

鋼彈盪單槓

鋼彈的 coding 筆記



C語言-檔案I/O

📅 2015-10-30 | 📅 2021-10-27 | 📁 C語言 | 💬 0 Comments

📄 6k | ⌚ 5 分鐘

這篇是講關於C語言如何操作檔案

簡介



儲存在變數的資料只是暫時的，想要在程式結束時儲存資料就必須靠檔案

這個章節要介紹的是如何操作檔案

fopen, fclose

如果要處理檔案要用fopen來開啟檔案，fopen函數的原型宣告如下：

```
1 FILE* fopen (const char*, const char*);
```

fopen的第一個參數是檔案名稱(路徑)，第二個參數是打開檔案的方式

以下表格是檔案打開的方式

模式	說明
r	開啟檔案進行唯讀，若檔案不存在，則傳回NULL
w	開啟檔案進行唯寫，若檔案不存在，則建立新檔，若檔案存在則將之刪除，再建立新檔
a	開啟檔案進行附加，若檔案存在，則資料從檔案尾端寫入，若檔案不存在則建立新檔
rb	以二進位模式開啟檔案進行唯讀
wb	以二進位模式開啟檔案進行唯寫

模式	說明
<code>ab</code> ↑	以二進位模式開啟檔案進行附加
<code>r+</code>	開啟檔案進行讀寫，若檔案不存在，則建立新檔，若檔案存在，資料將從檔案開頭進行覆寫
<code>w+</code>	開啟檔案進行讀寫，若檔案不存在，則建立新檔，若檔案存在則覆寫原有的資料
<code>a+</code>	開啟檔案進行附加、讀取，若檔案不存在則建立新檔，若檔案存在，則資料從檔案尾端寫入
<code>r+b</code>	以二進位方式開啟檔案進行讀寫
<code>w+b</code>	以二進位方式開啟檔案進行讀寫
<code>a+b</code>	以二進位方式開啟檔案進行附加、讀取

`fopen`會使用緩衝區來減少I/O，以提高效率，所以在讀寫的過程中其實是對緩衝區做讀寫

使用檔案的好習慣就是不使用時就要使用`fclose`關閉，參數為檔案指標，以下為原型宣告

```
1  int fclose(FILE *fp);
```

如果檔案正常關閉則回傳0，不正常關閉則回傳非0的數字

fgetc, fputc, fgets, fputs

開啟檔案後可以用`fgetc`讀取檔案的一個字元，用`fputc`輸出一個字元

函數宣告原型如下

```
1  int fgetc(FILE* fp);
2  int fputc(int ch, FILE *fp);
```

`fgetc`傳檔案指標進去就可以取得該檔案的一個字元，直到檔案結尾EOF(End Of File)

可以像下面這樣判斷檔案結尾

```
1  char ch;
2  ch = fgetc(file);
3  while( ch != EOF ) {
```

```
4     ...
5     ...
6     ...
7     ch = fgetc(file);
8 }
↑
0
```

fputc傳一個字元跟檔案指標就能寫一個字元進去

而如果想一次處理一行字串就要使用fgets、fputs，函數宣告原型如下

```
1 char* fgets(char *str, int length, FILE *fp);
2 int fputs(char *str, FILE *fp);
```

fgets第一個參數是要儲存的字串位址，第二個是要讀幾個字，由於最後一個字要是'\0'

所以真正的長度為length-1，第三個參數是檔案指標

fputs第一個參數是要輸出的字串，第二個參數是要輸出到哪個檔案

以下程式是示範用fgetc,fputc複製檔案

```
1 #include <stdio.h>
2
3 int main() {
4
5     FILE *file1 = fopen("source.txt", "r");
6     if(!file1) {
7         puts("來源檔案開啟失敗");
8         return 1;
9     }
10
11     FILE *file2 = fopen("target.txt", "w");
12     if(!file2) {
13         puts("目的檔案開啟失敗");
14         return 1;
15     }
16
17     char ch;
18     ch = fgetc(file1);
19     while(ch != EOF) {
20
21         fputc(ch, file2);
22         ch = fgetc(file1);
```

```
23
24     }
25
26     fclose(file1);
27     fclose(file2);
28
29     return 0;
30
31 }
```

