

C/C++資料結構與程式設計

檔案處理

NTU CSIE

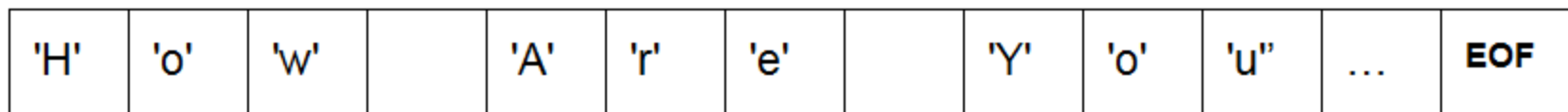
講師：張傑帆

檔案函式簡介

- C語言所提供的檔案處理函式，將這些輸入的資料儲存在指定的資料檔(Data File)中，這些資料檔的副檔名通常設為 *.txt或 *.dat。
- 當需要處理資料檔中的資料時，可以透過檔案處理函式，取得資料檔內的資料並存放到變數或陣列(主記憶體中)中進行處理，資料處理完畢再將資料寫回指定的資料檔內。
- 這樣資料不但容易維護，而且同一份程式可處理相同資料格式不同的資料檔，增加了程式的使用彈性。

資料流的觀念

- 「**資料流**」就是將整個資料檔內的資料如下圖視為一串連續的字元組成。
- 像MS-DOS和UNIX處理資料都是透過資料流觀念，存取方式採一個字元一個字元方式處理來簡化高階輸出入動作，使得任何輸出入的動作都在資料流上運作，如此在撰寫資料檔內資料的輸出入程式時，可以不用理會週邊裝置的多樣性。
- 資料流藉著檔案指標的移動來存取資料流上的資料，檔案指標目前所指的資料位置即是要處理的資料，經過存取後檔案指標會往後移動。
- 每個資料檔最後面都有一個檔案結束符號(EOF)用來告知該資料檔到此結束後面沒有資料，若檔案指標指到EOF便表示資料已經存取完畢。




資料流可視為一串連續的字元

- 電腦週邊設備資料的存取可以使用 **二進位檔(Binary File)**和 **文字檔(Text File)**這兩種格式。
- 二進位檔內的資料是採用二進位格式進行儲存，例如聲音檔(*.wav)、影像檔(*.avi)、執行檔(*.exe)、圖形檔(*.pcx)都是採用二進位格式儲存檔案，採用二進位格式來儲存檔案會比較節省儲存空間。
- 文字檔內的每個字元都是使用該字元對應ASCII碼的格式來進行存檔，一個字元的ASCII碼就佔用1 Byte，10 個字元就佔用10 Bytes，因此採用文字檔的格式進行存檔會比較浪費儲存空間。

循序檔與隨機檔

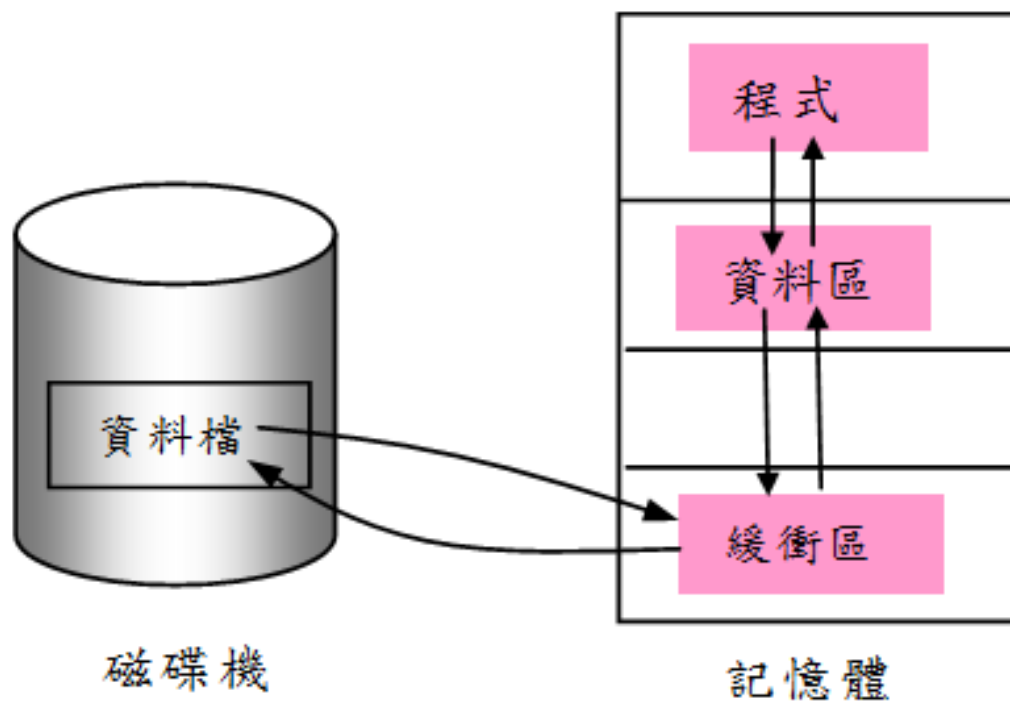
- 循序檔(Sequential File)全名為循序存取檔(Sequential Access File)。
- 此種資料在寫入時是從資料檔的開頭開始按照資料寫入的先後順序逐筆寫入磁碟中，讀取時不管資料的前後次序都必須從第一筆開始逐筆往後讀取，一直到檔案結束為止。
- 此種檔案每筆資料長度不拘，較節省磁碟空間，但查詢資料時都必須從第一筆開始找起較費時間。

- 
- ➡ 隨機檔(Random File)全名為隨機存取檔(Random Access File)。
 - ➡ 由於此種檔案大都使用結構方式來存放資料，因此每筆資料的長度是相同。
 - ➡ 因此存取此種資料不管資料是位於最前面、中間或最後面，都是經過計算位置便可直接存取該筆資料，而不必像循序檔找尋資料時都必須從最開頭開始找起，而是採跳躍式直接存取。

C語言檔案處理函式

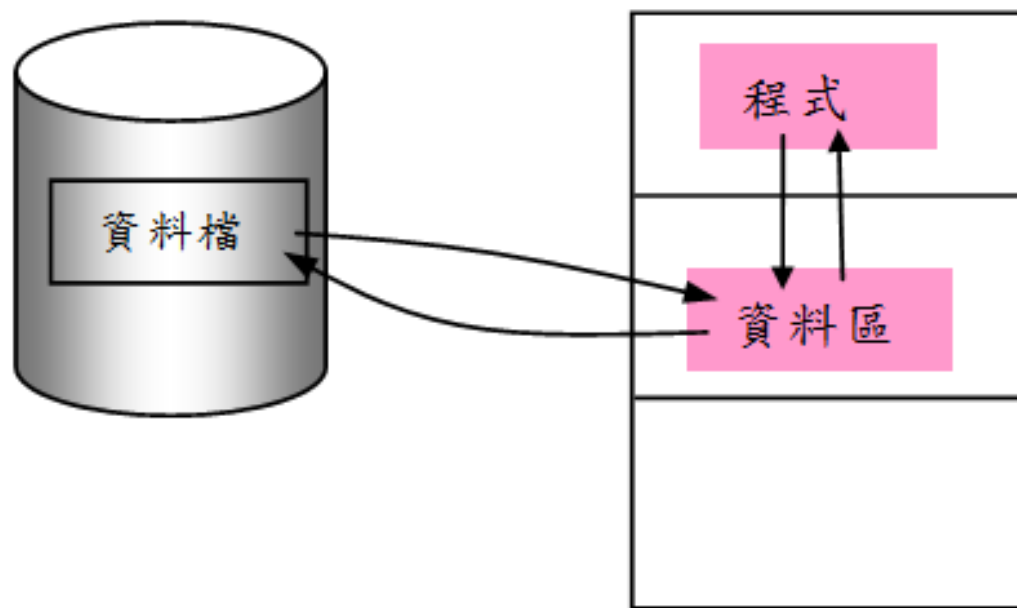
- C語言的檔案處理函式按照有無提供緩衝區可區分為：提供緩衝區的標準輸出入(Standard I/O)函式，以及無提供緩衝區的系統輸出入(System I/O)函式。
- 所謂「緩衝區」(Buffer)即是指資料在存取時，為了加快程式執行時的速度，會在主記憶體中事先建立一塊緩衝區來存放部分資料，接著再透過這個緩衝區來存取整塊資料，而不直接和磁碟進行存取的動作。

標準輸出入(Standard I/O)函式



圖一

系統輸出入(System I/O)函式



圖二

記憶體

標準輸出入函式簡介

- 標準輸出入函式產生的資料檔，當您使用`fopen()`函式開啟檔案成功時會傳回一個具有`FILE`檔案型別的結構，`FILE`結構內存放著一些和該資料檔開啟的相關資訊。


- 有關該檔案的位置，緩衝區的長度及其在記憶體中的位址等等，設計程式時可以不必去理會這些資訊。
- 只要記得使用這些標準輸出入函式之前要先宣告 `FILE *fptr;`，`fptr`是檔案指標，它所指到的是具有FILE檔案型別結構。將此fptr檔案指標指到fopen()函式所傳回FILE檔案型別結構的位址，若該資料檔開啟成功，即可對開啟的檔案做相對應的處理。
- 其宣告方式如下：

函式	功能說明
fopen	語法：FILE *fopen(const char *filename, const char *mode) 功能：使用 mode 模式來開啟指定的 filename 資料檔。若開啟檔案成功會傳回一個具有 FILE 檔案結構的指標；若開啟失敗，則會傳回 NULL。
fclose	語法：int fclose(FILE *fptr) 功能：將檔案指標 fptr 所指的檔案關閉。若傳回 0 表示關檔成功。

函式	功能說明
fopen	<p>語法：FILE *fopen(const char *filename, const char *mode)</p> <p>功能：使用 mode 模式來開啟指定的 filename 資料檔。若開啟檔案成功會傳回一個具有 FILE 檔案結構的指標；若開啟失敗，則會傳回 NULL。</p>
fclose	<p>語法：int fclose(FILE *fptr)</p> <p>功能：將檔案指標 fptr 所指的檔案關閉。若傳回 0 表示關檔成功。</p>
fgetc	<p>語法：int fgetc(FILE *fptr)</p> <p>功能：讀取檔案指標 fptr 所指位置中的字元後，檔案指標即往後移一個字元位置，若檔案指標已指到 EOF(檔案結束符號)，則傳回-1。</p>
fputc	<p>語法：int fputc(int ch, FILE *fptr)</p> <p>功能：將字元 ch 寫入檔案指標 fptr 所指向檔案的位置。</p>
fgets	<p>語法：char *fgets(char *str, int n, FILE *fptr)</p> <p>功能：在檔案指標 fptr 所指檔案位置讀取 n 個字元並放入 str 字串(即 str 指標所指的位址)。</p>

fputs	語法：int fputs(const char *str, FILE *fptr) 功能：將字串 str(即 str 指標所指位址內的字串)寫入檔案指標 fptr 所指向檔案的位置。
fscanf	語法：int fscanf(FILE *fptr, "格式化字串", [引數串列]) 功能：以指定的格式由緩衝區中 fptr 所指位址讀取資料並放入引數串列對應的變數，適用循序檔。
fprintf	語法：int fprintf(FILE *fptr, "格式化字串", [引數串列]) 功能：以指定的格式將引數串列對應的變數資料分別寫入至 fptr 指標所指的檔案，適用於循序檔。
rewind	語法：void rewind(FILE *) 功能：檔案指標移到檔案的最開頭。
fseek	語法：int fseek(FILE *fptr, long offset, int whence) 功能：檔案指標由 whence 移動 offset Bytes。

fwrite	<p>語法：size_t fwrite(const void *ptr, size_t size, size_t n, FILE *fptr)</p> <p>功能：將 ptr 結構指標所指的資料區塊內一次讀取 n 筆，每筆資料大小為 size Bytes 寫入至 fptr 檔案指標所指的資料檔內。若寫入成功，會傳回所寫入的結構筆數且 fptr 檔案指標會往後移動 n 筆結構的長度(即 n x size 大小)。此函式適用於隨機檔。</p>
fread	<p>語法：size_t fread(void *ptr, size_t size, size_t n, FILE *fptr)</p> <p>功能：由 fptr 檔案指標所指的檔案內讀取 n 筆記錄至 ptr 結構指標所指的結構內，讀取時必須指定結構的大小 size。若讀取成功，傳回所讀取的結構筆數且檔案指標會往後移動 n 筆結構的長度(即 n x size 的大小)，若讀不到資料，則會傳回 0，此函式適用於隨機檔。</p>
feof	<p>語法：int feof(FILE *)</p> <p>功能：判斷檔案指標是否已經指到檔案結尾符號(EOF)。若傳回 0(NULL)表示檔案尚未結束，傳回非 0 值表示檔案已經結束。</p>
fileno	<p>語法：int fileno(FILE *)</p> <p>功能：傳回所開啟檔案對應的檔案處理代碼(handle_no)。</p>

