**4.25: Modifying the socket-based date time server to be multithreaded. MAKE SURE YOU TURN in executable .java files. Create two versions**

a. **Creating a new thread for each request**
   Please see attached `DateServerThread.java`
b. **Using a Java-based thread-pool 204**
   Please see attached `DateServerPool.java`

**4.19 The program shown in Figure 4.16 uses the Pthreads API. What would be the output from the program at LINE C and LINE P?**

CHILD: value = 5
PARENT: value = 0

**Read and summarize following page:**
http://www.thegeekstuff.com/2013/11/linux-process-and-threads

Anything that occurs on the Linux OS is called a process, a process can access system resources that are provided by the kernel. Linux processes can be thought of as an instance of a running program. A process can be pre-empted, or put on hold, which allows multiple processes to be juggled by the computer. Pre-emption gives the impression that the processes are happening simultaneously because switches occur very quickly.

Processes can also communicate with one another using different communication methods. In Linux the fork() function is used to create a child process, when the child process tries to change the data a copy of the data is allocated for the child. Allowing the changes to be child specific.

Processes have execution flows calls threads, processes which have multiple threads are called multithreaded processes. In Linux the kernel does not differentiate between a process and a thread, threads are instead seen as a special type of process called a lightweight processes (LWP). The main difference between a normal process and a light weight process is that light weight processes share resources such as address space, files, etc…

In Linux to create a process you call the fork() function, later to create a thread or light weight process use the clone() function with appropriate arguments.