第六週

王子銜, 陳毅軒, 吳尚龍

電機通訊程式設計

March 25, 2024

Outline

開檔

宣告

FILE*

在讀檔寫檔時，需要一個指標指向檔案的位址。

```
1  FILE *fp;
2  // fp 指向變數爲 FILE 位址
```

fopen

用 fopen 來開檔，把前面宣告的 fp 指標指向 fopen 回傳的位址。

```
1  FILE *fp = fopen (const char *filename, const char *mode);
2  // fp 指向 fopen 回傳的位址
```

fopen 使用方法

- filename - 字串，表示要開啓的檔案名稱。
- mode - 字串，表示檔案的存取模式，可以是以下的值：

"r"	讀取模式，僅可讀取，若檔案不存在會回傳 NULL
"w"	建立一個僅可寫入的空白檔案 如果檔案名稱已存在，則會刪除現有檔案。
"a"	附加模式且僅可寫入 如果檔案名稱不存在，則會建立新的檔案。
"r+"	可讀取也可寫入，如果檔案不存在會回傳 NULL
"w+"	建立一個用於讀寫的空檔案 如果檔案名稱已存在，則會刪除現有檔案。
"a+"	附加模式且可讀寫 如果檔案名稱不存在，則會建立新的檔案。

傳回值

傳回值

該函數回傳一個 FILE 指標，否則回傳 NULL。

防呆

可以使用以下片段來測試檔案是否開啓成功

```c
if (fp == NULL) {
  perror("檔案開啓失敗"); // 將訊息輸出至 stderr
  exit(1);
}
```

讀檔

讀檔函式

fgetc、getc

會在檔案中讀取一個字元，fgetc 跟 getc 用法一樣
會回傳讀入的字元，或是回傳 EOF (end of file)

```
1  // file.txt: aaa
2  FILE *fp = fopen("file.txt", "r");
3  char c = fgetc(fp);   // getc(fp) 也可以
4  printf("%c", c);
5  // output: a
```

讀檔函式

fgets

會在檔案中讀取一整行，需給定 buffer、長度上限
會回傳 buffer 的指標，或是回傳 NULL

```c
// file.txt: aaa aaa
FILE *fp = fopen("file.txt", "r");
char buffer[100];
fgets(buffer, sizeof(buffer) / sizeof(buffer[0]), fp);
printf("%s", buffer);
// output: aaa aaa
```

讀檔函式

fscanf

可以跟 scanf、printf 一樣指定格式化字串
會回傳成功讀了幾個東西，或是回傳 EOF

```
1  // file.txt: 48763 aaaaaa
2  FILE *fp = fopen("file.txt", "r");
3  int x; char buffer[100];
4  fscanf(fp, "%d", &x);
5  fscanf(fp, "%s", buffer);
6  printf("%d ", x);
7  printf("%s", buffer);
8  // output: 48763 aaaaaa
```

讀檔函式

feof

會回傳布林值，如果當前檔案已經讀到EOF，回傳 true，否則回傳 false

```
1  // file.txt: 48763 aaaaaa
2  FILE *fp = fopen("file.txt", "r");
3  while (!feof(fp)) {
4    char c = fget(fp);
5    printf("%c", c);
6  }
7  // output: 48763 aaaaaa
```

寫檔

寫檔函式

fputc、putc

會在檔案中寫入一個字元，fputc 跟 putc 用法一樣
會回傳寫入的字元，寫入失敗會回傳 EOF

```
1  FILE *fp = fopen("file.txt", "w");
2  fputc('a', fp);   // putc('a', fp) 也可以
3  // file.txt: a
```

寫檔函式

fputs

會在檔案中寫入一個字串，需給定 buffer，不會自動換行
會回傳非負值，寫入失敗會回傳 EOF

```
1  FILE *fp = fopen("file.txt", "w");
2  char buffer[] = "48763";
3  fputs(buffer, fp);
4  fputs("48763", fp);
5  // file.txt: 4876348763
```

寫檔函式

fprintf

可以跟 scanf、printf 一樣指定格式化字串
會回傳成功寫入了幾個位元，寫入失敗會回傳負值

```
1  FILE *fp = fopen("file.txt", "w");
2  fprintf(fp, "%s %d", "kirito", 48763);
3  // file.txt: kirito 48763
```

補充

fseek 使用方法

移動 FILE 指標的位置

```
1  int fseek(FILE *_File, long _Offset, int _Origin)
```

- offset 放置字元的偏移量，-1 代表往回一個字元，1 代表前進一個字元
- origin 放置參考點，可以是文件頭、目前字元、文件尾
  - SEEK_SET: 文件頭
  - SEEK_CUR: 目前 fp 的位置
  - SEEK_END: 文件尾

作業說明

作業說明

因爲這週是普物期中考

爲了減輕各位的負擔

我們把沒什麼好考的主題放在這週

本週只有一道 demo 題

各位應該都能在課堂上完成

詳情請看 DomJudge 的題本