- 
- 在系統輸出入函式中提供filelength()
函式可傳回檔案代碼(handle_no)所對應檔案的檔案大小，它是以Bytes為單位，此函式原型宣告於io.h標頭檔內。

如何開啟檔案

欲開啟指定的檔案可使用 `fopen()` 函式，其語法如下：

語法：`FILE *fopen(const char *filename, const char *mode)`

功能：使用 `mode` 模式來開啟指定的 `filename` 檔案。若開啟檔案成功會傳回一個 `FILE` 結構的指標；開啟檔案失敗則會傳回 `NULL`。

說明

1. 檔案若開啟成功，電腦自動賦予一個屬於這個檔案的 `FILE` 結構型別的指標，`FILE` 結構內所宣告的成員會記錄所開啟檔案的相關資訊，透過這些結構成員可以取得所開啟檔案的檔案大小、緩衝區大小、資料緩衝區的位址...等相關訊息。若檔案開啟失敗，則傳回 `NULL(0)`。
2. `fopen()` 函式的第一個引數 `filename` 用來設定欲開啟的資料檔。若開啟的資料檔與執行檔在相同路徑(資料夾)下只寫檔名即可。若資料檔和程式檔分別存放在不同的資料夾，就必須指定完整路徑和檔名。

[例 1] 開啟一個和執行檔在相同路徑下 sample.txt 文字資料檔。

```
FILE fptr;  
fptr=fopen("sample.txt", "r");
```

[例 2] 開啟一個位於 C 磁碟 test 資料夾下的 sample.txt 文字資料檔。

```
FILE fptr;  
fptr=fopen("C:\\test\\sample.txt", "r"); //方法一  
fptr=fopen("C:/test/sample.txt", "r"); //方法二
```

指定路徑時如方法 1，有用到「\」符號，由於這個符號在 C 語言中被視為逸出字元，因此必須在該符號前面再插入一個「\」倒斜線，變成「\\」。或者如方法 2 改用「/」斜線。

3. `fopen()`函式的第二個引數 `mode` 用來設定欲開啟資料檔的類型和指定資料檔的存取模式。第二個引數必須為字串格式，頭尾必須使用雙引號括住，為增加程式的彈性也可改用字串變數。存取模式如下表：

模式	功能說明
"w"	開啟一個可以寫入資料的檔案。若檔案存在，則該檔內的資料都會被刪除變成一個空檔；如果檔案不存在，則會自動建立一個新檔。
"a"	開啟一個可以寫入資料的檔案。若檔案存在，則新增的資料會附加在該檔案的最後面；如果檔案不存在，則會自動建立一個新檔。
"r"	開啟一個已經存在的檔案，若檔案不存在，則 <code>fopen()</code> 函式會傳回 <code>NULL</code> 。
"w+"	開啟一個可以讀寫資料的檔案。若檔案存在，則該檔內的資料都會被刪除變成一個空檔；如果檔案不存在，則會自動建立一個新檔。
"a+"	開啟一個可以讀寫資料的檔案。若檔案存在，則新增的資料會附加在該檔案的最後面；如果檔案不存在，則會自動建立一個新檔。

模式	功能說明
"r+"	開啟一個可以讀寫資料的已存在檔案，若檔案不存在，則 fopen() 函式會傳回 NULL。
"wb+"	開啟一個可以讀寫資料的二進位檔案。若檔案存在，則該檔內的資料都會被刪除變成一個空檔；如果檔案不存在，則會自動建立一個新檔。
"ab+"	開啟一個可以讀寫資料的二進位檔案。若檔案存在，則新增的資料會附加在該檔案的最後面；如果檔案不存在，則會自動建立一個新檔。
"rb+"	開啟一個可以讀寫資料的已存在二進位檔案，若檔案不存在，則 fopen() 函式會傳回 NULL。

4. 檔案指標的使用：

通常在讀寫資料檔時都要先判斷檔案是否開啟成功，如此才可以正確的讀寫檔案。一般標準輸入函式讀檔步驟如下：

Step 01 先宣告一個具有 FILE 檔案結構的 fptr 指標

```
FILE *fptr;
```

Step 02 使用 fopen()函式指定欲開啟的 C 磁碟 gotop 資料夾下的 sample.txt 檔案，開啟模式設為 "r+" 表示要開啟一個已存在可讀寫的資料檔。

```
fptr = fopen("C:\\gotop\\sample.txt", "r+");
```

Step 03

若檔案開啟失敗，會傳回 0(NULL)。可使用 if 選擇敘述來判斷是否開啟失敗。

```
if(fptr==NULL)    /* 判斷檔案是否開啟失敗 */  
{  
    printf("開檔失敗! .... ");  
    exit(0);      /* 結束程式 */  
}
```

.....
若開啟成功繼續往下執行

若檔案開啟失敗，則結束程式或要求使用者重新指定要開啟的資料檔；若開啟檔案成功，則 fptr 指標會指到資料檔的最開頭，並在主記憶體中配置一個緩衝區以供使用，此時您可直接進行處理檔案內資料的讀寫工作。

如何關閉檔案

- 檔案開啟後若不再繼續使用，要記得使用`fclose()`函式將指定的檔案關閉，並將FILE檔案結構指標的相關資源，及所佔用的緩衝區歸還給系統。
- 若採讀取模式來開啟檔案，可以不必做關閉檔案的動作。
- 若是採寫入模式一定記得要使用`fclose()`函式關閉檔案，否則最後放在緩衝區的資料便無法寫回資料檔內，而發生資料遺失的情形。

語法：`int fclose(FILE *fptr)`

功能：將檔案指標 `fptr` 所指的資料檔關閉。若傳回 0 表示關檔成功。

如何取得檔案大小

- 想知道某一個檔案的大小，可以使用 `fileno()` 和 `filelength()` 函式來取得。

語法：`int fileno(FILE *fptr)`

標頭檔：`stdlib.h`

功能：傳回所開啟檔案指標 `fptr` 對應的檔案代碼。當開啟檔案成功後，作業系統會自動給予一個號碼，此號碼用來代表所開啟的檔案。

語法：`long filelength(int handle_no)`

標頭檔：`io.h`

功能：傳回檔案代碼對應檔案的檔案大小，以 Bytes 為單位。

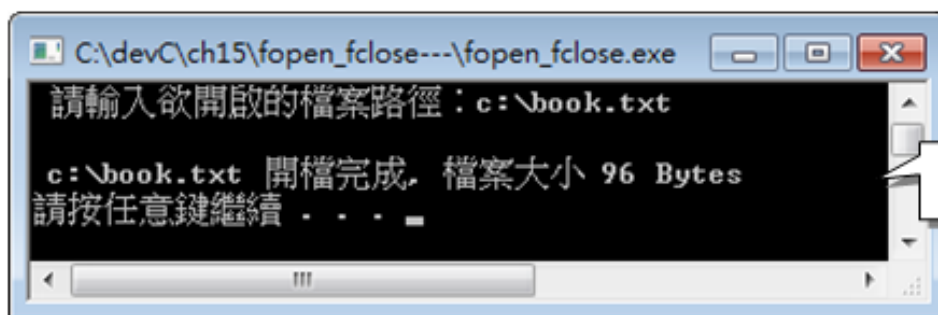
[例] 如何取得 C:\gotop 資料夾下的 sample.txt 資料檔的長度，其寫法如下：

```
FILE fptr ;  
int fno, fsize ;  
fptr=fopen("C:\\gotop\\sample.txt", "r"); /* 開啟檔案 */  
fno=fopen(fptr); /* 傳回檔案代碼並指定給 fno */  
fsize=filelength(fno); /* 傳回檔案代碼對應檔案的大小並指定給 fsize */  
fclose(fptr); /* 關閉檔案 */
```


範例：

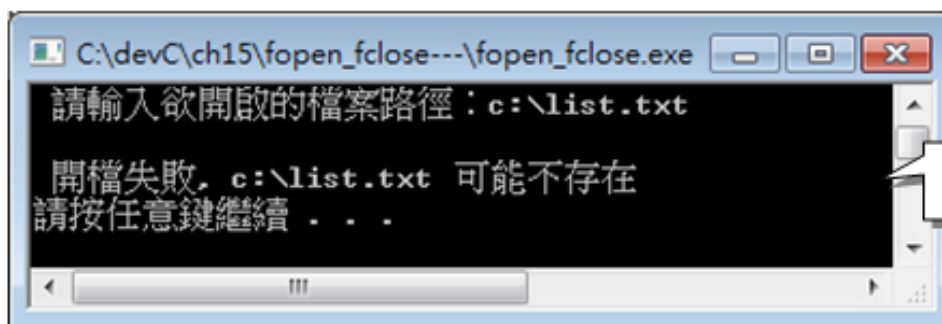
由鍵盤輸入欲開啟的檔案，若檔案開啟成功，則顯示該檔案的大小，以 Bytes 為單位；若開啟檔案失敗，則顯示「開檔失敗…」訊息。

執行結果



```
C:\devC\ch15\lopen_fclose---\lopen_fclose.exe
請輸入欲開啟的檔案路徑：c:\book.txt
c:\book.txt 開檔完成, 檔案大小 96 Bytes
請按任意鍵繼續 . . .
```

開啟檔案成功



```
C:\devC\ch15\lopen_fclose---\lopen_fclose.exe
請輸入欲開啟的檔案路徑：c:\list.txt
開檔失敗, c:\list.txt 可能不存在
請按任意鍵繼續 . . .
```

開啟檔案失敗

```
08  FILE *fptr;
09  char fpath[LEN];
10  int fno, fsize;
11  printf(" 請輸入欲開啟的檔案路徑：");
12  gets(fpath);          /* 使用者輸入欲開啟的檔案並指定給 fpath */
13  fptr=fopen(fpath, "r"); /* 開啟已經存在的檔案 */
14  if(fptr==NULL)        /* 判斷是否開檔失敗 */
15  {
16      printf("\n 開檔失敗， %s 可能不存在\n", fpath);
17      system("PAUSE");
18      exit(0);          /* 結束程式 */
19  }
20  fno=fileno(fptr);      /* 取得檔案代碼 */
21  fsize=filelength(fno); /* 取得檔案大小，以 Byte 為單位 */
22  printf("\n %s 開檔完成， 檔案大小 %d Bytes\n", fpath, fsize);
23  fclose(fptr);
```

如何以字元讀寫檔案

- ➡ 若想要從資料檔讀取一個字元，可使用 `fgetc()` 輸入函式。
- ➡ 反之，欲將一個字元寫入檔案，可使用 `fputc()` 輸出函式，其語法分別如下：

