，以 **struct** 或 **class** 表示)
- 欄位
- 位元組 (byte)
- 位元 (bit)

### ■ 資料檔案

- 紀錄**關鍵值** (key)
- 循序檔案 (以key來排序)



→ 想要擷取某個特定記錄，需由關鍵值 (某個) 來取得

## 11.3 檔案和串流

- C 將每個檔案視為連續的位元組串流

- 檔案末端具有檔案結尾記號 (end-of-file, **EOF**)

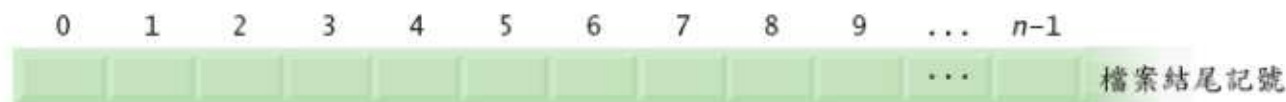


圖 11.2 C 將檔案視為具有  $n$  個位元組的資料

- 檔案開啓後就會有一個資料流(stream)

- 檔案的處理被視為一個串流(stream)，由眾多字元所組成的串流
- 提供程式與檔案之間的通訊管道
- 標準串流來源：
  - stdin – 標準輸入 (鍵盤)
  - stdout – 標準輸出 (螢幕)
  - stderr – 標準錯誤 (螢幕)

## ■ FILE 結構

- 程式設計人員需透過 **FILE** 來存取檔案串流的資料
- 處理檔案的動作交給 `stdio.h` 函式庫，裡面定義了 **FILE** 結構體

**#include <stdio.h>**

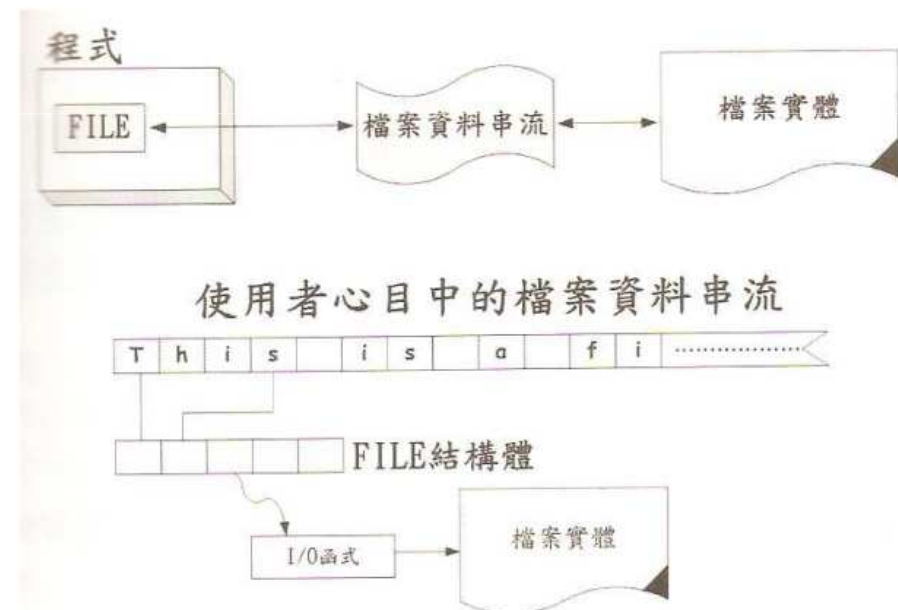
**FILE \*fptr;**

**fptr=fopen("a.txt", "r");**

→開啓一個檔案會傳回  
指向**FILE**結構的指標

### □ 檔案存取方式

- 循序存取 (**Sequential Access**)  
輸入的資料會按照先後次序存放
- 隨機存取 (**Random Access**)  
存放沒有先後順序，但每筆長度一樣空間



