

# 第十二章

## 檔案 (File)

# 大綱

檔案處理簡介

文字檔案處理函數(使用緩衝區)

二進位資料處理函數(使用緩衝區)


二進位資料處理函數(不使用緩衝區)

# 檔案的分類

## ■ 檔案儲存方式

- 存放3392
- 文字檔(儲存ASCII code)

51 (3)	51 (3)	57 (9)	50 (2)
--------	--------	--------	--------

 4 bytes

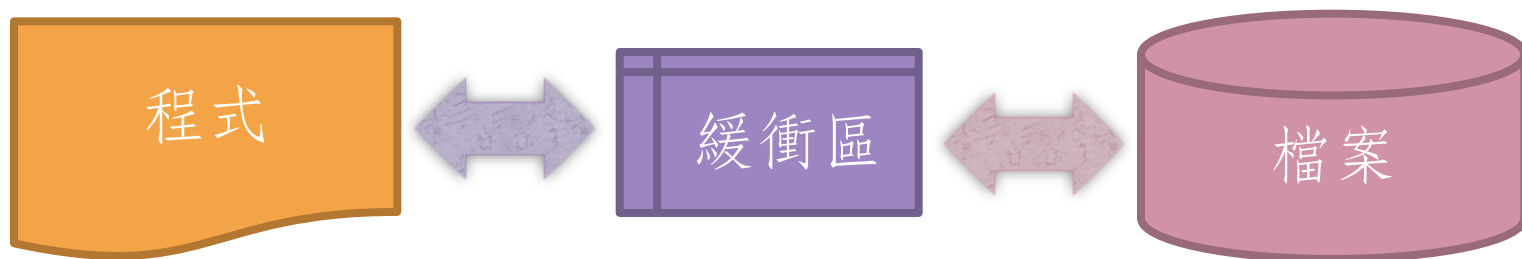
- 二進位檔

1101	0100	0000
------	------	------

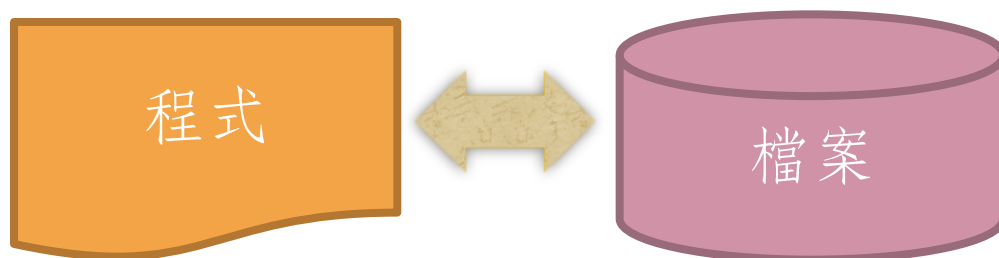
 3 bytes

# 檔案處理

## ■ 有緩衝區的處理



## ■ 無緩衝區的檔案處理



# 檔案處理流程

## 開檔

- 檢查檔案是否正確開啟

## 處理檔案

- 讀取內容
- 寫入內容

## 關檔

# 檔案處理函數

## ■ 有緩衝區的檔案處理函數

動作	敘述	註解
開檔	<code>FILE * fopen(const char *filename, const char* mode);</code>	<code>mode={"r", "w", "a"}</code>
關檔	<code>int fclose(FILE *fptr);</code>	
檢查檔尾	<code>int feof(FILE *fptr);</code>	
讀取字元	<code>int fgetc(FILE *fptr);</code>	
寫入字元	<code>int fputc(int c, FILE *fptr)</code>	
讀取字串	<code>char* fgets(char* str, int max, FILE *fptr);</code>	
寫入字串	<code>int fputs(const char* str, FILE *fptr);</code>	
區塊輸入	<code>size_t fread(void *p, size_t size, size_t count, FILE *fptr);</code>	
區塊輸出	<code>size_t fwrite(const void *p, size_t size, size_t count, FILE *fptr);</code>	