語法：`int fgetc(FILE *fptr)`

功能：讀取檔案指標 `fptr` 目前所指向資料檔位置中的字元後，讀取完畢檔案指標並往下移一個字元位置，若檔案指標已經指到檔案結尾，會傳回 -1。

語法：`int fputc(int ch, FILE *fptr)`

功能：將字元 `ch` 寫入檔案指標 `fptr` 所指向檔案的位置。

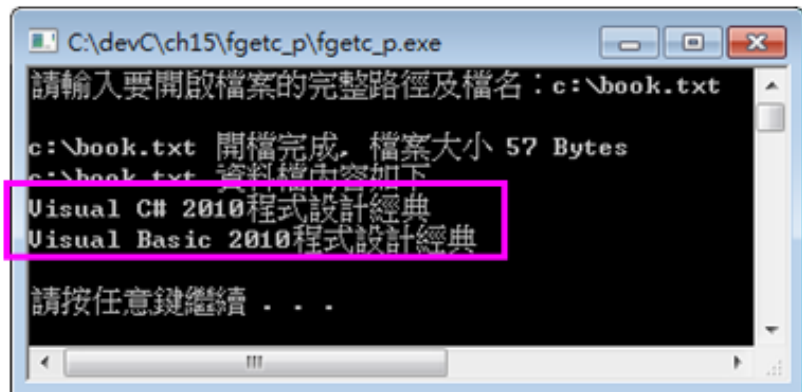
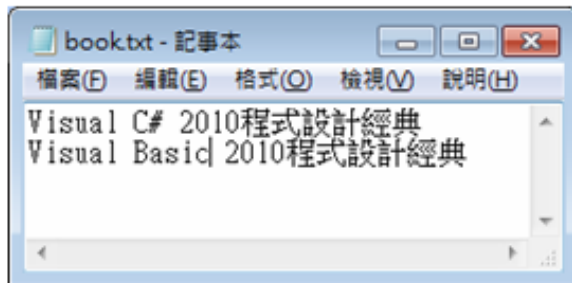


範例

練習使用 `fgetc()` 函式來讀取指定的資料檔。

執行結果

1. 程式執行時會要求您輸入要開啟檔案的完整路徑及檔名，開檔成功時會顯示所開啟檔案的大小，且會將資料檔內所有資料讀出並顯示在螢幕上。
2. 開檔失敗會顯示「開檔失敗，XXX 可能不存在」的訊息。
3. 如左圖 `book.txt` 資料檔的內容，如右圖是當本例指定要開啟 `book.txt` 檔，此時即會印出 `book.txt` 內的所有資料。



Step 01 將使用者輸入檔案路徑名稱放入 `fpath`，接著建立一個可以讀取 `fpath` 資料檔的檔案指標 `fptr`。

Step 02 判斷檔案指標 `fptr` 是否為 `NULL`，若為 `NULL` 表示開檔失敗，此時結束程式的執行；若 `fptr` 不等於 `NULL`，則繼續下一步驟。

Step 03 開檔成功時在螢幕上顯示所開啟檔案的大小。

Step 04 透過 `while` 迴圈及 `fgetc()` 函式來讀取資料檔內的所有資料並顯示在螢幕上，其方式如下：

1. 使用 `fgetc()` 函式讀出檔案指標目前所指的字元並放入 `ch` 字元變數內，接著再判斷檔案指標是否未指到 `EOF`。若檔案指標未指到 `EOF` 則跳到 ②；否則離開迴圈跳到 Step5。
2. 印出 `ch` 字元後，再執行 ①。

Step 05 使用 `fclose()` 函式關閉檔案。

```
08  FILE *fptr;
09  char fpath[LEN] ;
10  int fno, fsize;
11  char ch;
12  printf("請輸入要開啟檔案的完整路徑及檔名：");
13  gets(fpath);
14  fptr=fopen(fpath, "r");
15  if(fptr==NULL)
16  {
17      printf("\n 開檔失敗, %s 可能不存在\n", fpath);
18      system("PAUSE");
19      exit(0);
20  }
21  fno=fileno(fptr);
22  fsize=filelength(fno);
23  printf("\n%s 開檔完成, 檔案大小 %d Bytes\n", fpath, fsize);
24  printf("%s 資料檔內容如下\n", fpath);
25  while((ch=fgetc(fptr))!=EOF)
26  {
27      printf("%c", ch);
28  }
29  fclose(fptr);
```

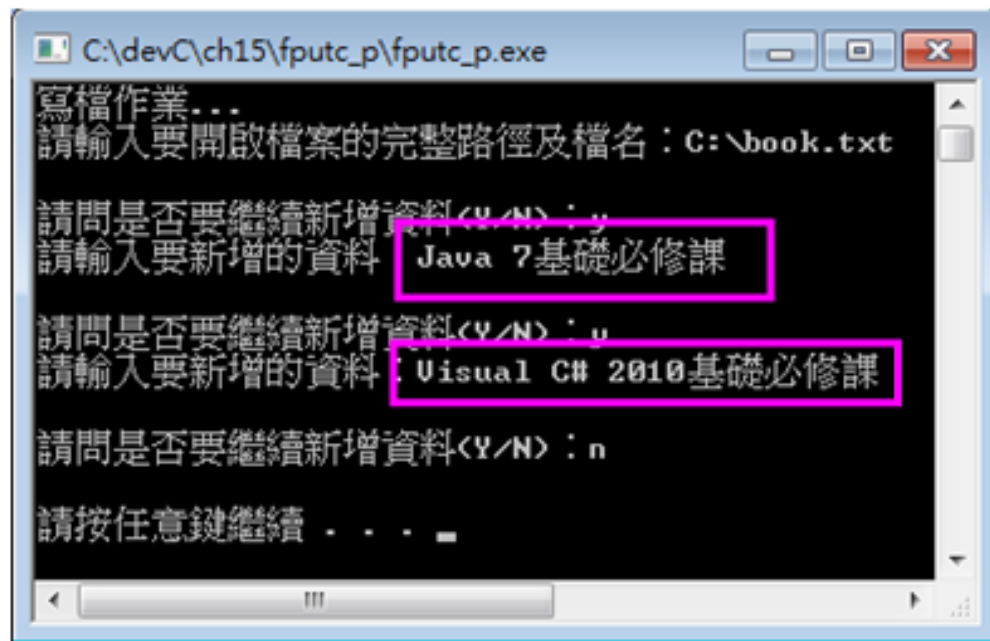


範例：

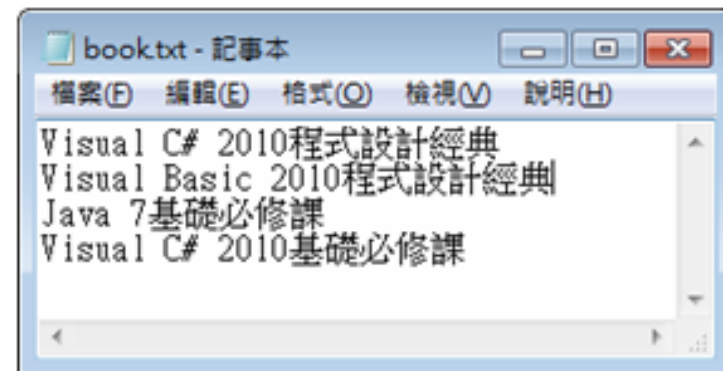
練習使用 `fputc()` 函式將鍵盤輸入的資料寫入指定的資料檔內。

執行結果

1. 程式執行時會要求您輸入要開啟檔案的完整路徑及檔名，開檔成功時會顯示所開啟檔案的大小。開檔失敗會顯示「開檔失敗，XXX 可能不存在」的訊息。
2. 若開檔成功時會詢問您是否要新增資料，若按 'y' 或 'Y' 鍵則可讓您由鍵盤輸入的資料寫入資料檔內，若按其他鍵則不寫入資料到資料檔內。如左下圖我們指定在 `c:\book.txt` 內新增了「Java 7 基礎必修課」及「Visual C# 2010 基礎必修課」資料，結果如右下圖使用記事本查詢資料已正確的寫入 `c:\book.txt` 資料檔內。



```
C:\devC\ch15\fputc_p\fputc_p.exe
寫檔作業...
請輸入要開啟檔案的完整路徑及檔名：C:\book.txt
請問是否要繼續新增資料(Y/N)：y
請輸入要新增的資料：Java 7基礎必修課
請問是否要繼續新增資料(Y/N)：y
請輸入要新增的資料：Visual C# 2010基礎必修課
請問是否要繼續新增資料(Y/N)：n
請按任意鍵繼續 . . .
```



```
book.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Visual C# 2010程式設計經典
Visual Basic 2010程式設計經典
Java 7基礎必修課
Visual C# 2010基礎必修課
```


Step 01 將使用者輸入檔案路徑名稱放入 `fpath`，再建立一個可以寫入附加資料到 `fpath` 資料檔的檔案指標 `fptr`。

Step 02 判斷檔案指標 `fptr` 是否為 `NULL`？若為 `NULL` 表示開檔失敗，此時結束程式的執行；若 `fptr` 不等於 `NULL`，則繼續下一步驟。

Step 03 透過 `while` 無窮迴圈讓使用者可以持續的由鍵盤輸入要寫入資料檔的資料。其方式如下：

1. 詢問是否要新增資料到資料檔中？按鍵盤的 'y' 或 'Y' 會跳到 ②；按其他鍵則跳到 ③。
2. 將使用者輸入的字串放入 `keyin` 字元陣列，然後將 `keyin` 字元陣列中的字元依序取出，再使用 `fputc()` 函式將取出的字元寫入到 `fptr` 所指向的資料檔。等到 `keyin` 所有的字元全部寫入到資料檔時即跳回到 ①。
3. 使用 `fclose()` 函式關閉檔案，此時將緩衝區內的資料寫入資料檔，並離開無窮迴圈，執行 Step4。

Step 04 結束程式。

程式碼 FileName : fputc_p.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <conio.h>
06 #include <io.h>
07 #define LEN 100
08
09 int main(int argc, char *argv[])
10 {
11     FILE *fptr;
12     char fpath[LEN], keyin[LEN];
13     int fno, fsize, i;
14     char ch;
15     printf("寫檔作業...\n");
16     printf("請輸入要開啟檔案的完整路徑及檔名：");
17     gets(fpath);
18     fptr=fopen(fpath, "a+");
19     if(fptr==NULL)
20     {
21         printf("\n 開檔失敗, %s 可能不存在\n", fpath);
22         exit(0);
23     }
```

```
24  while(1)
25  {
26      printf("\n 請問是否要繼續新增資料 (Y/N) : ");
27      if(toupper(getche())=='Y')
28      {
29          printf("\n 請輸入要新增的資料 : ");
30          gets(keyin);
31          for(i=0; i<strlen(keyin) ; i++)
32          {
33              fputc(keyin[i], fptr);
34          }
35          fputc('\n', fptr);
36      }
37      else
38      {
39          fclose(fptr);
40          break;
41      }
42  }
43  printf("\n\n");  system("PAUSE");
44  return 0;
45 }
```



範例：fgetc_fputc.c

將 fget_p.c 及 fputc_p.c 兩個範例整合，使製作的範例可以讀寫指定的資料檔。

執行結果

1. 程式執行時可讓您選擇兩個功能，按 '1' 表示要開新檔；按 '2' 或其他按鍵表示要開舊檔。接著要求輸入欲開新檔或開舊檔的完整路徑。
2. 開檔成功後會印出檔案路徑和檔案大小。
3. 接著會詢問您是否要輸入資料。按 'y' 或 'Y'，即可讓您新增資料，新增的資料會加在舊資料的後面；若按 'n', 'N' 或按其他鍵，即停止繼續新增資料。最後會印出檔案的路徑、檔案的大小、檔案的內容。

C:\devC\ch15\fgetc_fputc\fgetc_fputc.exe

請選擇項目->1.開新檔 2.開舊檔:2
請輸入欲開啟的檔案路徑:C:\book.txt

C:\book.txt 開檔完成, 原檔案大小 103 Bytes

請問是否要繼續新增資料(Y/N):y
請輸入要新增的資料:ASP .NET 4.0基礎必修課

請問是否要繼續新增資料(Y/N):n

資料更新完成...

C:\book.txt 目前檔案大小 127 Bytes

C:\book.txt 資料檔內容如下

Visual C# 2010程式設計經典

Visual Basic 2010程式設計經典

Java 7基礎必修課

Visual C# 2010基礎必修課

ASP .NET 4.0基礎必修課

請按任意鍵繼續 . . .

1. 欲從資料檔依序讀取每一個字元並印在螢幕上，可以使用 `while{...}` 迴圈判斷檔案指標是否已經指到檔案結尾 EOF，若成立則印出 `fgetc()` 函式所讀到的字元；否則即離開迴圈停止列印。程式寫法如下：

