

OS-Project2 Report

第32組

b05203047 徐衍新

b05902114 陳昕璘

b05902126 詹丰億

Progriammng design

主要依據Sample Code進行修改。在user program中，透過mmap映射char指標，接著再用memcpy來傳資料，以及使用ioctl設定IP address。一開始fcntl是正常的，但是當我們使用mmap來映射位置的時候，會出現Segmentation Fault。後來發現問題是mmap指向垃圾空間，導致memcpy時出錯。而kernel program方面，將mmap的case補上，用ksend和krecv，經由local node傳送data，反覆直到資料傳送完畢。

Performance comparison between file I/O and memory-mapped I/O

（以下測試皆為固定master，求10次執行時間之平均）

file 1

	time
slave-mmap	0.782620
slave-fcntl	0.683710

file 2

	time
slave-mmap	0.700340
slave-fcntl	0.542980

file 3

	time
slave-mmap	0.765760
slave-fcntl	0.898960

file 4

	time
slave-mmap	2.614450
slave-fcntl	6.332570

Result

```
root@yi-VirtualBox:/home/yi/OS-project2/user_program# ./slave ../data/file1_out fcntl 127.0.0.1
ioctl success
Transmission time: 0.093300 ms, File size: 4 bytes
```

(此為file1的slave-mmap)

看得出在file size小的時候fcntl的速度較快，但是隨著file size增加，mmap漸漸地顯出速度優勢。我們分析原因應該是，因為mmap的呼叫需考慮overhead（處理kernel以及VMM...），而file I/O是對file descriptor操作，較為抽象化且單純，所以檔案小的時候，mmap的overhead會佔據大部分時間，速度不及file I/O。然而，若檔案很大，file I/O的page fault將是一項昂貴成本，而且要對有大量copy cache動作，此時mmap速度就會更快。

Work list of team members

b05902126：程式架構設計，除錯
b05902114：測試環境，比較結果
b05203047：除錯，比較結果