Report

1. Program design

Clean 等外圍的是為了清理出現的東西方便而組合的指令。Ksocket 是把所需用到的所有 socket 給全部組合了起來。
最重要的三個:program_device，slave 和 master

user_program:
因為查到的某些 makefile 呼叫的東西在 4.x 版本沒辦法跑，所以我就使用了我虛擬機原本就安裝的 3.x 版本，使得整個 makefile 順利讀取。

master_init: 建立可存取 register 的檔案，重新設定函式可存取的記憶體空間、位址的數據，以及基本的 socket 連接(bind、listen 等。)
master_exit: 以 misc_deregister 移除 minor number 的使用。

vm_fault: 轉換 kernel 的 logical address 並將其_count 加一以免被內核釋放掉。
master_close: 用 kfree 釋放記憶體以免造成 memory leak
master_open: 就是分配個記憶體
master_ioctl: 根據參數 icotl_num 的不同，實現創造 socket、連接\傳送數據\結束等功能。
send_msg: 傳送訊息。
master_mmap: 將內核空間的內存映射到用戶空間

Slave_device:
開啟檔案後->建立 socket 接收 slave 的連接->測定起始時間->case f:基本的讀\寫，case m:進行 mmap 映射，根據剩餘空間的大小決定映射的檔案大小->斷開連接->取得時間，得到總時間

Master_device:
取得檔名、作法及 ip 位址->開啟檔案、測定起始時間->與 master 建立連接->case f:基本的讀\寫，case m:映射 dev_d 檔，並依 ret 值決定要結束\映射 file_d 檔的資料大小->結束連接->取得時間，得到總時間

2.
Result

File 1:
(master)
Transmission time: 0.002700 ms, File size: 4 bytes
(slave)
Transmission time: 0.042200 ms, File size: 4 bytes

3. The comparison the performance between file I/O and memory-mapped I/O, and explain why.

MMAP 主要有 2 種用法，一個是建立匿名映射，可以起到父子進程之間共享內存的作用。另一個是磁盤文件映射進程的虛擬地址空間（這個和交換分區的概念不同，交換分區的概念是基於頁面置換的實現），可以節省一次內存拷貝.mmap 的主要的好處在於，減少一次內存拷貝。在我們平時 VFS 的讀/寫系統調用中，文件內容的拷貝要多經歷內核緩衝區這個階段，所以比 MMAP 多了一次內存拷貝，MMAP 只有用戶空間的內存拷貝（這個階段讀/寫也有）。正是因為減少了從 Linux 的的頁緩存到用戶空間的緩衝區的這一次拷貝，所以 MMAP 大大提高了性能

4.
分工:
Clean 等雜工:b05902123
User program:b05902123(主要),b05902107
Slave 和 master:b05902107(主要),b04902067
Ksocket 與版本確認等:b04902067
Report:b04902067,b05902123,b05902107