OS project2 report

1. Design

This is a Master-Slave framework, we need to make both Master and Slave device support mmap.

- **Synchronus**
  - Master side
    - master_device/master_device.c
      - When notified by user_program, find the data in memory.
      - And then use ksend to send the data to slave device.
    - user_program/master.c
      - Map the file to the memory of user_program.
      - Map the memory of device to *user_program.
      - Use memcpy to copy the file to the mapped memory.
      - Use ioctl to notify that device mapping is finished.`
  - Slave side
    - slave_device/slave_device.c
      - Receive the memory data from master device.
      - Use memcpy to copy the memory data to the buffer.
      - Open a socket and wait for the connection from user_program/slave.c.
    - user_program/slave.c
      - Connect to slave device and get the data.
      - Map the memory of device to user_program.
      - Map the content of data to the output file.
- **Asynchronus**
  - Master side
    - user_program/master.c
      - As synchronus.
    - master_device/master_device.c
      - First thread:
        - When notified by user_program, find the data in memory.
        - Add the data into queue.
      - Second thread:
        - When the queue is not empty, use ksend to send the data to slave device.
  - Slave side:
    - slave_device/slave_device.c
      - First thread:
        - Receive the memory data from master device and add into the queue.
      - Second thread:
        - When the queue is not empty, use memcpy to copy the memory data to the buffer.

            - Open a socket and wait for the connection from user_program/slave.c.
        - user_program/slave.c
            - As synchronus.

2. Result

To make our experiment more explainable, we add another large file in our experiment.

- Synchronus
    - Fcntl I/O

```
ioctl success
Master: Transmission time: 0.033800 ms, File size: 32 bytes
Slave: Transmission time: 0.036700 ms, File size: 32 bytes
ioctl success
Master: Transmission time: 0.048900 ms, File size: 4619 bytes
Slave: Transmission time: 0.043900 ms, File size: 4619 bytes
ioctl success
Master: Transmission time: 0.119300 ms, File size: 77566 bytes
Slave: Transmission time: 0.144000 ms, File size: 77566 bytes
ioctl success
Master: Transmission time: 4.887300 ms, File size: 12022885 bytes
Slave: Transmission time: 7.656600 ms, File size: 12022885 bytes
```

    - Mmap I/O

```
ioctl success
Master: Transmission time: 0.051800 ms, File size: 32 bytes
Slave: Transmission time: 0.059400 ms, File size: 32 bytes
ioctl success
Master: Transmission time: 0.063100 ms, File size: 4619 bytes
Slave: Transmission time: 0.075400 ms, File size: 4619 bytes
ioctl success
Master: Transmission time: 0.092000 ms, File size: 77566 bytes
Slave: Transmission time: 0.132500 ms, File size: 77566 bytes
ioctl success
Master: Transmission time: 0.789400 ms, File size: 12022885 bytes
Slave: Transmission time: 2.260500 ms, File size: 12022885 bytes
```

- Asynchronus
    - Fcntl I/O

```
ioctl success
Master: Transmission time: 0.059800 ms, File size: 32 bytes
Slave: Transmission time: 0.040000 ms, File size: 32 bytes
ioctl success
Master: Transmission time: 0.088800 ms, File size: 4619 bytes
Slave: Transmission time: 0.056600 ms, File size: 4619 bytes
ioctl success
```

```
Master: Transmission time: 0.209800 ms, File size: 77566 bytes
Slave: Transmission time: 0.369100 ms, File size: 77566 bytes
ioctl success
Master: Transmission time: 9.696200 ms, File size: 12022885 bytes
Slave: Transmission time: 15.123000 ms, File size: 12022885 bytes
```

- Mmap I/O

```
ioctl success
Master: Transmission time: 0.068900 ms, File size: 32 bytes
Slave: Transmission time: 0.063700 ms, File size: 32 bytes
ioctl success
Master: Transmission time: 1.172100 ms, File size: 4619 bytes
Slave: Transmission time: 2.171300 ms, File size: 4619 bytes
ioctl success
Master: Transmission time: 0.529400 ms, File size: 77566 bytes
Slave: Transmission time: 2.507900 ms, File size: 77566 bytes
ioctl success
Master: Transmission time: 5.637600 ms, File size: 12022885 bytes
Slave: Transmission time: 7.687300 ms, File size: 12022885 bytes
```

- Page Descriptors

```
[  499.338817] master: 8000000068600267
[  499.339255] slave: 8000000068400225

[  499.344998] master: 8000000068600267
[  499.348158] slave: 8000000068400225

[  499.353967] master: 8000000068400267
[  499.354520] slave: 8000000068600225

[  499.382450] master: 8000000068400267
[  499.424784] slave: 8000000068600225
```

3. Analysis

- **Syncronus**
  - Fcntl I/O is faster when file size is small, since a pagesize is 4k. Mmap I/O will take time to copy the whole page at once.

  - Mmap I/O is faster when file size is large enough, since Mmap I/O only needs to copy the data twice, while Fcntl I/O need to copy the data four times.

- **Asyncronus**
  - Speed comparison is the same as syncronus case.

- The time Mmap I/O taking is roughly half of Fcntl I/O's, which shows the copy times difference between them.
- Asyncronus I/O is slower than syncronus in this case due to the potential busy waiting issue of queue.

4. Member

- B05902086 周　逸：全部的程式部分
- B05902052 劉家維：寫report
- B06501051 陳政瑞：寫report