

Group_17

- lseek(file_description, offset, where)
lseek 函式可以把檔案內當前的讀寫位置(file pointer)移到指定位置(where) + 偏移量(offset)。
先用 lseek 找到 read/write 的位置，再使用函式進行 read/write。
- read(file_description, buffer, cnt)
從當前的讀寫位置開始，往後將cnt個元素丟到buffer裡
- write(file_description, buffer, cnt)
從當前的讀寫位置開始，往後將cnt個元素從buffer寫到檔案裡
- program 實作

```
#include <fcntl.h>
#include <unistd.h>

#define BUFF_MAX 4096

int main() {
    int fd = open("sample.txt", O_RDWR);
    char buffer[BUFF_MAX];
    const char* target = "NTHU student.";

    // 將file pointer 利用 lseek 移動到 sample.txt 中倒數第 9 個字元
    lseek(fd, -9, SEEK_END);

    // 利用 read 將當前 pointer +0 至 pointer +9 的字元讀進 buffer 中
    read(fd, buffer, 9);

    // 再次利用 lseek 將 pointer 移回上次移到的位置
    lseek(fd, -9, SEEK_END);

    // 利用 write 從當前 pointer 開始寫入 sizeof(target) 個字元
    write(fd, target, sizeof(target));

    return 0;
}
```

Implementation

- Result:

```
group_17@raspberrypi:~/Advanced-UNIX-Programming_Student/assignment1 $ ls
assignment1.c Makefile sample.txt
group_17@raspberrypi:~/Advanced-UNIX-Programming_Student/assignment1 $ make
gcc -c assignment1.c -o assignment1.o
gcc -std=c11 -O2 -Wall -o assignment1 assignment1.o
group_17@raspberrypi:~/Advanced-UNIX-Programming_Student/assignment1 $ ls
assignment1 assignment1.c assignment1.o Makefile sample.txt
group_17@raspberrypi:~/Advanced-UNIX-Programming_Student/assignment1 $ ./assignment1
student.
Hello, I am a NTHU student.group_17@raspberrypi:~/Advanced-UNIX-Programming_Student/assignment1 $
```

- code:

```

1 #include <unistd.h>
2 #include <stdio.h>
3 #include <fcntl.h>
4
5 int main() {
6     const char student[] = "student.";
7     const char nthu_student[] = "NTHU student.";
8     char buffer[80] = {};
9     int fd = open("sample.txt", O_RDWR);
10    if(fd != -1) {
11        int offset = - (int) sizeof(student);
12        lseek(fd, offset, SEEK_END);
13        read(fd, buffer, sizeof(student));
14        write(STDOUT_FILENO, buffer, sizeof(buffer) - 1);
15        lseek(fd, offset, SEEK_END);
16        write(fd, nthu_student, sizeof(nthu_student) - 1);
17        lseek(fd, 0, SEEK_SET);
18        read(fd, buffer, sizeof(buffer));
19        write(STDOUT_FILENO, buffer, sizeof(buffer) - 1);
20    } else {
21        puts("QQ");
22    }
23    return 0;
24 }

```

- Line 9:
利用 open 將 sample.txt 讀入。fd 不為 -1 表示讀取成功。
- Line 12:
利用 lseek 將 pointer 調整到 student. 的字元 's'。
- Line 13:
利用 read 將該字元至文件結尾的字元全部讀進 buffer。
- Line 14:
在 terminal 上輸出字串 buffer。
- Line 15:
由於 read 會改變 pointer 的位置，我們再次利用 lseek 調整 pointer, 同 Line 11。
- Line 16:
利用 write 將 NTHU student. 從當前 pointer 開始寫入。
- Line 17:
利用 lseek 將 pointer 調整到文件首。
- Line 18:
利用 read 讀取最多 80 個字元存進 buffer，讀取字元上限不超過文件總字元。
- Line 19:
利用 write 將字串 buffer 輸出在終端機上。