



程式設計

第8章 檔案處理 File Processing

蘇維宗(Wei-Tsung Su)
suwt@au.edu.tw
564D





目標

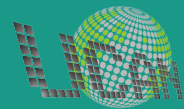
檔案與資料流

處理文字檔案(text file)

處理二進制檔案(binary file)



檔案與資料流



資料流(stream)

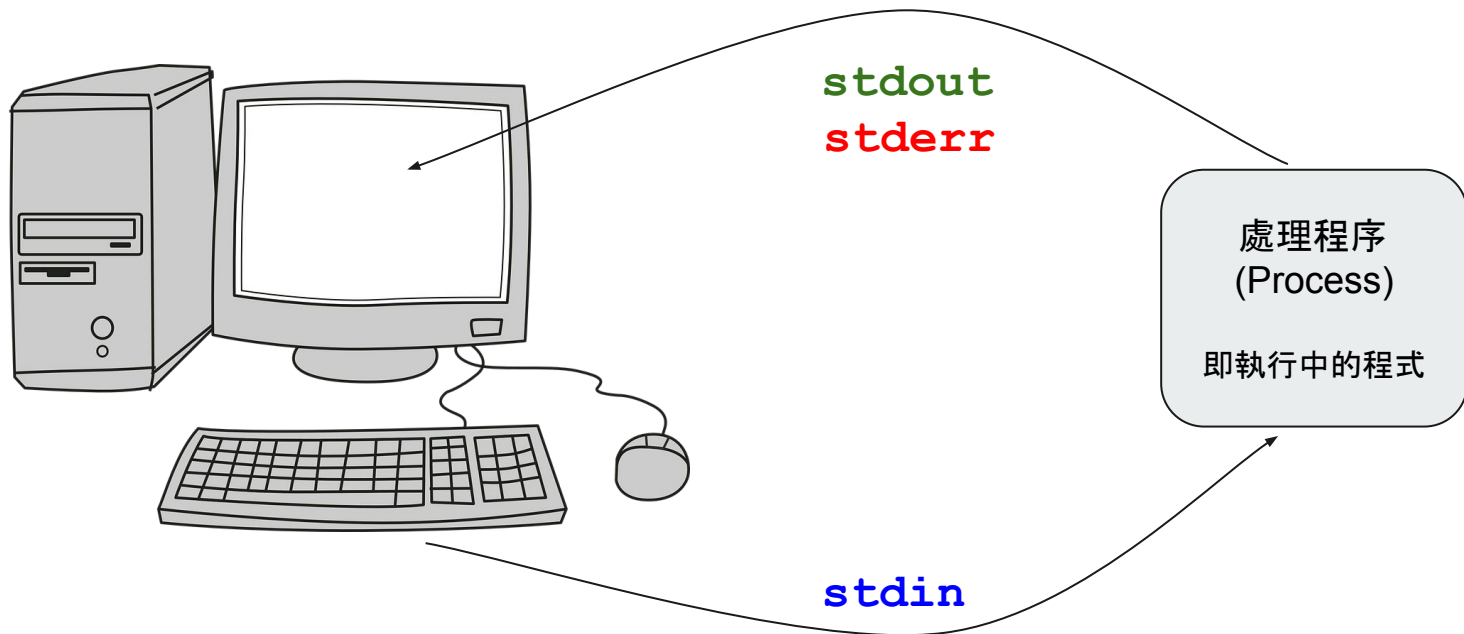
在C語言中，資料流是讓程式與檔案(或如鍵盤、螢幕等輸出入介面)交換資料的通訊管道。

在程式執行時，下面三個標準資料流也會被開啟

1. 標準輸入(**Standard Input**, 簡寫為**stdin**) // 預設為鍵盤輸入
2. 標準輸出(**Standard Output**, 簡寫為**stdout**) // 預設為螢幕
3. 標準錯誤(**Standard Error**, 簡寫為**stderr**) // 預設為螢幕

