

Kevin Tang
CSCE 313-511

PA2: A client process speaking to a server process

Executing:

```
make  
./client
```

Note that `./client` will run the `./dataserver` forked as a child process. It will run tests 1-4. Tests 1 and 2 are hard coded to get `1.csv`. Test 3 (requesting a binary file) requires you to specify the binary file input in the `BIMDC` folder in the `test3` function.

Design:

The way the client and server communicate is through FIFO channels. These channels can be created by requesting on the main channel for a new channel name, communication is done on a first in first out basis.

Timing:

For `1.csv`, requesting data points took 82.04 seconds, file request too 0.044 seconds, and binary 0.058 seconds.

```
TEST 1: saved BIMDC/1.csv data request into x1.csv  
TEST1 (data request) took: 81848956 musec  
TEST 2: saved BIMDC/1.csv file request into y1.csv  
TEST2 (file request) took: 43945 musec  
TEST 3: copying BIMDC/mybin to BinCpyOutput  
TEST3 (binary transfer) took: 58222 musec
```

For a 5GB binary file, it took 142.64 seconds. The bottleneck for transferring a 5GB file in binary form seems to be the size of request. Each request is only 100 bytes long and there is a maximum size for a request. Increasing the size for request decreases the amount of overhead. Another bottleneck seems to be the process of outputting information to file in binary form versus as `char*`. From the data collected, a binary file request is slower than the standard file request by about 20-30%. We can change the transfer time by doing a standard file request instead of a binary file request.