John F. Lake, Jr.
Computer Networks
Homework #5

**1.)** You can use fork(), select(), or threads. These are all valid ways to handle multiple clients, but threads are the preferred method.

**2.)** A thread is a semi-process with its own stack that executes a certain piece of code. Threads are preferred because they offer much less overhead than a fork() call, and allow easier communication between threads if you are manipulating external data. Threads all access the same memory, and context switching is much faster.

Multiple threads can run at once, performing the same action. Threads are important to network programming because they let you handle multiple clients at once.

**3.)** Mutexes are programming concepts used to prevent data inconsistencies between threads running in parallel. Mutexes allow certain threads to run and alter data, and they let the other threads know when it is safe to continue running.

Mutexes are important in network programming because multiple threads for different clients should not change shared data at the same time. Mutexes prevents any errors from occurring.

**4.)** Select() allows us to manage multiple clients with one process. With select(), you block until something happens with one of the file descriptors, and then you peform some action with it. Select() is very useful in network programming because the server will not need shared memory or synchronization among threads.

**5.)** 4 Types of Servers:

a. Iterative
   - i. Handles one client at a time.
   - ii. Other clients cannot use the server until the current client stops using it.
   - iii. Limits usability.

b. Concurrent- Threads
   - i. Handles multiple clients.
   - ii. Doesn’t require extra processes and reduces overhead.
   - iii. Need to implement mutexes to prevent data from being manipulated in a bad way.

c. Concurrent- fork()
   - i. Handles multiple clients
   - ii. Doesn’t require an implementation to prevent data from being manipulated.
   - iii. Uses multiple processes, so there is more overhead.

d. Hybrid – Select()

   i. Handles multiple clients.
   ii. Doesn’t require shared memory or synchronization to operate.
   iii. Allows server to run off of one process.