```
/* 使用 fgetc()函式取出字元並指定給 ch，再判斷檔案指標是否已經指到 EOF */  
while((ch=fgetc(fptr))!=EOF)  
{  
    printf("%c", ch);/* 若檔案指標還沒指定 EOF，則印出 ch 字元變數的值 */  
}
```

2. 使用 `fputc` 一次只能寫入一個字元到資料檔中，因此我們必須將所輸入字串內的字元一個字一個字的取出，接著再透過 `fputc()` 函式將字元寫入到資料檔中。最後再寫入一個 `'\n'` 換行字元，這樣下一次輸入的資料才會從檔案的下一行開始寫入。

```
char keyin[100];           /* 宣告 keyin 字元陣列 */
gets(keyin);               /* 使用者輸入的字串指定給 keyin 字元陣列 */
for(i=0; i<strlen(keyin); i++) /* 逐一取出 keyin 字元陣列中的每一個字元 */
{
    fputc(keyin[i], fptr); /* 將 keyin 字元陣列中第 i 個字元寫入至資料檔中 */
}
fputc('\n', fptr);         /* 最後寫入 '\n' 換行字元 */
```


程式碼 FileName : fgetc_fputc.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <conio.h>
06 #include <io.h>
07 #define LEN 100
08
09 int main(int argc, char *argv[])
10 {
11     FILE *fptr;           /* 宣告 fptr 檔案結構指標 */
12     char fpath[LEN], keyin[LEN];
13     int fno, fsize, i;
14     char ch;
15     printf("請選擇項目->1.開新檔 2.開舊檔:");
16     if(getche()=='1')
17     {
18         printf("\n 請輸入新檔案路徑:");
19         gets(fpath);
20         fptr=fopen(fpath, "w+");
21     }
```



```
22  else
23  {
24      printf("\n 請輸入欲開啟的檔案路徑：");
25      gets(fpath);
26      fptr=fopen(fpath, "a+");
27  }
28  if(fptr==NULL)
29  {
30      printf("\n 開檔失敗， %s 可能不存在\n", fpath);
31      exit(0);
32  }
33  fno=fileno(fptr);      /* 取得檔案代碼 */
34  fsize=filelength(fno); /* 取得檔案大小 */
35  printf("\n%s 開檔完成，原檔案大小 %d Bytes\n", fpath, fsize);
36  while(1)
37  {
38      printf("\n 請問是否要繼續新增資料 (Y/N)：");
39      if(toupper(getche())=='Y')
40      {
41          printf("\n 請輸入要新增的資料：");
42          gets(keyin);
```

```
43         for(i=0; i<strlen(keyin) ; i++)
44         {
45             fputc(keyin[i], fptr);
46         }
47         fputc('\n', fptr);
48     }
49     else
50     {
51         fclose(fptr);
52         break;
53     }
54 }
55 fptr=fopen(fpath, "r");
56 fno=fileno(fptr);
57 fsize=filelength(fno);
58 printf("\n\n 資料更新完成...");
59 printf("\n%s 目前檔案大小 %d Bytes\n", fpath, fsize);
60 printf("%s 資料檔內容如下\n", fpath);
```



```
60  printf ("%s 資料檔內容如下\n", fpath);
61  while ( (ch=fgetc (fptr)) !=EOF)
62  {
63      printf ("%c", ch);
64  }
65  fclose (fptr);
66  printf ("\n");
67  system ("PAUSE");
68  return 0;
69 }
```

如何以字串讀寫檔案

- 上一節是採字元方式，依序一個字元接一個字元來做檔案讀寫的動作。
- 若您希望每次讀取資料檔採字串方式，就必須透過fgets()輸入函式來讀取；使用fputs()輸出函式來寫入，其語法分別如下：

語法：char *fgets(char *str, int n, FILE *fptr)

功能：在檔案指標 fptr 所指向檔案位置讀取 n 個字元並放入 str 字串。如果讀不到字串時會傳回 NULL。

語法：int fputs(const char *str, FILE *fptr)

功能：將字串 str 寫入檔案指標 fptr 所指向檔案的位置。寫入資料成功時會傳回非零值，寫入失敗會傳回 EOF。




範例：

將 `fgetc_fputc.c` 範例以字元讀寫資料檔的方式，改成使用 `fgets()`及 `fputs()` 函式來讀寫指定的資料檔。本例的執行結果與 `fgetc_fputc.c` 範例相同。

問題分析

1. 欲將字串寫入到指定的資料檔中，可以先使用 `gets()`函式接收使用者所輸入的字串，接著使用 `fputs()`函式，將欲新增的字串寫入到檔案指標所指的資料檔中。最後再寫入一個換行字元，這樣下一次輸入的資料才會從檔案的下一行開始寫入。程式寫法如下：

```
char keyin[100];  
gets(keyin);          /* 將使用者輸入的字串指定給 keyin */  
fputs(keyin, fptr);    /* 將 keyin 字串寫入到 fptr 檔案指標所指向的檔案 */  
fputs("\n", fptr);     /* 寫入換行字元 */
```

- 
2. 如下寫法您使用 `fgets()` 函式從資料檔讀取 100 個字元並指定給 `t` 字串，接著再將 `t` 字串印出，若資料檔讀不到資料時會傳回 `NULL`，此時會離開 `while` 無窮迴圈。

```
char t[100];  
while(fgets(t, 100, fptr)!=NULL)  
{  
    printf("%s", t);  
}
```


程式碼 FileName : fgets_fputs.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <conio.h>
06 #include <io.h>
07 #define LEN 100
08
09 int main(int argc, char *argv[])
10 {
11     FILE *fptr;
12     char fpath[LEN], keyin[LEN], t[LEN];
13     int fno, fsize;
14     printf("請選擇項目->1.開新檔 2.開舊檔:");
15     if(getche()=='1')
16     {
17         printf("\n 請輸入新檔案路徑:");
18         gets(fpath);
19         fptr=fopen(fpath, "w+");    /*讀寫新檔*/
20     }
```



```
21  else
22  {
23      printf("\n 請輸入欲開啟的檔案路徑：");
24      gets(fpath);
25      fptr=fopen(fpath, "a+");      /*讀寫附加檔案模式*/
26  }
27  if(fptr==NULL)
28  {
29      printf("\n 開檔失敗， %s 可能不存在\n", fpath);
30      exit(0);
31  }
32  fno=fileno(fptr);      /* 取得檔案代碼 */
33  fsize=filelength(fno);      /* 取得檔案大小 */
34  printf("\n%s 開檔完成，原檔案大小 %d Bytes\n", fpath, fsize);
35  while(1)
36  {
37      printf("\n 請問是否要繼續新增資料(Y/N)：");
38      if(toupper(getche())=='Y')
39      {
40          printf("\n 請輸入要新增的資料：");
```

```
41         gets(keyin);          /*使用者輸入字串指定給 keyin*/
42         fputs(keyin, fptr); /*keyin 字串寫入 fptr 所指的檔案*/
43         fputs("\n", fptr); /*將換行字元寫入 fptr 所指的檔案*/
44     }
45     else
46     {
47         fclose(fptr);
48         break;
49     }
50 }
51 fptr=fopen(fpath, "r");
52 fno=fileno(fptr);          /*取得檔案代碼*/
53 fsize=filelength(fno);     /*取得檔案大小*/
54 printf("\n 資料更新完成...");
55 printf("\n%s 目前檔案大小 %d Bytes\n", fpath, fsize);
56 printf("%s 資料檔內容如下\n", fpath);
```



```
57  /* 以字串的方式讀取資料檔中的資料 */
58  while(fgets(t, LEN, fptr) != NULL)
59  {
60      printf("%s", t);
61  }
62  fclose(fptr);
63  printf("\n");
64  system("PAUSE");
65  return 0;
66 }
```



如何複製資料檔

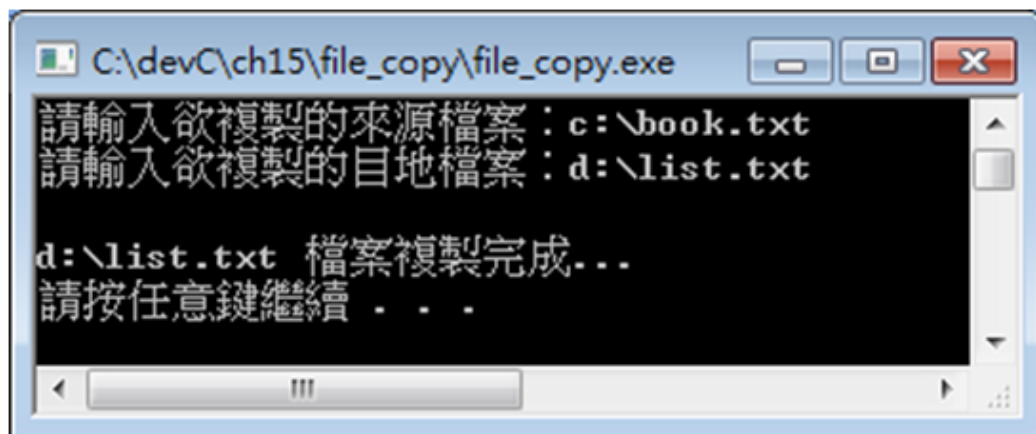
- 複製檔案的技巧就是產生兩個檔案結構指標，一個指標用來讀取來源檔案的資料；另一個指標用來將讀出的資料寫入到目的檔案。
- 兩個檔案指標一起移動待來源指標的檔案指標指到EOF時便結束複製的動作，接著再將兩個資料檔關閉。



範例

試寫一個檔案的複製程式，程式執行先要求您輸入來源和目的地檔的路徑和檔名。若檔案複製成功，會顯示「檔案複製成功…」訊息，否則會顯示「開檔失敗，無法複製...」訊息。

執行結果



```
C:\devC\ch15\file_copy\file_copy.exe
請輸入欲複製的來源檔案：c:\book.txt
請輸入欲複製的目的地檔案：d:\list.txt

d:\list.txt 檔案複製完成...
請按任意鍵繼續 . . .
```

問題分析

1. 先使用 `gets` 函式分別輸入來源和目的的路徑和檔名，存入 `paths` 和 `patht` 字元陣列中。
2. 以讀取模式開啟來源檔案，並同時檢查是否開啟失敗？若開啟失敗提示開檔失敗，並使用 `exit(0)` 結束程式。
3. 以寫入模式開啟目的檔案，並同時檢查是否開啟失敗？若開啟失敗，提示開檔失敗，並使用 `exit(0)` 結束程式。
4. 使用 `fgets()` 函式每次由來源檔案讀取 100 個字元到 `temp` 字元陣列中，接著再使用 `fputs()` 函式存入目的檔案，重覆此動作一直碰到 EOF 檔案結束符號為止。
5. 分別將來源和目的檔案關閉。

程式碼 FileName : file_copy.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <ctype.h>
04 #include <io.h>
05 #define LEN 100
06
07 int main(int argc, char *argv[])
08 {
09     FILE *fptrsource, *fptrtarget;
10     char paths[LEN], patht[LEN];
11     char keyin[LEN], temp[LEN];
12     int fno, fsize;
13     printf("請輸入欲複製的來源檔案：");
14     gets(paths);
15     fptrsource=fopen(paths, "r");
16     printf("請輸入欲複製的目的地檔案：");
17     gets(patht);
18     fptrtarget=fopen(patht, "w");
```

```
19  if(fptrsource==NULL || fptrtarget==NULL)
20  {
21      printf("\n 開檔失敗，無法複製\n");
22      exit(0);
23  }
24  while(fgets(temp, LEN, fptrsource)!=NULL)
25  {
26      fputs(temp, fptrtarget);
27  }
28  fclose(fptrsource);
29  fclose(fptrtarget);
30  printf("\n%s 檔案複製完成...\n", patht, fsize);
31
32  system("PAUSE");
33  return 0;
34 }
```


