PA6: The Server Moved Out!   
Due Friday 11/20/20 at 11:59pm, Total 100 pts

**Introduction**

In this PA, you are going to add a IPC class called NetworkRequestChannel to provide communication across the network over TCP/IP. Specifically, you allow the client-side end of a request channel to reside on one machine, and the server-side end of the channel on another machine. The communication over a request channel is to be provided by TCP connection(s). Since the communication API (not just the underlying functionality and features) through TCP is different from FIFO, you need to restructure the server.cpp and client.cpp as well.

First, you are to modify the server program from PA5 to handle incoming requests over network request channels instead of FIFO. The server must be able to handle multiple request channels from the same client (i.e., a single host multiple threads) or from multiple processes on different machines. As a matter of fact, this is quite straightforward because the server side of a TCP connection does not care where the connection is being made from. You also must modify the client to send requests over network request channels.

**The Assignment**

You are to write a program (call it client.cpp) that consist of p patient threads, w worker threads, and file transfer using TCP/IP request channels. Then write your server.cpp so that multiple instances of the client program, either from the same or from different client machines, can connect to the server simultaneously. The client program is to be called in the following form for data requests:

**./client -n <#reqs> -w <****# workers> -b <bb size> -p <#patients> -h <host name> -r <port no>**

Or, like the following for file requests:  
**./client -n <#reqs> -w <# workers> -b <bb size> -f <filename> -h <host name> -r <port no>**

The server is to be called in the following form: **./server -r <port no>**

Note that you the server no longer runs using exec() function from the client. Rather, it is run in the terminal in a different machine, or a different terminal in the same machine (i.e., TCP/IP also works as an IPC method). The server executable does not need the host name argument because it runs the service in the current host. It just needs to take the port number from the user that must occupy and reserve from the OS in that machine. The client on the other hand needs to know where the server is running (i.e., the domain name of the host or its IP address) and at which port number.

**What to Hand In**

You are to hand in a .zip file that comprises the following files:

* All necessary code files and the makefile, even the ones that you did not modified from PA5
* Measure the performance of the PA6 under varying n, w, p for data requests and varying test sizes of files - text and binary and compare with those from PA5. Present the results of these comparisons using graphs like previous PAs.
  + How does TCP/IP method differ from FIFO in terms of speed?
  + What is the maximum number of connections you can create without changing the ulimit parameter? Is the number same as in PA5?
  + What happens to the point of diminish return? Does it change from what you saw in PA5?