

一、設計

為了編譯老師給的 sample code，我們將 kernel 降到 3.2 之後，sample code 便可以順利編譯

接著是實做 mmap 的部份與 sudo code

user_program/master.c:

利用 mmap 給予每個 page 一個 address，再用 memcpy 將檔案傳給 master device，最後用 munmap 來解除記憶體映射

```
case 'm':
    while(not finish file reading){
        file_address = mmap();
        kernel_address = mmap();
        memcpy(file_address, kernel_address);
        munmap(file_address);
        munmap(kernel_address);
    }
```

master_device/master_device.c:

在 switch 的 case master_IOCTL_MMAP 中，利用 ksocket 中的 ksend function 來將 master.c 丟過來的檔案傳給 slave_device

case master_IOCTL_MMAP:

```
    ksend(master_file);
```

slave_device/slave_device.c:

在 switch 的 case slave_IOCTL_MMAP 中，利用 ksocket 中的 krecv function 來將 master_device 傳過來的檔案收下來，收到之後再用 memcpy 傳給 user program 的 slave

case slave_IOCTL_MMAP:

```
    while(not finish file sending){
        krecv(master_file, buffer);
        memcpy(buffer, user_slave);
    }
```

user_program/slave.c:

一樣先用 mmap 得到映射的記憶體位置，然後利用 memcpy 將收到檔案的記憶體內容全部 copy 到要輸出的檔案裡

case 'm':

```
    while(buffer not empty){
        file_address = mmap();
        kernel_address = mmap();
        memcpy(file_address, kernel_address);
        munmap(file_address);
        munmap(kernel_address);
    }
```

二、測試

test_case 1

```
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 1 f f
Transmission time: 0.324400 ms, File size: 4 bytes
Transmission time: 0.301500 ms, File size: 4 bytes
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 1 m m
Transmission time: 0.241800 ms, File size: 4 bytes
Transmission time: 0.261800 ms, File size: 4 bytes
```

test_case 2

```
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 2 f f
Transmission time: 0.014300 ms, File size: 577 bytes
Transmission time: 0.028200 ms, File size: 577 bytes
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 2 m m
Transmission time: 0.286300 ms, File size: 577 bytes
Transmission time: 0.304900 ms, File size: 577 bytes
```

test_case 3

```
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 3 f f
Transmission time: 0.463500 ms, File size: 9695 bytes
Transmission time: 0.492400 ms, File size: 9695 bytes
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 3 m m
Transmission time: 0.010200 ms, File size: 9695 bytes
Transmission time: 0.080300 ms, File size: 9695 bytes
```

test_case 4

```
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 4 m m
Transmission time: 23.134900 ms, File size: 1502860 bytes
Transmission time: 23.165500 ms, File size: 1502860 bytes
nie@nie-VirtualBox:~/OSproject2-master$ sudo ./test.sh 4 f f
Transmission time: 3.599000 ms, File size: 1502860 bytes
Transmission time: 5.853500 ms, File size: 1502860 bytes
```

三、結論與比較

四份範例資料大小逐個遞增，但是並未出現 transmission time fcntl 一定大於 mmap 的情況。甚至對於不同文件，同一種方式，transmission time 也並未隨文件增大而增大。

後來進一步發現，多次執行同一文件的同一種方式，transmission time 也會發生變化，總之 transmission time 和以下幾個事件有關係：

- 1, 測試的順序，若先測 mmap 再測 fcntl，fcntl 的時間就會很短
- 2, 測試完一種方式後，是否刪除 file_out
- 3, 其他不知道的因素

總之，I/O 問題沒有特別嚴格的規律性