



第一題



3



第二題



4



11.1 簡介

■第一章已經學過所謂的資料階層。儲存在變數和陣列裡的資料是暫時的,當程式結束後這種資料都將會遺失。程式會利用檔案 (Files) 來長期地保存大量的資料。電腦會將檔案存放在輔助性儲存裝置,像是硬碟、CD、DVD和快閃儲存裝置。本章中,我們將介紹C程式如何建立、更新和處理資料檔案。

WATSE

11.2 檔案和串流

❖C將每個檔案視為連續的位元組串流 (參見圖In)。 在程式開始執行時,會有三個檔案,還有和這些 檔案結合的資料流也會自動開啟,它們是標準輸 入 (standard input)、標準輸出 (standard output) 和標準錯誤 (standard error)。標準輸入、標準輸 出、和標準錯誤會分別以下列三個指標來進行操 作:stdin、stdout和stderr。

0 1 2 3 4 5 6 7 8 9 ... n-1 檔案結尾記號

圖11.1 C將檔案視為具有n個位元組的資料

11.3 開檔與關檔



- ❖檔案類型:文字檔、二進位檔
- ❖ 開檔 fopen()

#include <stdio.h>

FILE *fopen(const char *path, const char *mode)

- 回傳:函式回傳一個FILE指標指向檔案開頭,若 開啟檔案有誤則回傳NULL。
- path: 字串變數,代表檔案的檔名(含路徑)
- mode: "r", "w", "a", "r+", "w+", "a+", "?b"
- ❖ 關檔fclose()

int fclose(FILE *stream)



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:以二進位模式開啟或是建立一個檔案,將資料寫到檔案尾端。

#include <stdlib.h>

#include <stdio.h>

```
FILE *fp1;
void main(void){
    char filename[80];
    printf("請輸入檔名(可含路徑):");
    scanf("%s", filename);
    fp1 = fopen(filename, "r");
    if(fp1!= NULL)
        printf("檔案%s開啟中...\n", filename);
    else{
        printf("檔案%s開啟失敗\n", filename);
        exit(1); /* 強迫結束程式*/ }
    fclose(fp1);
    printf("檔案%S關閉\n", filename);
    system("pause");
```

WATSE

9

讀寫字串函式

}



- 1. 回傳值:回傳字串或NULL
- 2. s:存放讀入的字串
- 3. size: 最長的讀入字串長度
- 4. stream:已開啟的檔案指標

int fputs(const char *s, FILE *stream);

Fill the input from stdin to a file (I)



```
#include <stdio.h>
int main( void )
{
    char content[ 200 ]; // input string
    FILE *cfPtr; // cfPtr = clients.dat file pointer
    // fopen opens file. Exit program if unable to create file
    if ( ( cfPtr = fopen( "clients.dat", "w" ) ) == NULL ) {
        puts( "File could not be opened" );
    } // end if
```

11



Fill the input from stdin to a file (II)



```
else {
    puts( "Enter the basic information of yours." );
    puts( "Enter EOF to end input." );
    gets(content);
    // write content into file with fprintf
    while ( !feof( stdin ) ) {
        fputs( content, cfPtr );
        gets( content );
      } // end while
    fclose( cfPtr ); // fclose closes file
    } // end else
} // end main
```

11.3 建立一個循序存取檔案



❖不同系統中代表檔案結尾的按鍵組合。

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

13



Copy a file to another one(I)

#include <stdlib.h>



14



Copy a file to another one(II)



```
printf("讀取並寫入中.....\n");

while((fgets(tempstr,80,fp1))!=NULL) /* 使用迴圖讀取來源檔內容 */
{
    printf("%s",tempstr);
    fputs(tempstr,fp2); /* 使用迴圖寫入目的檔內容 */
}
printf("\n讀取並寫入完畢......\n");

fclose(fp1);
fclose(fp2);
system("pause");
}
```

15



格式化檔案輸出入函式

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

從檔案中讀出一段固定格式的文字

例如:

fscanf(cfPtr, "%d %d %d", &id, &Math, &Comp);

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

寫入一段格式化文字到檔案串流中

例如:

fprintf(cfPtr, ''%d %d %d \n'', id, Math, Comp);



Copy a file to another one(II) - alternate



```
printf("讀取並寫入中......\n");
```

17



本周作業



- ❖請試著將成績排序的程式改成 (非struct)
 - 從鍵盤讀取
 - 寫入檔案
 - 排序(由大排到小)
 - ■輸出至螢幕與檔案中
- ❖ 第二步再將成績排序程式改成
 - ■由檔案讀取
 - 排序(由小排到大)
 - ■輸出至螢幕與檔案中