如何存取格式化文字檔

- fscanf() 函式提供格式化讀取資料檔的功能；至於 fprintf() 函式是以指定的格式將資料寫到指定的資料檔中。
- 上述兩個函式的用法和 scanf() 及 printf() 函式差不多，差別在於後者是存取記憶體內的變數，至於 fscanf() 及 fprintf() 函式則是用來存取磁碟中的資料檔。其語法如下：

語法：int fscanf(FILE *fptr, "格式化字串", [引數串列])

功能：以指定的格式由緩衝區中讀取資料並放置引數串列對應的變數。

說明

1. `fptr`：第一個引數 `fptr` 是指定欲讀取資料檔的檔案結構指標。
2. 格式化字串：第二個引數的用法與第四章 `scanf()` 函式的第一個引數相同，可以使用 `%s, %d %c...` 等來設定欲讀取資料檔中資料的格式。
3. 引數串列：第三個引數的用法與第四章 `scanf()` 函式的第二個引數相同，是用來設定將讀取的資料放至指定的變數位址。
4. 若 `fscanf()` 函式讀取資料成功會傳回所讀取資料的個數，並將資料按照指定格式存入記憶體中的變數、陣列或結構，檔案指標同時會往下移；若讀取資料失敗則會傳回 `EOF`。

4. 若 `fscanf()` 函式讀取資料成功會傳回所讀取資料的個數，並將資料按照指定格式存入記憶體中的變數、陣列或結構，檔案指標同時會往下移；若讀取資料失敗則會傳回 EOF。

[例] 假設 `C:\product.txt` 檔中存有下面一筆資料，該筆資料含有編號、品名、單價三個欄位資料，每個資料之間是使用空白來區隔(△代表空白)。

A01△碁峰果汁△200

欲從 C:\product.txt 資料檔中讀取該筆資料給指定的變數，步驟如下：

- ① 宣告 pid、pname 字元陣列用來存放編號和品名的資料，宣告 price 整數變數用來存放單價：

```
char pid[5], pname[20];
```

```
int price
```

- ② 宣告 fptr 是一個檔案指標，指到所開啟資料檔的起始位址，並將開啟的檔案設為讀取模式：

```
FILE fptr;
```

```
fptr=fopen("C:\\product.txt", "r");
```

- ③ 使用 fscanf()函式以指定的格式讀取 product.txt 的資料，若讀取成功會傳回 3(因為設定的引數串列有三個)，則 pid↵"A01"；pname↵ "碁峰果汁"； price↵200；若讀取失敗會傳回 EOF。

```
fscanf(fptr, "%s△%s△%d", pid, pname, &price);
```

接著介紹 fprintf()函式：

語法：int fprintf(FILE *fptr, “格式化字串”, [引數串列])

功能：以指定的格式將引數串列對應的變數資料寫入至 fptr 指標所指的檔案，適用於循序檔。

說明

1. fptr：第一個引數 fptr 用來指定欲寫入資料檔的檔案結構指標。
2. 格式化字串：第二個引數的用法與第四章 printf()函式的第一個引數相同，可以使用%s, %d, %c...等來設定欲寫入至資料檔中資料的格式。
3. 引數串列：第三個引數的用法與第四章 printf()函式的第二個引數相同，可將指定的引數串列中的變數，逐一寫入指定的資料檔中。

[例] 延續上例 C:\product.txt 檔中已經存有一筆產品記錄，包含編號、品名、單價資料，每一個資料之間是使用空白來區隔(△代表空白)。

A01△碁峰果汁△200

如果希望在該資料檔的最後面再存入一筆資料，其寫法如下：

```
char pid[5]="A02", pname[20]="牛肉";  
int price=200                      /* 宣告 price 整數變數 */  
FILE fptr;                          /* 宣告 fptr 檔案指標 */  
fptr=fopen("C:\\product.txt", "a+"); /*將檔案存取模式為附加資料模式 */  
fprintf(fptr, "%s△%s△%d\n", pid, pname, price); /*寫入資料 */
```

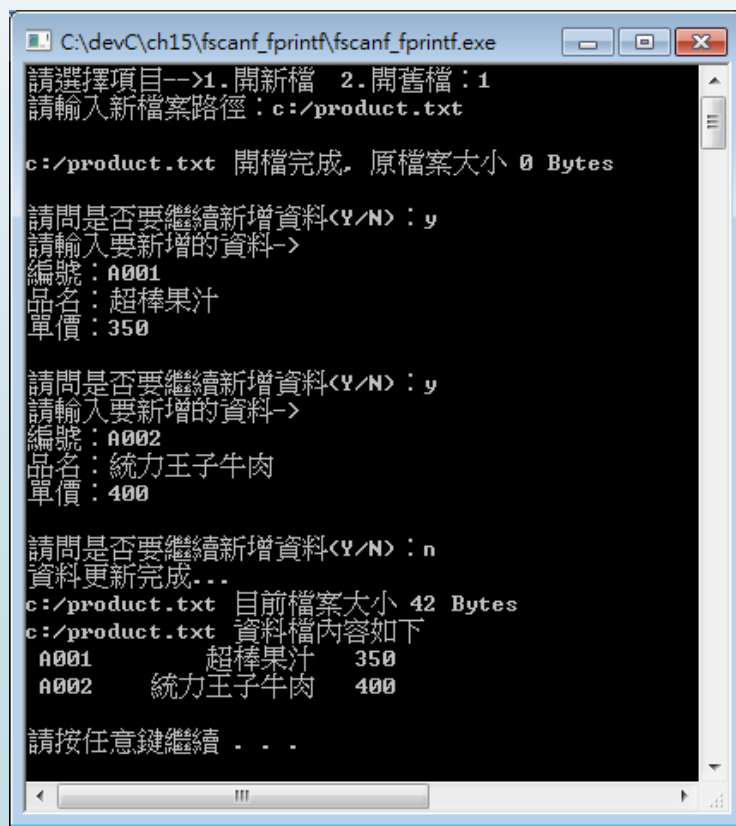
寫入成功後，C:\product.txt 檔案內容如下：

A01△碁峰果汁△200
A02△牛肉△200



範例：

由鍵盤輸入欲新增或開啟的產品資料檔，接著程式會詢問您是否要新增產品資料？若按 'y' 或 'Y' 鍵，則可讓您新增產品記錄的編號、品名、單價資料；若按其他鍵則會印出資料檔內所有的產品記錄。本例使用 `fprintf()` 函式將使用者輸入的編號、品名、單價寫入至指定的資料檔；使用 `fscanf()` 函式由資料檔中讀取所有的產品編號、品名及單價。



```
C:\devC\ch15\fsanf_fprintf\fsanf_fprintf.exe
請選擇項目-->1. 開新檔 2. 開舊檔:1
請輸入新檔案路徑: c:/product.txt

c:/product.txt 開檔完成, 原檔案大小 0 Bytes

請問是否要繼續新增資料(Y/N): y
請輸入要新增的資料->
編號: A001
品名: 超棒果汁
單價: 350

請問是否要繼續新增資料(Y/N): y
請輸入要新增的資料->
編號: A002
品名: 統力王子牛肉
單價: 400

請問是否要繼續新增資料(Y/N): n
資料更新完成...
c:/product.txt 目前檔案大小 42 Bytes
c:/product.txt 資料檔內容如下
A001      超棒果汁      350
A002      統力王子牛肉  400

請按任意鍵繼續 . . .
```


1. 如果要將使用者輸入的編號、品名、單價寫入到資料檔中，首先可以宣告 pid、pname、str_price 字元陣列，宣告 price 整數變數，接著透過 gets() 函式將使用者輸入的編號、品名、單價依序指定給 pid、pname、str_price，接著將 str_price 字串轉成整數並指定給 price，最後再使用 fprintf() 函式以指定的格式將 pid、pname、price 寫入到 fptr 所指向的檔案緩衝區中。程式寫法如下：

```
char pid[5], pname[30], str_price[5];
int price=0;
.....
printf("編號：");
gets(pid);
printf("品名：");
gets(pname);
printf("單價：");
gets(str_price);
price=atoi(str_price);
fprintf(fptr, "%s△%s△%d\n", pid, pname, price);
```


2. 本例使用 `fscanf()` 函式讀取資料檔的資料，我們可以使用 `while` 迴圈並配合 `fscanf()` 函式將讀出的編號、品名、單價等資料指定給 `pid`、`pname` 和 `price`，並判斷檔案指標是否還沒有指到檔案結尾 `EOF`；若檔案指標沒有指到檔案結尾 `EOF` 則會印出目前取得的 `pid`、`pname`、`price`；若檔案指標已經指到檔案結尾 `EOF` 則會離開 `while` 迴圈。其程式寫法如下：

```
char pid[5], pname[30];
int price=0;
.....
/* 判斷檔案指標是否指到檔案結尾 */
while(fscanf(fp, "%s %s %d", pid, pname, &price)!=EOF)
{
    printf("%5s△%15s△%5d\n", pid, pname, price);
}
```

程式碼 FileName : fscanf_fprintf.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <conio.h>
06 #include <io.h>
07 #define LEN 100
08
09 int main(int argc, char *argv[])
10 {
11     FILE *fptr;
12     char fpath[LEN];
13     char pid[5], pname[30], str_price[5];
14     int fno, fsize, price=0;
15     printf("請選擇項目-->1.開新檔 2.開舊檔:");
16     if(getche()=='1')
17     {
18         printf("\n 請輸入新檔案路徑:");
19         gets(fpath);
20         fptr=fopen(fpath, "w+");    /*以建立新檔的模式開檔*/
21     }
```

```
22  else
23  {
24      printf("\n 請輸入欲開啟的檔案路徑：");
25      gets(fpath);
26      fptr=fopen(fpath, "a+");    /*以附加資料的模式開檔*/
27  }
28  if(fptr==NULL)
29  {
30      printf("\n 開檔失敗， %s 可能不存在\n", fpath);
31      exit(0);
32  }
33  fno=fileno(fptr);
34  fsize=filelength(fno);
35  printf("\n%s 開檔完成，原檔案大小 %d Bytes\n", fpath, fsize);
36  while(1)
37  {
38      printf("\n 請問是否要繼續新增資料 (Y/N)：");
39      if(toupper(getche())=='Y')    /* 按'y'或'Y'表示要新增資料 */
40      {
41          printf("\n 請輸入要新增的資料->\n");
42          printf("編號：");
```