## ■ 標準程式庫的讀取/寫入函式

- `fgetc( )`: 從檔案中讀取一個字元

`fgetc(stdin)`

← 等於 `getchar( )`

- `fputc( )`: 可寫入一個字元於檔案中

`fputc('a', stdout)`

← 等於 `putchar( 'a' )`

- `fgets( )`: 可從檔案中讀取一行文字

- `fputs( )`: 將一行文字寫入指定檔案中

- `fscanf( )` / `fprintf( )`: 和`scanf`與`printf`相當的檔案處理函式

## 11.4 建立一個循序存取檔案

### ■ 循序檔案建立

- 開檔: **檔案指標 = fopen(“檔名”, “開啓模式”);**

回傳值為 NULL，代表檔案開啓錯誤

- 關檔: **fclose(檔案指標);**

回傳值為 -1 (關閉檔案發生錯誤)，回傳值為 0 (成功關閉檔案)

- 寫入: **fprintf(檔案指標, 寫入格式, ...);**

回傳值為 EOF (寫入錯誤)

- 例如:

```
#include <stdio.h>
int main(void) {
    FILE *fptr;
    fptr = fopen("data1.txt", "w");
    fprintf(fptr, "%10s  %3d\n", "May", 95);
    fprintf(fptr, "%10s  %3d\n", "John", 80);
    fclose(fptr);
    return 0;
}
```

r  
w  
a



模式	說明
r	開啓用來讀取的檔案
w	建立一個用來寫入的檔案。如果檔案已經存在，就會刪除已經存在的內容。
a	附加；開啓或是建立一個用來將資料寫到檔案結尾的檔案。
r+	開啓一個用來更新資料的檔案（可讀寫）。
w+	建立一個用來更新資料的檔案。如果檔案已經存在，就會刪除已經存在的內容。
a+	附加：開啓或是建立一個檔案，將資料寫到檔案尾端。
rb	以二進位模式開啓一個用來讀取的檔案。
wb	以二進位模式開啓一個用來寫入的檔案。如果檔案已經存在，程式就會刪除已經存在的內容。
ab	以二進位模式附加、開啓或是建立一個用來將資料寫到檔案結尾的檔案。
rb+	以二進位模式開啓一個用來更新資料的檔案（可讀寫）。
wb+	以二進位模式開啓一個用來更新資料的檔案。如果檔案已經存在，程式就會刪除已經存在的內容。
ab+	附加：以二進位模式開啓或是建立一個檔案，將資料寫到檔案尾端。

圖11.6 | 檔案開啟模式

- **範例**：建立應收帳款檔案(循序存取)，有客戶的帳號、名稱、餘額

```
1  /* Fig. 11.3: fig11_03.c */
3  #include <stdio.h>
5  int main( void )
6  {
7      int account;
8      char name[ 30 ];
9      double balance;
10
11     FILE *cfPtr;
12
13     if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL ) {
14         printf( "File could not be opened\n" );
15     }
16     else {
17         printf( "Enter the account, name, and balance.\n" );
18         printf( "Enter EOF to end input.\n" );
19         printf( "? " );
20         scanf( "%d%s%lf", &account, name, &balance );
21         while ( !feof( stdin ) ) {
22             fprintf( cfPtr, "%d %s %.2f\n", account, name, balance );
23             printf( "? " );
24             scanf( "%d%s%lf", &account, name, &balance );
25         }
26         fclose( cfPtr ); /* fclose closes file */
27     }
28     return 0;
29 }
```

FILE指標定義建立一個新的檔案指標

fopen 函式開啟檔案; w引數表示檔案是用來寫入的

到達檔案末端時，feof會回傳true

fprintf 將字串寫入檔案

fclose 關閉檔案



## 常見的程式設計錯誤 11.1

使用寫入方式（"w"）開啓某一個已經存在的檔案，但是使用者實際上希望保存這個檔案。將會在沒有警告的情形下刪除這個檔案。



## 常見的程式設計錯誤 11.2

在程式中參考某一個檔案之前，忘了先開啓這個檔案會產生邏輯錯誤。

- 判斷檔案結尾: **feof(stdin)**
  - 回傳值為非零的值 (檔案結尾)
  - 回傳值為0 (還沒檔案結尾)

作業系統	按鍵組合
Linux/Mac OS X/UNIX	<Ctrl> d
Windows	<Ctrl> z

```
Enter the account, name, and balance.  
Enter EOF to end input.  
? 100 Jones 24.98  
? 200 Doe 345.67  
? 300 White 0.00  
? 400 Stone -42.16  
? 500 Rich 224.62  
? ^Z
```