以下程式是示範用fgets,fputs複製檔案

```
1  #include <stdio.h>
2
3  int main() {
4
5      FILE *file1 = fopen("source.txt", "r");
6      if(!file1) {
7          puts("來源檔案開啟失敗");
8          return 1;
9      }
10
11     FILE *file2 = fopen("target.txt", "w");
12     if(!file2) {
13         puts("目的檔案開啟失敗");
14         return 1;
15     }
16
17     char str[50];
18     while(fgets(str, 50, file1) != NULL) {
19         fputs(str, file2);
20     }
21
22     fclose(file1);
23     fclose(file2);
24
25     return 0;
26
27 }
```

fscanf,fprintf

檔案操作也可以做格式化，以下是原型宣告

```
1  int fprintf(FILE *fp, char *formatstr, arg1, arg2, ...);
2  int fscanf(FILE *fp, char *formatstr, arg1, arg2, ...);
```

↑除了第一個參數要給檔案指標以外，其餘跟scanf,printf一樣

以下是範例程式

```
1  #include <stdio.h>
2
3  int main() {
4      char ch;
5
6      FILE *file = fopen("test.txt", "w");
7      if(!file) {
8          puts("無法寫入檔案");
9          return 1;
10     }
11
12     fprintf(file, "%s\t%d\r\n", "AAA", 100);
13     fprintf(file, "%s\t%d\r\n", "BBB", 90);
14     fprintf(file, "%s\t%d\r\n", "CCC", 80);
15
16     fclose(file);
17
18     file = fopen("test.txt", "r");
19     if(!file) {
20         puts("無法讀入檔案");
21         return 1;
22     }
23
24     char name[10];
25     int score;
26
27     puts("名字\t分數");
28     while(fscanf(file, "%s\t%d", name, &score) != EOF) {
29         printf("%s\t%d\n", name, score);
30     }
31
32     fclose(file);
33
34     return 0;
35
36 }
```

fread,fwrite

使用二進位模式讀寫檔案就需要用fread,fwrite



讀寫二進位檔案都是使用位元組(Byte)為單位的區塊(Block)，以下為函數原型宣告

```
1  int fread(char *buffer, int size, int count, FILE *fp);
2  int fwrite(char *buffer, int size, int count, FILE *fp);
```

fread的意思是將count個元素從fp檔案中讀出到buffer，每個元素大小為size位元組

fwrite的意思是從buffer寫入count個元素到fp檔案中，每個元素大小為size位元組

以下為複製檔案的範例程式

```
1  #include <stdio.h>
2
3  int main() {
4
5      FILE *file1 = fopen("source.txt", "rb");
6      FILE *file2 = fopen("target.txt", "wb");
7
8      if(!file1) {
9          puts("檔案讀入失敗");
10         return 1;
11     }
12
13     if(!file2) {
14         puts("檔案輸出失敗");
15         return 1;
16     }
17
18     char ch;
19
20     while(!feof(file1)) {
21         fread(&ch, sizeof(char), 1, file1);
22
23         if(!feof(file1)) {
24             fwrite(&ch, sizeof(char), 1, file2);
25         }
26     }
27
28     fclose(file1);
```

```
29     fclose(file2);
30
31     return 0;
32
33     ↑
34 }
```