```
43      gets(pid);                /*輸入編號*/
44      printf("品名：");
45      gets(pname);              /*輸入品名*/
46      printf("單價：");
47      gets(str_price);          /*輸入單價指定給 str_price 字串 */
48      price=atoi(str_price); /*將 str_price 字串轉成整數並指定給 price */
49      /* 將 pid, pname, price 寫入到 fptr 所指的檔案緩衝區中 */
50      fprintf(fptr, "%s %s %d\n", pid, pname, price);
51  }
52  else
53  {
54      fclose(fptr); /*關閉檔案*/
55      break;
56  }
57 }
```

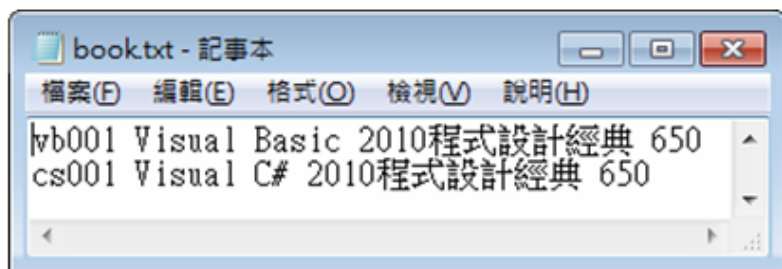
```
58  fptr=fopen(fpath, "r");
59  fno=fileno(fptr);           /* 取得檔案代碼 */
60  fsize=filelength(fno);      /* 取得檔案大小 */
61  printf("\n 資料更新完成...");
62  printf("\n%s 目前檔案大小 %d Bytes\n", fpath, fsize);
63  printf("%s 資料檔內容如下\n", fpath);
64  /* 若檔案指標 fptr 還沒有指到檔案結尾，則印出資料檔的資料 */
65  while(fscanf(fptr, "%s %s %d", pid, pname, &price)!=EOF)
66  {
67      /* 將讀取的資料印出 */
68      printf("%5s %15s %5d\n", pid, pname, price);
69  }
70  fclose(fptr);              /*關閉檔案*/
71  printf("\n");
72
73  system("PAUSE");
74  return 0;
75}
```

如果我們在本範例輸入下面兩筆含有空白的資料：

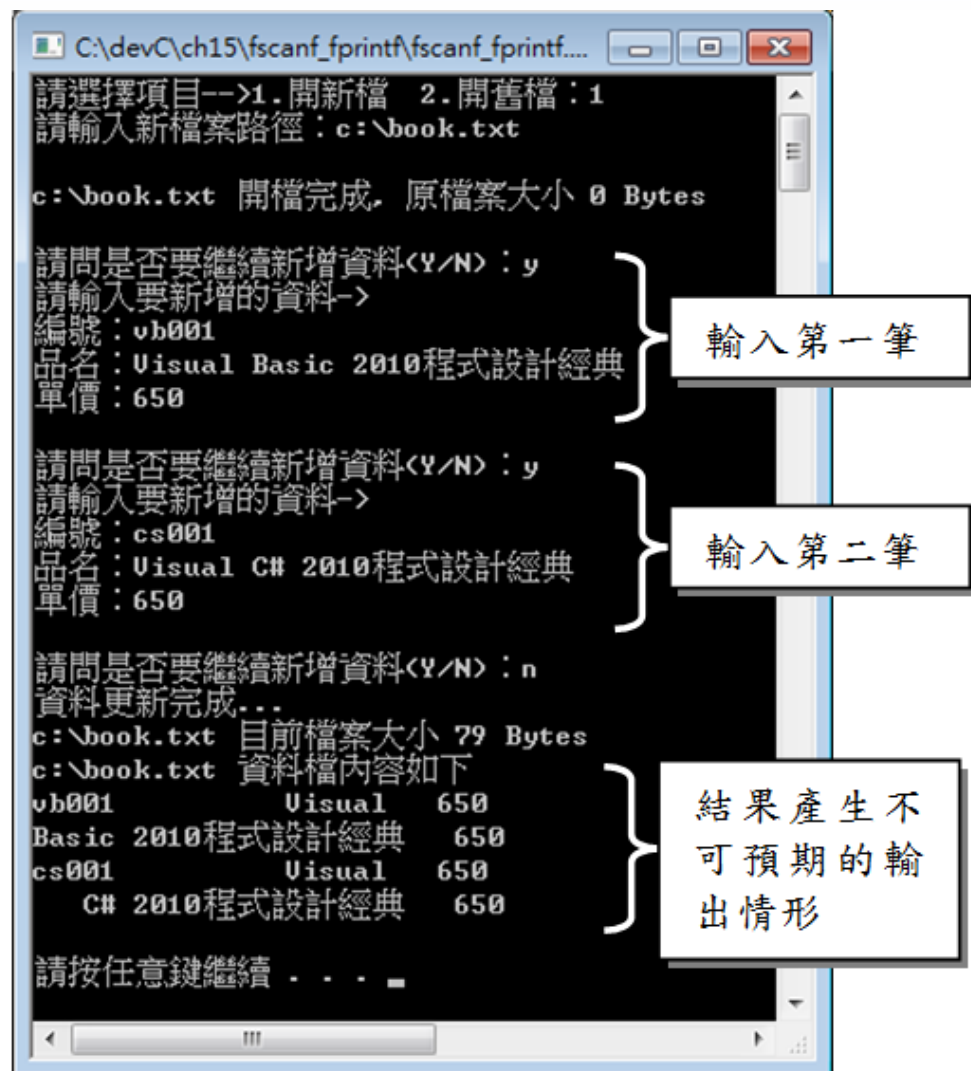
編號：cs001, 品名：Visual△C#△2010 程式設計經典, 單價：580

編號：vb001, 品名：Visual△Basic△2010 程式設計經典, 單價：600

如下圖，雖然這兩筆資料可以正確的寫入資料檔中，結果在讀取資料時會發生不可預期的輸出情形。原因是當 `fscanf()` 函式讀取資料時會以空白字元當做資料與資料之間的區隔，所以才會發生這樣的狀況。關於輸出含有空白資料的問題，就必須使用 `fread()` 及 `fwrite()` 函式來製作結構化輸出入隨機檔方可解決。請參考下節隨機檔的存取。



正確的寫入資料



隨機檔

隨機檔的存取

- `fscanf()`及`fprintf()`函式來存取檔案的資料，若遇到有空白時會發生不可預期的輸出結果，因此可使用`fwrite()`及`fread()`函式以結構I/O方式來解決這個問題。
- 隨機檔是每筆資料以結構的方式存入磁碟中。
- `fwrite()`可將整塊資料即結構(即一筆或多筆資料)寫入到檔案指標所指的檔案；而`fread()`可將目前檔案指標的位置上讀取一塊資料並放入指定的結構中。
- 因為資料存取都是採用結構的形式存在，因此每一筆資料的大小都是固定，讀取資料時只要告知要讀取第幾筆資料，即可利用公式算出該筆資料的位置，透過`fseek()`函式即可將檔案指標快速移到該資料位置，因此存取資料的速度比循序檔快，若該資料檔經常要做資料的新增、刪除、修改或查詢的作業時是非常方便。

語法：size_t fwrite(const *ptr, size_t size, size_t n, FILE *fptr)

功能：將 ptr 結構指標所指的資料區塊內一次讀取 n 筆，每筆資料大小為 size Bytes 寫入至 fptr 檔案指標所指的資料檔內。若寫入成功，會傳回所寫入的結構筆數且 fptr 檔案指標會往後移動 n 筆結構的長度(即 $n \times \text{size}$ 大小)。此函式適用於隨機檔。將結構 ptr 的資料一次寫入 n 筆至 fptr 指標所指的檔案內，寫入結構資料時必須指定結構 ptr 的大小 size。寫入成功時會傳回所寫入的結構筆數且檔案指標會往後移動 n 筆結構的長度(即 $n \times \text{size}$ 大小)。此函式適用於隨機檔。

語法：`size_t fread(void *ptr, size_t size, size_t n, FILE *fptr)`

功能：由 `fptr` 檔案指標所指的檔案內讀取 `n` 筆記錄至 `ptr` 結構指標所指的結構內，讀取時必須指定結構的大小 `size`。若讀取成功，傳回所讀取的結構筆數且檔案指標會往後移動 `n` 筆結構的長度(即 `n x size` 的大小)，若讀不到資料，則會傳回 0，此函式適用於隨機檔。

- `fwrite()`及`fread()`函式的第一個引數指標結構`ptr`必須設定該結構的位址，因此結構變數前面必須加上`&`。



範例：

由鍵盤輸入欲新增或開啟的檔案，接著程式會詢問您是否要新增書籍？若按 'y' 或 'Y' 鍵，則可讓您新增書籍的編號、品名、單價、庫存；按其他鍵則會印出資料檔內所有的記錄以及共有多少筆記錄。

```
C:\devC\ch15\freed_fwrite\freed_fwrite.exe
請選擇?>1.開新檔 2.開舊檔:1
請輸入新檔案路徑:C:\book.txt

請問是否要繼續新增資料(Y/N):y
請輸入要新增的資料->
書號:vb001
書名:Visual Basic 2010程式設計經典
單價:650
庫存:1000

請問是否要繼續新增資料(Y/N):y
請輸入要新增的資料->
書號:cs001
書名:Visual C# 2010程式設計經典
單價:650
庫存:1800

請問是否要繼續新增資料(Y/N):n
資料更新完成...
C:\book.txt 目前檔案大小 96 Bytes
C:\book.txt 資料檔內容如下

記錄總筆數:2
vb001 Visual Basic 2010程式設計經典 650 1000
cs001 Visual C# 2010程式設計經典 650 1800

請按任意鍵繼續 . . .
```

1. 定義 book 結構：

本例使用 `fwrite()` 和 `fread()` 函式將一筆書籍結構資料寫入到指定的資料檔中，所以請定義下面 `book` 結構，結構成員有 `bookid` 表示書號、`bookname` 表示書名、`price` 表示單價、`qty` 表示庫存、`flag` 用來表示是否被刪除，若 `flag` 為 "*" 表示被刪除。flag 成員在修改、刪除的 `update_del.c` 範例才會用得到。

```
typedef struct
{
    char bookid[6];
    char bookname[30];
    int price;
    int qty;
    char flag[3];
}book;
```

2. 宣告 book 結構變數 mybook，接著再將使用者所輸入的書號、書名、單價、庫存依序指定給 mybook.bookid、mybook.bookname、mybook.price、mybook.qty。
3. 使用 fwrite()函式將一筆 mybook 結構變數寫入到檔案中，其寫法如下：fwrite()函式第一個引數必須指定要寫入結構變數的位址，第二個引數指定要寫入結構的長度，第三個引數指定要寫入的筆數，第四個引數指定要寫入到那個檔案。

```
fwrite(&mybook, sizeof(mybook), 1, fptr );
```

↑
結構變數

↑
記錄長度

↑
筆數

↑
檔案指標

4. 使用 `fread()` 函式讀取多筆資料時可以再配合 `while(){...}` 迴圈判斷檔案指標是否讀到資料，若有讀到資料可以進入 `while` 迴圈內顯示該筆資料的內容，否則即離開迴圈。其寫法如下：

```
while(fread(&mybook, sizeof(mybook), 1, fptr)!=NULL)
{
    printf("%6s %30s %5d %5d\n",
           mybook.bookid, mybook.bookname, mybook.price, mybook.qty);
}
```

5. 檔案大小除以結構大小即會求出記錄筆數，其寫法如下：


```
rectot = filelength(fileno(fp)) / sizeof(book);
```

↑ ↑ ↑


記錄筆數 檔案大小 結構大小

程式碼 FileName : fread_fwrite.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <conio.h>
06 #include <io.h>
07 #define LEN 100
08 typedef struct          /*定義書籍結構*/
09 {
10     char bookid[6];      /*書號*/
11     char bookname[30];   /*書名*/
12     int price;           /*單價*/
13     int qty;             /*單價*/
14     char flag[3];        /*刪除旗標*/
15 }book;
16
17 int main(int argc, char *argv[])
18 {
19     FILE *fptr;
20     char fpath[LEN], t_price[10], t_qty[10];
21     int fno, fsize, rectot;
```



```
22  book mybook;
23  printf("請選擇功能->1.開新檔  2.開舊檔：");
24  if(getche()=='1')
25  {
26      printf("\n 請輸入新檔案路徑：");
27      gets(fpath);
28      fptr=fopen(fpath, "w+");
29  }
30  else
31  {
32      printf("\n 請輸入欲開啟的檔案路徑：");
33      gets(fpath);
34      fptr=fopen(fpath, "a+");
35  }
36  if(fptr==NULL)
37  {
38      printf("\n 開檔失敗， %s 可能不存在\n", fpath);
39      exit(0);
40  }
41  while(1)
42  {
43      printf("\n 請問是否要繼續新增資料(Y/N)：");
```





```
44     if (toupper(getche()) == 'Y')
45     {
46         printf("\n 請輸入要新增的資料->\n");
47         printf("書號：");
48         gets(mybook.bookid);
49         printf("書名：");
50         gets(mybook.bookname );
51         printf("單價：");
52         gets(t_price);
53         mybook.price = atoi(t_price);
54         printf("庫存：");
55         gets(t_qty);
56         mybook.qty = atoi(t_qty);
57         /*將結構寫入檔案*/
58         fwrite(&mybook, sizeof(mybook), 1, fptr);
59     }
60     else
61     {
62         fclose(fptr);
63         break;
64     }
65 }
```

```
66  fptr=fopen(fpath, "r");
67  fno=fileno(fptr);
68  fsize=filelength(fno);
69  /*取得記錄總筆數*/
70  rectot=filelength(fileno(fptr))/sizeof(book);
71  printf("\n 資料更新完成...");
72  printf("\n%s 目前檔案大小 %d Bytes\n", fpath, fsize);
73  printf("%s 資料檔內容如下\n", fpath);
74  printf("\n 記錄總筆數：%d\n", rectot);
75  /* 判斷是否還有資料*/
76  while(fread(&mybook, sizeof(mybook), 1, fptr)!=NULL)
77  {    /*取出一筆記錄，檔案指標再往後移動一筆記錄的長度*/
78      printf("%6s %30s %5d %5d\n",
79          mybook.bookid, mybook.bookname, mybook.price, mybook.qty);
80  }
81  fclose(fptr);
82  printf("\n");
83
84  system("PAUSE");
85  return 0;
86 }
```