圖 11.4 幾種電腦系統中代表檔案結尾的按鍵組合

# 1 使用者可以存取到的範圍

`cfPtr = fopen("clients.dat", "w");`  
fopen回傳指向 FILE 結構的指標  
(定義在 `<stdio.h>` 中)

cfPtr



## 2

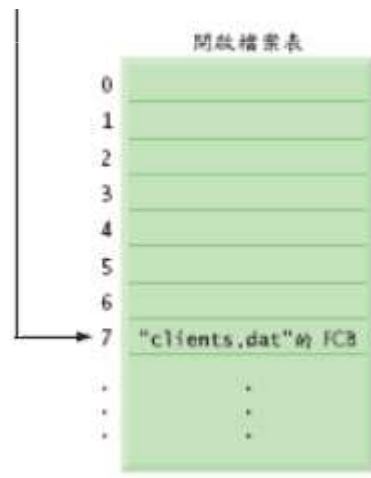
"clients.dat" 的 FILE 結構  
包含一個描述子，也就是一個小的整數，它是指向  
開啟檔案表的索引



## 3

當程式發出 I/O 呼叫，像是  
`fprintf(cfPtr, "%d %s %.2f",`  
`account, name, balance);`

程式會在 FILE 結構中找到  
描述子(7)，並使用此描述  
子在開啟檔案表中找到 FCB



## 4

此程式會呼叫一個作業系統  
服務，此服務會使用 FCB 中的  
資料來控制所有到磁碟機  
中真正檔案的輸入及輸出。

當檔案開啟時，此項目會從  
磁碟上的 FCB 被複製出來

# 練習

- 試著產生10 個1~46 之間的整數亂數，並寫入純文字檔 (“rand.txt”)中

# 11.5 由循序存取檔案讀出資料

## ■ 讀取循序檔資料

**fscanf(檔案指標, 讀入格式, 存放位置);**

回傳值為 正整數，代表讀取成功

回傳值為 EOF，讀到檔案盡頭或發生錯誤

□ 例如:

```
#include <stdio.h>
int main(void) {
    char name[10];
    int grade;
    FILE *fptr;
    if((fptr = fopen("data1.txt", "r"))!=NULL){
        fscanf(fptr, "%s%d", name, &grade);
        fclose(fptr);
    }
    return 0;
}
```

□ 從檔案開端到尾部循序地讀取資料

□ 設定指標重新回到檔案開頭: **rewind(檔案指標);**



## ■ 範例：讀取並輸出循序檔資料

```
1  /* Fig. 11.7: fig11_07.c */
3  #include <stdio.h>
5  int main( void )
6  {
7      int account;
8      char name[ 30 ];
9      double balance;
11     FILE *cfPtr;
14     if ( ( cfPtr = fopen( "clients.dat", "r" ) ) == NULL ) {
15         printf( "File could not be opened\n" );
16     }
17     else {
18         printf( "%-10s%-13s%\n", "Account", "Name", "Balance" );
19         fscanf( cfPtr, "%d%s%f", &account, name, &balance );
22         while ( !feof( cfPtr ) ) {
23             printf( "%-10d%-13s%7.2f\n", account, name, balance );
24             fscanf( cfPtr, "%d%s%f", &account, name, &balance );
25         }
27         fclose( cfPtr );
28     }
30     return 0;
32 }
```

Account	Name	Balance
100	Jones	24.98
200	Doe	345.67
300	White	0.00
400	Stone	-42.16
500	Rich	224.62

**fopen** 函式會開啟檔案  
**r**指檔案是用來讀取的

**fscanf**函式從檔案中讀出一個字串

## ■ 範例：信用查詢程式

```
Enter request
1 - List accounts with zero balances
2 - List accounts with credit balances
3 - List accounts with debit balances
4 - End of run
? 1

Accounts with zero balances:
300      White      0.00

? 2

Accounts with credit balances:
400      Stone     -42.16

? 3

Accounts with debit balances:
100      Jones     24.98
200      Doe       345.67
500      Rich      224.62

? 4
End of run.
```

```
1  /* Fig. 11.8: fig11_08.c */
3  #include <stdio.h>
6  int main( void )
7  {
8      int request;
9      int account;
10     double balance;
11     char name[ 30 ];
12     FILE *cfPtr;
15     if((cfPtr=fopen("clients.dat", "r"))==NULL){
16         printf( "File could not be opened\n" );
17     }
18     else {
21         printf( "Enter request\n"
22             " 1 - List accounts with zero balances\n"
23             " 2 - List accounts with credit balances\n"
24             " 3 - List accounts with debit balances\n"
25             " 4 - End of run? " );
26         scanf( "%d", &request );
27
```