# 開啟檔案的範例

- 宣告檔案變數

- 開啟檔案

- 成功則讀取；失敗則顯示錯誤訊息

```
FILE *fptr;  
  
fptr=fopen("ch12-01.txt","r");  
if(fptr!=NULL)  
{  
  
    fclose(fptr);  
}  
else  
{  
    printf("OPEN FAILED!\n");  
}
```

# 處理文字

## ■ 留意讀到分行符號的處理方式

```
fptr1=fopen("ch12-01.txt","r");  
fptr2=fopen("ch12-02.txt","w");  
if(fptr1!=NULL && fptr2!=NULL)  
{  
    while((ch=fgetc(fptr1)) != EOF)  
    {  
        fputc(ch,fptr2);  
        printf("%c",ch);  
    }  
    fclose(fptr2);  
    fclose(fptr1);  
}
```



# 處理字串

## ■ 留意字串空間是否足夠

```
const int MAX = 80;
char ch[MAX];

fptr1=fopen("ch12-01.txt","r");
fptr2=fopen("ch12-03.txt","w");
if(fptr1!=NULL && fptr2!=NULL)
{
    while (fgets(ch,MAX,fptr1)!=NULL)
    {
        printf("%s",ch);
        fputs(ch,fptr2);
    }
    fclose(fptr2);
    fclose(fptr1);
}
```

# 處理區塊

## ■ 緩衝區應有足夠大小容納所讀入資料

```
char ch[MAX+1];
int bytes = 0;
fptr1=fopen("ch12-01.txt","r");
fptr2=fopen("ch12-04.txt","w");
if(fptr1!=NULL && fptr2!=NULL)
{
    while (!feof(fptr1))
    {
        bytes = fread(ch,sizeof(char),MAX,fptr1);
        fwrite(ch,sizeof(char),bytes,fptr2);
        if (bytes < MAX)
            ch[bytes] = 0;
        printf("%s",ch);
    }
    fclose(fptr2);
}
```



# 二進位資料的儲存

- `fwrite()` 可以一次寫入多(count)筆資料

```
FILE *f2;  
int i2=100;  
double d2=200.66;  
int a[5] = {1,2,3,4,5};  
  
f2=fopen("ch12-07.dat", "w");  
if (f2!=NULL)  
{  
    fwrite(&i2, sizeof(int), 1, f2);  
    fwrite(&d2, sizeof(double), 1, f2);  
    fwrite(a, sizeof(int), 5, f2);  
    fclose(f2);  
}
```

# 二進位資料讀取

- fread() 可以一次讀入多(count)筆資料

```
FILE *f1;
int i2;
double d2;
int a[5];

f1=fopen("ch12-07.dat","r");
if(f1!=NULL)
{
    fread(&i2,sizeof(int),1,f1);
    fread(&d2,sizeof(double),1,f1);
    fread(a,sizeof(int),5,f1);
    fclose(f1);
}
```

# 無緩衝區之二進位資料儲存

## ■ write()一次寫入一筆資料

```
struct st_date
{
    int year;
    int month;
    int day;
};

int main(void)
{
    int f2,cnt=2,i;
    struct st_date day[cnt]={1999,9,1},{2001,3,3}};

    f2=open("ch12-08.dat",O_WRONLY|O_CREAT);
    write(f2,&cnt,sizeof(int));
    for (i=0;i<cnt;i++)
        write(f2,&day[i],sizeof(struct st_date));
    close(f2);
}
```



# 無緩衝區之二進位資料讀取

## ■ read()一次讀入一筆資料

```
struct st_date
{
    int year;
    int month;
    int day;
};

int main(void)
{
    int f2,cnt=2,i;
    struct st_date day[cnt];

    f2=open("ch12-08.dat",O_RDONLY);
    read(f2,&cnt,sizeof(int));
    for (i=0;i<cnt;i++)
        read(f2,&day[i],sizeof(struct st_date));
    close(f2);
}
```



Any question?