隨機檔的修改與刪除

- 隨機檔內每一筆資料的長度都是相同的，因此您可以透過 `fseek()` 函式指定檔案指標要往後移幾筆資料位址，接著再使用 `fread()` 函式讀出指定的資料；或是使用 `fwrite()` 函式更新指定的資料。
- 若要重新查詢資料就必須使用 `rewind()` 函式將檔案指標移到檔案的最開頭，這樣檔案指標才會從檔案的最開頭開始往下移動。



語法：void rewind(FILE *fptr)

功能：將檔案指標移到檔案的最開頭。

語法：int fseek(FILE *fptr, long offset, int whence)

功能：檔案指標由 whence 位址移到 offset Bytes 的位址。

[例 1] 檔案指標讀取第 n 筆的寫法：

```
rewind(fptr); /*檔案指標移到最開頭*/  
fseek(fptr, sizeof(mybook)*n, 0); /*檔案指標由 0 位址往後移 sizeof(mybook)*n 位址 */  
/*即表示檔案指標後移 n 筆資料的位址*/  
fread(&mybook, sizeof(mybook), 1, fptr); /*讀出檔案指標目前所指的資料給 mybook*/
```

[例 2] 檔案指標更新第 n 筆的寫法：

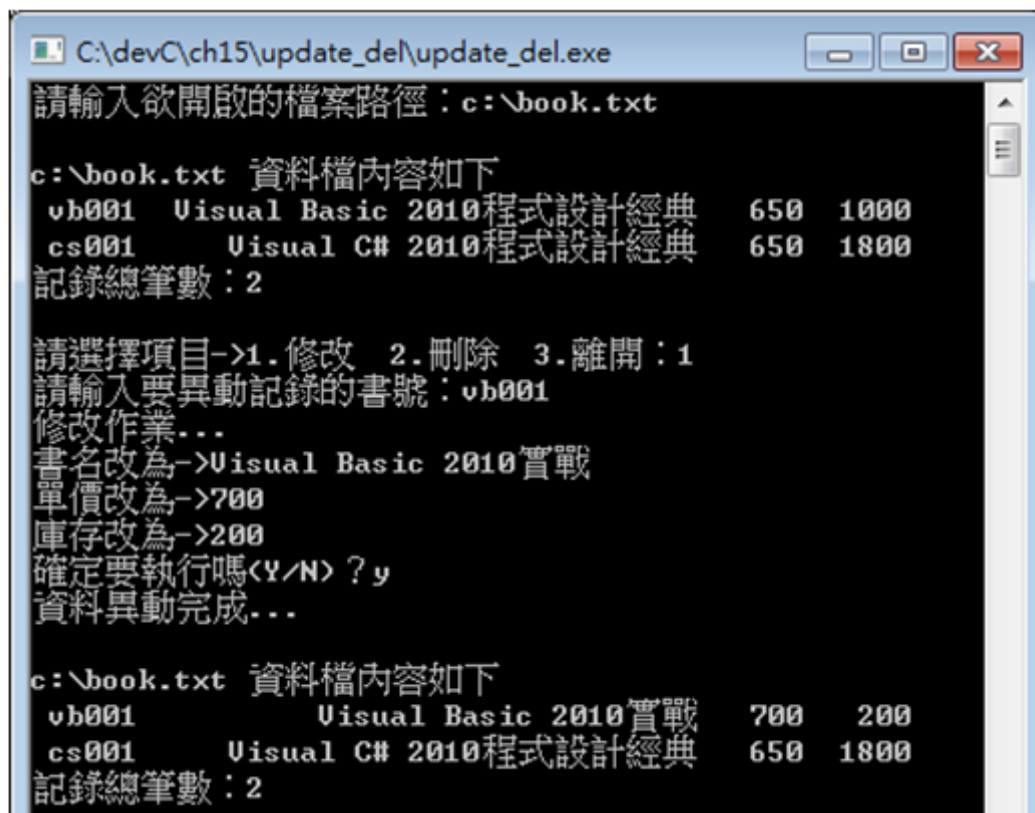
```
rewind(fptr); /*檔案指標移到最開頭*/  
fseek(fptr, sizeof(mybook)*n, 0); /*檔案指標由 0 位址往後移 sizeof(mybook)*n 位址  
/*即表示檔案指標後移 n 筆資料的位址*/  
fwrite(&mybook, sizeof(mybook), 1, fptr); /*將 mybook 寫入檔案指標目前所指的資料*/
```



範例：

使用 `rewind()` 及 `fseek()` 函式來製作可以刪除及修改隨機檔內指定的資料。如下圖，您可以開啟指定的檔案，並選擇功能清單「1.修改 2.刪除 3.離開」，接著依據要異動記錄的書號，來指定要修改的書名、單價、庫存。修改完後會再一次詢問您是否(Y/N)確定要執行異動資料。

執行結果



```
C:\devC\ch15\update_del\update_del.exe
請輸入欲開啟的檔案路徑：c:\book.txt

c:\book.txt 資料檔內容如下
vb001  Visual Basic 2010程式設計經典  650  1000
cs001  Visual C# 2010程式設計經典    650  1800
記錄總筆數：2

請選擇項目->1.修改 2.刪除 3.離開：1
請輸入要異動記錄的書號：vb001
修改作業...
書名改為->Visual Basic 2010實戰
單價改為->700
庫存改為->200
確定要執行嗎(Y/N)？y
資料異動完成...

c:\book.txt 資料檔內容如下
vb001  Visual Basic 2010實戰  700  200
cs001  Visual C# 2010程式設計經典  650  1800
記錄總筆數：2
```


1. 定義 book 結構：

book 結構中的 flag 成員用來表示是否被刪除，若 flag 為 "*" 表示該筆資料被刪除。(參考第 9~16 行)

```
typedef struct
{
    char bookid[6];
    char bookname[30];
    int price;
    int qty;
    char flag[3];
}book;
```

2. 這個程式我們撰寫了 showfiledata()函式，這個函式會印出指定隨機檔的所有資料。(參考第 111~129 行)

程式碼 FileName : update_del.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <conio.h>
06 #include <io.h>
07 #define LEN 100
08
09 typedef struct          /*book 結構*/
10 {
11     char bookid[6];      /*書號*/
12     char bookname[30];   /*書名*/
13     int price;           /*單價*/
14     int qty;             /*庫存*/
15     char flag[3];        /*刪除旗標*/
16 }book;
17
18 void showfiledata(char[]);
19
20 int main(int argc, char *argv[])
21 {
```



```
22  FILE *fptr;
23  char fpath[LEN];
24  char search_bookid[6], t_price[10], t_qty[10];
25  book mybook;
26  int recno=0;      /*資料記錄編號*/
27  char ch;
28  printf("請輸入欲開啟的檔案路徑：");
29  gets(fpath);
30  fptr=fopen(fpath, "r+");
31  if(fptr==NULL)
32  {
33      printf("\n 開檔失敗， %s 可能不存在\n", fpath);
34      exit(0);
35  }
36  showfiledata(fpath);
37  printf("\n 請選擇項目->1.修改  2.刪除  3.離開：");
38  ch=getche();
39  if(ch=='3' && ch !='1' && ch != '2')      /*選擇功能清單*/
40  {
41      printf("\n\n 結束程式...\n");
42      exit(0);
43  }
```

```
44  printf("\n 請輸入要異動記錄的書號：");
45  gets(search_bookid);          /*輸入要查詢的書號*/
46  rewind(fptr);                 /*檔案指標移到檔案最開頭*/
47  while(1)
48  {
49      /* 檔案指標移到最後結尾會傳回 NULL，表示找不到資料 */
50      if(fread(&mybook, sizeof(mybook), 1, fptr)==NULL)
51      {
52          printf("\n 沒有書號 %s 這筆記錄...\n", search_bookid);
53          exit(0);
54      }
55      else
56          /*找到資料*/
57      {
58          /*判斷資料是否被刪除*/
59          if(strcmp(mybook.bookid, search_bookid)==0 &&
60              strcmp(mybook.flag, "")!=0)
61          {
62              if(ch=='1')
63              {          /*輸入要修改的資料*/
64                  printf("修改作業...\n");
65                  printf("書名改為->");
```

```
66         gets(mybook.bookname );
67         printf("單價改為->");
68         gets(t_price);
69         mybook.price = atoi(t_price);
70         printf("庫存改為->");
71         gets(t_qty);
72         mybook.qty = atoi(t_qty);
73     }
74     else if(ch=='2')
75     {
76         /*將 mybook.flag 設為 "*", 表示設定刪除旗標*/
77         printf("刪除作業...\n");
78         strcpy(mybook.flag , "*");
79     }
80     printf("確定要執行嗎 (Y/N) ? ");
81     /*詢問是否執行異動資料的作業*/
82     if(toupper(getche())=='Y')
83     {
84         /*檔案指標移到第 recno 筆資料位址*/
85         fseek(fp, sizeof(mybook)*recno, 0);
86         /*將 mybook 寫入檔案指標目前所指的資料位址*/
87         fwrite(&mybook, sizeof(mybook), 1, fp);
```

```
88             fclose(fptr);
89             break;
90         }
91         else
92         {
93             printf("\n\n放棄執行...\n");
94             exit(0);
95         }
96     }
97     else
98     {
99         /*recno 記錄編號加 1，表示移到下一筆資料*/
100         recno++;
101     }
102 }
103 }
104 printf("\n 資料異動完成...\n");
105 showfiledata(fpath);
106 printf("\n");
107 system("PAUSE");
108 return 0;
109 }
```