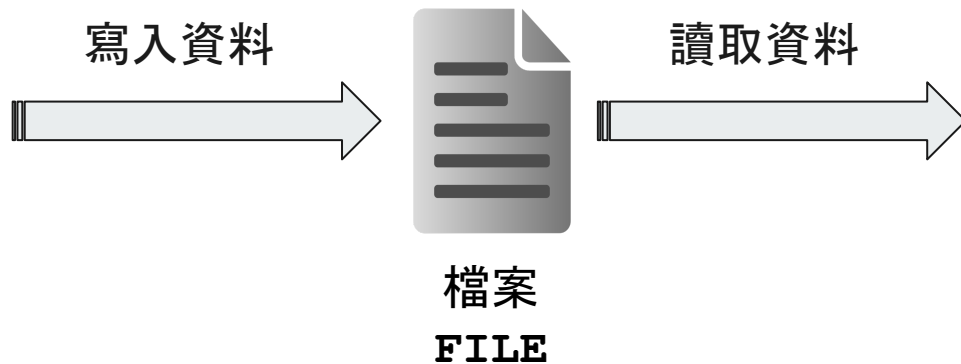


標準資料流的通訊管道



檔案資料流

C語言將被開啟的檔案視為能被寫入(write)與讀取(read)資料的資料流。





從資料流讀取資料的相關函式

從資料流讀取字元

```
int fgetc(FILE* stream);
```

從資料流讀取字串

```
int fgets(char* str, int num, FILE* stream);
```

從資料流讀取格式化字串

```
int fscanf(FILE* stream, const char* format, ...);
```

從資料流讀取二進制(binary)資料

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





寫入資料到資料流的相關函式

寫入字元到資料流

```
int fputc(int character, FILE *stream);
```

寫入字串到資料流

```
int fputs(const char *str, FILE *stream);
```

寫入格式化字串到資料流

```
int fprintf(FILE *stream, const char *format, ...);
```

寫入二進制(binary)資料到資料流

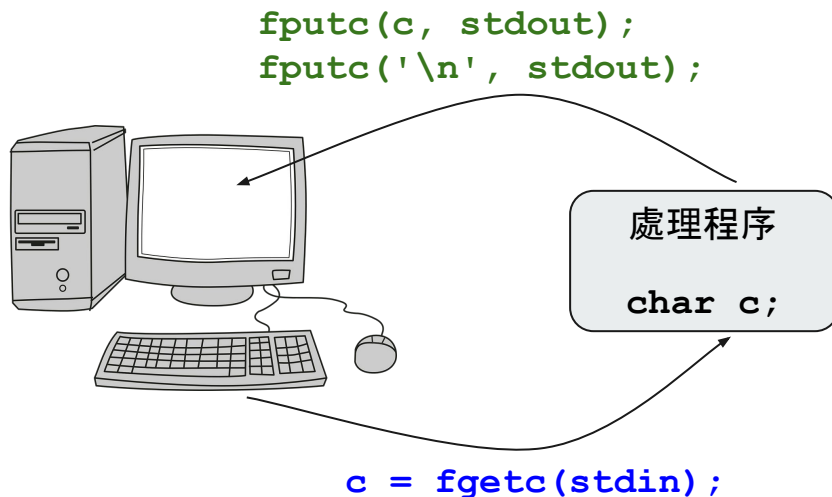
```
size_t fwrite(const void *ptr, size_t size, size_t count, FILE * stream );
```



使用資料流的範例

理解資料流的概念後就可以透過上述幾個函式來對檔案(或如鍵盤、螢幕等輸出入介面)進行處理。例如

1. `char c;`
2. `c = fgetc(stdin);`
3. `fputc(c, stdout);`
4. `fputc('\n', stdout);`





練習：格式化輸出入

以標準資料流

讓使用者在鍵盤上輸入"Hello 人名"後

在螢幕上輸出"Bye 人名"

限制條件：

使用標準資料流

| 輸入 | 輸出 |
|-------------|--------------|
| Hello,Bob | Bye Bob |
| Hello,Alice | Bye Alice |
| Hello Bob | Format Error |

A few moments later ...



開啟/關閉檔案資料流

可以透過下列函式來開啟/關閉檔案資料流(並以[FILE](#)結構指標指向資料流)

開啟檔案

`FILE* fopen(const char* filename, const char* mode);` //mode可決定讀寫模式

關閉檔案

`int fclose(FILE* stream)`

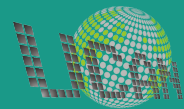


檔案處理的基本程式架構

```
1.  FILE *fp;
2.  fp = fopen([path/檔案路徑], [mode/開啟模式]);
3.  if(fp != NULL) { // 檢查fp是否為NULL來判斷是否開檔成功
4.      //在這邊處理檔案
5.  } else {
6.      printf("File open error!\n");
7.      return EXIT_FAILURE;
8.  }
9.  fclose(fp);
```