先開檔



```

28      /* process user's request */
29      while ( request != 4 ) {
32          fscanf( cfPtr, "%d%s%f", &account, name, &balance );
34          switch ( request ) {
36              case 1:
37                  printf( "\nAccounts with zero balances:\n" );
40                  while ( !feof( cfPtr ) ) {
42                      if ( balance == 0 ) {
43                          printf( "%-10d%-13s7.2f\n", account, name, balance );
45                      }
48                      fscanf( cfPtr, "%d%s%f", &account, name, &balance );
50                  }
52                  break;
54              case 2:
55                  printf( "\nAccounts with credit balances:\n" );
58                  while ( !feof( cfPtr ) ) {
60                      if ( balance < 0 ) {
61                          printf( "%-10d%-13s7.2f\n", account, name, balance );
63                      }
66                      fscanf( cfPtr, "%d%s%f", &account, name, &balance );
68                  }
70                  break;

```

選項1, 顯示欠款0元的帳戶

選項2, 顯示借款的帳戶

```

72         case 3:
73             printf( "\nAccounts with debit balances:\n" );
76             while ( !feof( cfPtr ) ) {
78                 if ( balance > 0 ) {
79                     printf( "%-10d%-13s%7.2f\n", account, name, balance );
81                 }
84                 fscanf( cfPtr, "%d%s%f", &account, name, &balance );
86             }
88             break;
90         }
92         rewind( cfPtr );
93
94         printf( "\n? " );
95         scanf( "%d", &request );
96     }
98     printf( "End of run.\n" );
99     fclose( cfPtr );
100 }
102 return 0;
104 }

```

選項1, 顯示有存款的帳戶

rewind函式將檔案指標移到檔案的開端

/\*可更改為 fputc(ch, fptr2);\*/

## ■ 檔案字元的讀取

- 讀取單一字元  
回傳值 = **fgetc**( 檔案指標 );
- 寫入單一字元  
回傳值 = **fputc**( 字元, 檔案指標 );
- 回傳值為**EOF**，代表讀到檔案盡頭

```
FILE *fptr;  
char ch;  
int count=0;  
fptr=fopen("c:\\student.txt", "r");  
if(fptr!=NULL){  
    while((ch=fgetc(fptr))!=EOF){  
        printf("%c", ch);  
        count++;  
    }  
    fclose(fptr);  
}  
else  
    printf("開檔失敗!");  
return 0;
```

## ■ 檔案字串的讀取

- 讀取一段文字  
回傳值 = **fgets**(字串, 長度, 檔案指標 );
- 寫入一段文字  
回傳值 = **fputs**(寫入字串, 檔案指標 );
- 回傳值為NULL, 代表讀到檔案盡頭  
回傳值為EOF, 代表寫入錯誤

/\*可更改為 fputs(tempstr, fptr2); \*/

```
FILE *fptr;  
char tempstr[80];  
fptr=fopen("c:\\student.txt", "r");  
if(fptr!=NULL){  
    while((fgets(tempstr, 80, fptr))!=NULL)  
        printf("%s", tempstr);  
    fclose(fptr);  
}  
else  
    printf("開檔失敗!");  
return 0;
```

# 練習

- 利用 `fscanf` 與 `fprintf` 函式，將檔案 `copy` 到另一個檔案中，如下：

File1	File2
AM990001 89 84 75	AM990001
AM990002 77 69 87	89
AM990003 65 68 77	84
	75
	AM990002
	77
	...

接著，試著將上述平均分數不及格的學生學號寫入  
“`NoPass.txt`”檔案，及格者的學號寫入 “`Pass.txt`”檔案