fseek

開啟檔案的時候，會有一個指標指向檔案開頭，每次讀取幾個字元，就會往後位移

C語言提供fseek來位移這個指標，方便我們要從任一地方讀取檔案

以下為fseek的函數原型宣告

```
1  int fseek(FILE *fp, long offset, int mode);
```

第一個參數是檔案，第二個參數為偏移量，第三個參數是從哪裡位移

以下是位移模式的說明

位移模式	說明
SEEK_SET	檔案開頭
SEEK_CUR	目前游標所在位置
SEEK_END	檔案結尾

這種檔案操作稱為非循序的檔案存取，一般會使用二進位模式搭配自定的資料結構

以下是範例程式

```
1  #include <stdio.h>
2
3  struct Employee{
4      char name[30]; // 名字
5      int age; //年齡
6      char gender; // 性別, 'M' or 'F'
7      double salary; // 薪水
8  };
9
```

```
10 void printPrompt();
11
12 int main() {
13
14     // 初始化
15     FILE *file = fopen("data.bin", "wb");
16     int count;
17     printf("要建立幾筆資料? ");
18     scanf("%d", &count);
19
20     struct Employee employee = {"", 0, '\0', 0.0};
21
22     int i;
23     for(i = 0; i < count; i++)
24         fwrite(&employee, sizeof(employee), 1, file);
25
26     fclose(file);
27
28     // 讀取
29     count = 0;
30
31     file = fopen("data.bin", "r+b");
32     if(!file) {
33         puts("無法讀取檔案");
34         return 1;
35     }
36
37     while(1) {
38         fread(&employee, sizeof(employee), 1, file);
39         if(!feof(file)) {
40             count++;
41         }
42         else {
43             break;
44         }
45     }
46
47     rewind(file); // 等同於 fseek(file, 0, SEEK_SET);
48
49     printf("輸入序號(1-%d)\n", count);
50     puts("輸入0離開");
51
52     int id;
53     while(1) {
54         printf("\n序號? ");
55         scanf("%d", &id);
```



```
56         if(id == 0) {
57             break;
58         }
59
60         printf("輸入名字 年齡 性別 薪水\n=> ");
61         scanf("%s %d %c %lf", employee.name, &(employee.age), &(employee.gender),
62
63         fseek(file, (id - 1) * sizeof(employee), SEEK_SET);
64         fwrite(&employee, sizeof(employee), 1, file);
65
66         printf("輸入序號(1-%d)\n", count);
67         puts("輸入0離開");
68     }
69
70     fclose(file);
71
72     return 0;
73
74 }
```

執行結果

要建立幾筆資料? 5

現在開始輸入資料

輸入序號(1-5)

輸入0離開

序號? 1

輸入名字 年齡 性別 薪水

=> Gundam 20 M 81000

輸入序號(1-5)

輸入0離開

序號? 5

輸入名字 年齡 性別 薪水

=> Box 20 M 22000

輸入序號(1-5)

輸入0離開

序號? 0

現在輸出所有資料

名字	年齡	性別	薪水
----	----	----	----

Gundam	20	M	81000.00
	0	0	0.00
	0	0	0.00
↑	0	0	0.00
Box	20	M	22000.00

freopen

freopen跟fopen不一樣的地方是fopen是打開檔案

freopen是重新導向串流(stream)直到程式結束為止，以下是函數原型宣告

```
1 FILE * freopen ( const char * filename, const char * mode, FILE * stream );
```

第一個參數是檔案名稱，第二個是檔案模式，第三個則是要重新導向的串流

串流分為stdin(標準輸入)、stdout(標準輸出)、stderr(標準錯誤)

以下是範例程式

```
1  #include <stdio.h>
2
3  int main() {
4
5      // 把檔案導向到標準輸入
6      freopen("source.txt", "r", stdin);
7      // 把標準輸出重新導向到檔案
8      freopen("target.txt", "w", stdout);
9
10     char str[50];
11     while(gets(str) != NULL) {
12         puts(str);
13     }
14
15     return 0;
16
17 }
```

參考

- 1. [C語言](#)
- 2. [未格式化檔案 I/O](#)
- ↑ 3. [格式化檔案 I/O](#)
- 4. [二進位檔案 I/O](#)
- 5. [資料流游標](#)
- 6. [隨機存取檔案](#)

歡迎關注我的其它發布渠道



RSS

[# C語言](#)

[← C語言-struct、union、enum](#)

[C語言-標頭檔與前置處理器 >](#)



© 2021 ❤️ Gundambox

📈 106k | ☕ 1:36

由 [Hexo](#) & [NexT.Gemini](#) 強力驅動