

二進位檔案 I/O

GCP

產品階層無時無刻免費使用，厚植個人實戰經驗。



使用二進位模式開啟檔案，在寫入或讀取檔案時並不會發生字元轉換，數值在記憶體中的位元是如何，寫入檔案時就是如何，而讀入時也是相同。

要讀入二進位檔案，可以使用 `fread()` 函式，在讀寫時是使用位元組（`byte`）為單位的區塊（`block`）進行讀寫，所以也稱區塊 I/O 函式：

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

`buffer` 參數為讀入的資料所要存放的地方，`size`是指讀入的每一筆資料長度（`byte`），`count` 是指讀入的筆數。

對於二進位檔案而言，若要判斷檔案是否到尾端，不能用讀到 `-1` 來作判斷，因為檔案讀取的過程中，可能就含有 `-1` 的值，您可以使用 `feof()` 函式來判斷是否到檔案結尾：

```
while(!feof(file)) {  
    ....  
}
```

下面這個程式可以讀入任意檔案，每次讀入一個位元組，並將讀入的資料以 `16` 進位數顯示：

```
#include <stdio.h>

int main(int argc, char* argv[]) {
    if(argc != 2) {
        puts("指令: read <filename>");
        return 1;
    }

    FILE *file = fopen(argv[1], "rb");
    if(!file) {
        puts("無法讀取檔案");
        return 1;
    }

    unsigned char ch;
    int count = 0;

    while(!feof(file)) {
        fread(&ch, sizeof(char), 1, file);

        printf("%2X ", ch);
        count++;
        if(count > 16) { // 換行
            putchar('\n');
            count = 0;
        }
    }
    putchar('\n');
```

```
    fclose(file);

    return 0;
}
```

執行結果：

read main.c

```
23 69 6E 63 6C 75 64 65 20 3C 73 74 64 69 6F 2E 68
3E 20  A 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  A 69
6E 74 20 6D 61 69 6E 28 69 6E 74 20 61 72 67 63 2C
20 63 68 61 72 2A 20 61 72 67 76 5B 5D 29 20 7B  A
20 20 20 20 69 66 28 61 72 67 63 20 21 3D 20 32 29
```

略...

如果要寫入檔案，可以使用 `fwrite()`：

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

下面這個程式可以讓您將任意的檔案複製至另一個指定的名稱：

```
#include <stdio.h>
```

```
int main(int argc, char* argv[]) {
    if(argc != 3) {
```

```
        puts("指令: copy <input> <output>");
        return 1;
    }

    FILE *file1 = fopen(argv[1], "rb");
    FILE *file2 = fopen(argv[2], "wb");

    if(!file1) {
        puts("檔案讀入失敗");
        return 1;
    }

    if(!file2) {
        puts("檔案輸出失敗");
        return 1;
    }

    char ch;

    while(!feof(file1)) {
        fread(&ch, sizeof(char), 1, file1);

        if(!feof(file1)) {
            fwrite(&ch, sizeof(char), 1, file2);
        }
    }

    fclose(file1);
```





```
fclose(file2);

return 0;
}
```

下面這個程式則示範如何將陣列資料寫入檔案，然後再將之讀出：

```
#include <stdio.h>
```

```
int main(int argc, char* argv[]) {
    FILE *file = fopen("temp", "wb");
    if(!file) {
        puts("無法讀取檔案");
        return 1;
    }

    int arr[5] = {1, 2, 3, 4, 5};
    fwrite((char*) arr, sizeof(arr), 1, file);
    fclose(file);

    file = fopen("temp", "rb");
    if(!file) {
        puts("無法讀取檔案");
        return 1;
    }

    fread((char*) arr, sizeof(arr), 1, file);
    printf("arr: ");
    int i;
```

```
for(i = 0; i < 5; i++) {  
    printf("%d ", arr[i]);  
}  
putchar('\n');  
  
fclose(file);  
  
return 0;  
}
```





PayPal 線上收款

廣告 PayPal



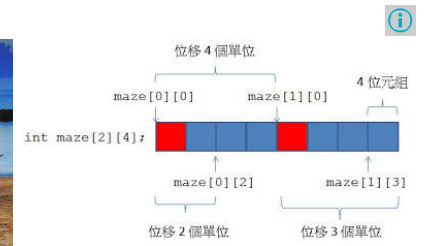
語言技術：C++ Gossip

openhome.cc



Shutterstock Stock Photos

廣告 Shutterstock



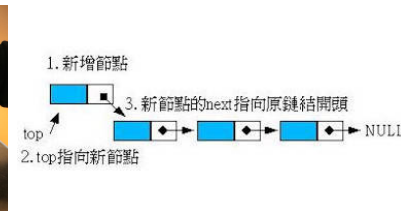
二維陣列

openhome.cc

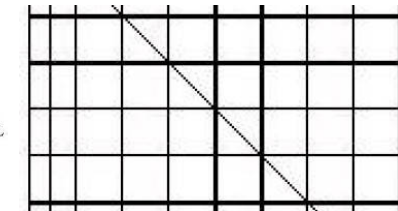


Cryptobank ICO

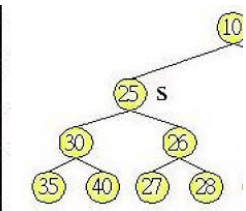
廣告 Crypterium ICO

堆疊- 使用鏈結實作 (C 語 八個皇后
言動態記憶體宣告)

openhome.cc



openhome.cc

Heap 排序法 - 堆
排序

openhome.cc