```
111 void showfiledata(char vfilepath[])
112 {
113     int rectot=0;
114     book vbook;
115     FILE *vfptr;
116     vfptr=fopen(vfilepath, "r");
117     printf("\n%s 資料檔內容如下\n", vfilepath);
118     while(fread(&vbook, sizeof(vbook), 1, vfptr) != NULL)
119     {
120         if(strcmp(vbook.flag, "*") != 0)
121         {
122             printf("%6s %30s %5d %5d\n",
123                 vbook.bookid, vbook.bookname, vbook.price, vbook.qty);
124             rectot++;
125         }
126     }
127     printf("記錄總筆數： %d\n", rectot);
128     fclose(vfptr);
129 }
```



系統輸出入函式

系統輸出入函式簡介

- 系統輸出入函式沒有提供緩衝區，而且此類函式是以檔案代碼來代替檔案指標，且不提供格式化的處理功能。
- 下表為常用的系統輸出入函式，使用這些函式時記得要含入 `io.h`。

函式	功能說明
open	<p>語法：int open(const char *filename, int oflag[, int mode])</p> <p>功能：使用 oflag 開啟模式來開啟指定的 filename 檔案。若開啟檔案成功會傳回檔案代碼(handle_no)；開啟檔案失敗則會傳回-1。</p>
close	<p>語法：int close(int handle_no)</p> <p>功能：將檔案代碼 handle_no 所指的檔案關閉。若傳回 0 表示關檔成功；傳回-1 表示關檔失敗。</p>
creat	<p>語法：int creat(const char *filename, int mode)</p> <p>功能：使用 mode 建檔模式來建立指定的 filename 檔案。若建檔成功會傳回檔案代碼(handle_no)；建檔失敗則會傳回-1。</p>
write	<p>語法：int fwrite(int handle_no, void *buffer, unsigned count)</p> <p>功能：由 buffer 陣列的 count 個資料寫入到 handle_no 所代表的檔案。寫入成功檔案指標會往後移 count 並傳回寫入資料的 Bytes 數；寫入失敗會傳回-1。</p>

read	<p>語法：int read(int handle_no, void *buffer, unsigned count)</p> <p>功能：由檔案代碼 handle_no 所代表的檔案中讀取 count 個資料並放入 buffer 陣列。讀取成功會傳回讀出資料的 Bytes 數；讀取失敗會傳回-1。</p>
tell	<p>語法：long tell(int handle_no)</p> <p>功能：傳回檔案代碼 handle_no 所代表檔案的目前檔案指標所指的位置。</p>
lseek	<p>語法：int lseek(int handle_no, long offset, int whence)</p> <p>功能：handle_no 檔案代碼所代表的檔案指標由 whence 移到 offset Bytes。功能與 fseek 類似。</p>
eof	<p>語法：int eof(int handle_no)</p> <p>功能：判斷檔案代碼代表的資料檔的檔案指標是否已經指到檔案結尾(EOF)。若傳回 0 表示檔案尚未結束，傳回 1 表示檔案已經結束，傳回-1 表示有錯誤發生。</p>

開啟與關閉檔案

一. `open()` 函式

- 系統輸出入函式的`open()`函式可開啟檔案並傳回對應的檔案代碼。其語法如下：
- `int open(const char *filename, int oflag[, int mode])`
- 第一個引數`filename`用來指定欲開啟的檔案。
- 第二個引數`oflag`用來設定檔案的開啟模式。
- 第三個引數用來設定檔案的存取模式，第三個引數可省略。

1. oflag 設定

oflag 用來設定檔案的開啟模式，可設定如下常數，若要使用一種以上的 oflag，可以使用「|」符號來區隔，這些常數定義於 `fcntl.h` 標頭檔內。

oflag	功能說明
O_CREAT	檔案不存在，會開新檔，且開檔模式為寫入模式。
O_APPEND	開檔模式是將資料寫入檔案結尾。
O_RDONLY	開檔模式為唯讀模式。設定 O_RDONLY 時，不可以再設定 O_WRONLY、O_RDWR。
O_WRONLY	開啟一個已經存在的檔案，開檔模式為寫入模式。設定 O_WRONLY 時，不可以再設定 O_RDONLY、O_RDWR。
O_RDWR	開檔模式為讀寫模式。
O_TEXT	以文字檔模式開檔。
O_BINARY	以二進檔模式開檔。

2. mode 設定

mode 用來設定檔案的讀寫模式，這些常數定義於 `sys/stat.h` 標頭檔內，可設定常數如下：

mode	功能說明
S_IWRITE	只允許寫入模式。
S_IREAD	只允許讀取模式。
S_IWRITE S_IREAD	允許讀寫模式。

```
int fno;           /*fno 用來存放檔案代表*/  
/*使用 open()函式開啟 C:\sample.txt 檔，開檔成功會傳回檔案代碼，開檔失敗會傳回-1*/  
fno=open("C:\\sample.txt", O_WRONLY|O_CREAT, S_IREAD|S_IWRITE);  
if(fno==-1)       /*若 fno 為-1 表示開檔失敗*/  
{  
    /*檔案開啟失敗*/  
    exit(0);  
}  
.....
```

二. close() 函式

- 若開啟的檔案不想進行存取資料時，可使用 close(fno) 函式關閉該檔案代碼所對應的檔案，若關檔成功會傳回0，關檔失敗會傳回-1。其寫法如下：

```
if (close(fno)==0)
{
    /*關檔成功*/
}
else
{
    /*關檔失敗*/
}
```

檔案的存取

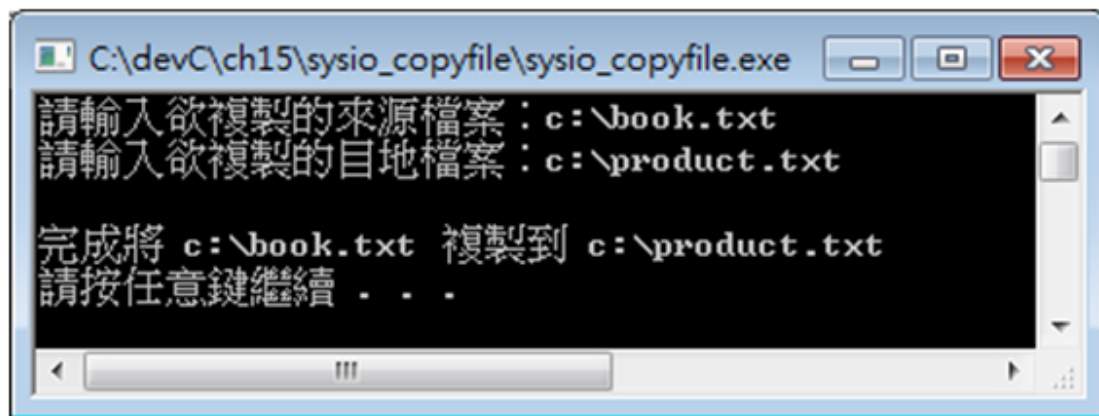
- ➡ 系統輸出入函式沒有提供緩衝區，因此使用read()和write()函式來讀取資料，必須宣告一個陣列用來當做讀寫資料的緩衝區。其寫法如下：

```
int buff[512];    /* 宣告 buff 陣列用來當做緩衝區 */  
                  /* 讀出 fno 所對應檔案中的 512 Bytes 資料並放入 buff 陣列中*/  
                  /* 讀取成功會傳回讀出資料的 Bytes 數，讀取失敗會傳回-1*/  
readbytes=read(fno, buff, 512);  
write(fno, buff, 512); /* 將 buff 陣列中 512Bytes 寫入到 fno 所對應的檔案中*/
```

範例：

複製檔案，程式執行時會要求您輸入來源以及目的檔案路徑和檔名。若複製檔案成功會印出「完成將 XX 複製到 XX…」訊息，否則會印出「複製失敗…」訊息。本例請使用 `open()`、`close()`、`eof()`、`read()`、`write()` 函式。

執行結果



```
C:\devC\ch15\sysio_copyfile\sysio_copyfile.exe
請輸入欲複製的來源檔案：c:\book.txt
請輸入欲複製的目的地檔案：c:\product.txt

完成將 c:\book.txt 複製到 c:\product.txt
請按任意鍵繼續 . . .
```

程式碼 FileName : sysio_copyfile.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <fcntl.h>
04 #include <io.h>
05 #include <sys/stat.h>
06 #define LEN 512
07
08 int main(int argc, char *argv[])
09 {
10     int fno1, fno2, readbytes;
11     int buff[LEN];
12     char f1path[LEN], f2path[LEN];
13     printf("請輸入欲複製的來源檔案：");
14     gets(f1path);
15     printf("請輸入欲複製的目的地檔案：");
16     gets(f2path);
17     fno1=open(f1path, O_RDONLY);          /*開啟檔案為唯讀模式*/
18     /*開檔模式為建新檔及附加寫入模式，存取模式為允許讀寫*/
19     fno2=open(f2path, O_CREAT|O_WRONLY, S_IWRITE|S_IREAD);
```



```
20  if(fno1==-1 || fno2==-1)
21  {
22      printf("\n 複製失敗...\n");
23      exit(0);
24  }
25  /*判斷檔案指標是否讀到 EOF 檔案結尾*/
26  while(eof(fno1)==0)
27  {
28      /*讀出 fno1 對應檔案的資料並放入 buff 陣列*/
29      readbytes=read(fno1, buff, LEN);
30      /*將 buff 陣列寫入 fno2 對應的檔案*/
31      write(fno2, buff, readbytes);
32  }
33  close(fno1);      /*關閉 fno1 檔案代碼對應的檔案*/
34  close(fno2);      /*關閉 fno2 檔案代碼對應的檔案*/
35  printf("\n 完成將 %s 複製到 %s\n", f1path, f2path);
36
37  system("PAUSE");
38  return 0;
39 }
```



範例：

使用系統輸出入函式製作與上一節 fread_fwrite.c 程式相同的範例。

```
C:\devC\ch15\system_io\system_io.exe
請選擇并?>1. 開新檔 2. 開舊檔: 1
請輸入新檔案路徑: c:book.txt


請問是否要繼續新增資料(Y/N): y
請輸入要新增的資料->
書號: vb001
書名: Visual Basic 2010程式設計經典
單價: 650
庫存: 1000

請問是否要繼續新增資料(Y/N): n
資料更新完成...
vb001 Visual Basic 2010程式設計經典 650 1000
記錄總筆數: 1


請按任意鍵繼續 . . .
```

程式碼 FileName : system_io.c

```
01 #include <stdio.h>
02 #include <stdlib.h>
03 #include <string.h>
04 #include <ctype.h>
05 #include <io.h>
06 #include <fcntl.h>
07 #include <sys/stat.h>
08 #include <conio.h>
09 #define LEN 100
10
11 typedef struct
12 {
13     char bookid[6];
14     char bookname[30];
15     int price;
16     int qty;
17     char flag[3];
18 }book;
19
20 int main(int argc, char *argv[])
21 {
22     int fno1, rectot=0;
```



```
23 char fpath[LEN], t_price[10], t_qty[10];
24 book mybook;
25 printf("請選擇功能->1.開新檔 2.開舊檔:");
26 if(getche()=='1')
27 {
28     printf("\n 請輸入新檔案路徑:");
29     gets(fpath);
30     fno1=open(fpath, O_WRONLY|O_CREAT, S_IRREAD|S_IWRITE);
31 }
32 else
33 {
34     printf("\n 請輸入欲開啟的檔案路徑:");
35     gets(fpath);
36     fno1=open(fpath, O_WRONLY|O_APPEND, S_IRREAD|S_IWRITE);
37 }
38 if(fno1==-1)
39 {
40     printf("\n 開檔失敗, %s 可能不存在\n", fpath);
41     exit(0);
42 }
```



```
44  while(1)
45  {
46      printf("\n 請問是否要繼續新增資料 (Y/N) : ");
47      if(toupper(getche())=='Y')
48      {
49          printf("\n 請輸入要新增的資料->\n");
50          printf("書號:");
51          gets(mybook.bookid);
52          printf("書名:");
53          gets(mybook.bookname );
54          printf("單價:");
55          gets(t_price);
56          mybook.price = atoi(t_price);
57          printf("庫存:");
58          gets(t_qty);
59          mybook.qty = atoi(t_qty);
60          write(fno1, &mybook, sizeof(mybook));
61      }
62      else
63      {
64          close(fno1);
65          printf("\n 資料更新完成...\n");
66          break;
67      }
68  }
```

```
69  fnol=open(fpath, O_RDONLY);
70  if(fnol==-1)
71  {
72      printf("\n 開檔失敗, %s 可能不存在\n", fpath);
73      exit(0);
74  }
75  while(!eof(fnol))
76  {
77      read(fnol, &mybook, sizeof(mybook));
78      printf("%6s %30s %5d %5d\n",
79          mybook.bookid, mybook.bookname,
80          mybook.price, mybook.qty);
81      rectot++;
82  }
83  close(fnol);
84  printf("記錄總筆數: %d\n\n", rectot);
85
86  system("PAUSE");
87  return 0;
88 }
```