Project 2 Report (Group 1)

1.The Design

File Transmit from Input file to master device

We map the input file and the device onto the master program by using *mmap()*. We map 4096 bytes of the file at a time until the whole input file has all been read. This can easily be done since we can know the file size in advance.

The master program will copy the mapped input file to the mapped master device by *memcpp()* in order to send the data to input file.

In the master device, we will send the file to the socket every time we copied (by using *memcpy()*) data to master device. We cut them into pieces of 512 bytes then send them into the socket (by calling *ksend()*).

Slave.c mmap

First we ask the kernel to map 4096 bytes(a page size) to the object represented by the file descriptor dev_fd, and we let the kernel choose the address by passing NULL to the argument addr. Then we use a while loop to receive the file until there is no data received.

Every time we receive the data, we call the ioctl function in slave device, and receive the data up to a page size. We ask the kernel to map size of the data to the object represented by the file descriptor file_fd, with the offset, current file size. We then set the file offset to new current file size. We use memcpy to copy the data we received to from device to file address. At last we unmap the file to the memory.

Reference:
https://coherentmusings.wordpress.com/2014/06/10/implementing-mmap-for-transferring-data-from-user-space-to-kernel-space/

2.Results

Transmission time of 4 sample input file shown below:

file1_fcntl:

file1_mmap:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out mmap 127.0.0.1
Transmission time: 0.230000 ms, File size: 32 bytes
```

file2_fcntl:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out fcntl 127.0.0.1
Transmission time: 0.447000 ms, File size: 4619 bytes
```

file2_mmap:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out mmap 127.0.0.1
Transmission time: 0.231000 ms, File size: 4619 bytes
```

file3_fcntl:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out fcntl 127.0.0.1
Transmission time: 0.983000 ms, File size: 77566 bytes
```

file3_mmap:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out mmap 127.0.0.1
Transmission time: 0.942000 ms, File size: 77566 bytes
```

file4_fcntl:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out fcntl 127.0.0.1
Transmission time: 78.300000 ms, File size: 12022885 bytes
```

file4_mmap:

```
wj@wj-Inspiron-3542 ~/Desktop/OS/OS/re/user_program $ sudo ./slave out mmap 127.0.0.1
Transmission time: 105.854000 ms, File size: 12022885 bytes
```

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

Memory-mapped I/O is faster than file I/O. Memory-mapped I/O may require more coping operation than memory-mapped I/O. Memory mapping can also simplify the operation by treating the file content as accessible, and do operations on it as a regular buffer in a process.

Somtimes mmap may take up more time than fcntl, like our result of input file4. There may be some reasons for this unexpected performance, the page faults(TLB miss) costs more than copying a page, mmap and munmap may be another reason that causes this result.

4.The contribution of each member

董文捷 B03902062: Code refactoring, slave_device.c
邵楚荏 B03902090: master.c
謝致有 B03902044: master_device.c
吳元魁 B02901137: slave.c