處理文字檔案



寫入字串到文字檔案

```
1.  FILE*  fp;
2.  fp = fopen("./file.txt", "w"); //請問mode為w與a的差異為何?
3.  if(fp != NULL) { //檢查fp是否為NULL來判斷是否開檔成功
4.      fputs("Bob,100,90,80\n", fp); //姓名,國文成績,英文成績,數學成績
5.      fputs("Alice,90,100,90\n", fp);
6.  } else {
7.      printf("File open error!\n");
8.      return EXIT_FAILURE;
9.  }
10. fclose(fp);
```



從文字檔案讀取字串

```
1.  FILE* fp;
2.  fp = fopen("./file.txt", "r");
3.  if(fp != NULL) { //檢查fp是否為NULL來判斷是否開檔成功
4.      char str[256];
5.      while(fgets(str, 256, fp) != NULL) { //判斷是否讀到最後一行
6.          fputs(str, stdout);
7.      }
8.  } else {
9.      printf("File open error!\n");
10.     return EXIT_FAILURE;
11. }
12. fclose(fp);
```





字串字符(Token)處理

學生資料格式

姓名, 國文成績, 英文成績, 數學成績

如何個別取出存入結構?

```
1.  typedef struct {  
2.      char name[10];  
3.      char chi, eng, mat;  
4.  } student;
```

檔案內容範例

Bob, 100, 90, 80

Alice, 90, 100, 90



字串字符處理函式

將字串切割成字符

```
char* strtok(  
    char* str,                //原始字串  
    const char* delimiters    //切割字串  
);
```

如果存在下一個字符，回傳指向字符的指標。
否則，回傳NULL。

```
1.  #include<string.h>  
2.  ...  
3.  char *record = "Bob,100,90,80";  
4.  char *tok = strtok(record, ","); //取出第一個字符  
5.  while(tok != NULL) {  
6.      printf("%s\n", tok);  
7.      tok = strtok(NULL, ",");    //取出下一個字符  
8.  }
```



練習：學生成績(讀文字檔)

讀 取 檔 案 格 式 如 下

```
Bob,100,90,80
```

```
Alice,90,100,90
```

將每一行學生成績的字符取出重新印出

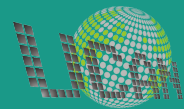
限制條件：

讀取文字檔並用字符切割

| 檔案 | 輸出 |
|----------------------------------|--|
| Bob,100,90,80 | Bob 100 90 80 |
| Bob,100,90,80 Alice,90,100,90 | Bob 100 90 80 Alice 90 100 90 |

A few moments later ...

處理二進制檔案





為何要儲存為二進制檔案?

通常比較節省空間, 例如

以文字與二進制方式儲存10000000個需要 ? 個位元組與 ? 個位元組

通常處理時間比較快, 例如

需要再額外呼叫`atoi("10000000")`將字串轉為數字

有些資料本身就是人看不懂的資料, 例如

加密後的資料、音樂檔、影片檔等



寫入資料到二進制檔案

```
1.  typedef struct {
2.      char name[6];
3.      unsigned char prog;
4.      unsigned char csie;
5.  } student;
6.  ...
7.  student std[2] = {
8.      {"Bob",100,100},
9.      {"Alice",90,90}
10. };
```

```
1.  FILE* fp;
2.  fp = fopen("./score.bin", "wb");
3.  if(fp != NULL) {
4.      //從記憶體位址std讀取2筆student型別的資料寫入檔案fp
5.      fwrite(std, sizeof(student), 2, fp);
6.  } else {
7.      printf("File open error!\n");
8.      return EXIT_FAILURE;
9.  }
10. fclose(fp);
11. return EXIT_SUCCESS;
```



從二進制檔案讀取資料

```
1.  typedef struct {
2.      char name[5];
3.      unsigned char prog;
4.      unsigned char csie;
5.  } student;
6.  ...
7.  student std[2];
```

```
1.  FILE* fp;
2.  fp = fopen("./score.bin", "rb");
3.  if(fp != NULL) {
4.      //從檔案fp讀取2筆student型別的資料到記憶體位址std
5.      fread(std, sizeof(student), 2, fp);
6.      //印出資料
7.  } else {
8.      printf("File open error!\n");
9.      return EXIT_FAILURE;
10. }
11. fclose(fp);
12. return EXIT_SUCCESS;
```



Q & A



Computer History Museum, Mt. View, CA

