Chapter 4

1. Process creation is heavy-weight while thread creation is light-weight.

2. Benefit of multi-thread: Responsiveness, Resource Sharing, Economy, scalability

3. Multi-core/multiprocessor gives challenge to programmers

-challenge include: Dividing activities, balance, data splitting, data dependency and testing and debugging

4. Parallelism implies a system can perform more than one task simultaneously

5. Two types of parallelism: Data parallelism and task parallelism

6. Three thread libraries: POSIX Pthread, Windows threads, Java Threads.

7. Many to one: many user-level threads mapped to single kernel thread (Solaris Green Threads)

One to one: Each user-level thread maps to kernel thread. (Windows, Linux)

8. Two level Model: similar to many to many except that it allows a user thread to be bound to kernel thread (IRIX, HP-UX)

9. Two ways of implementing thread, library entirely in user space, kernel-level library supported by the OS

10. Two types of dispatch queues: serial – blocks removed in FIFO order, concurrent – removed in FIFO order but several may be removed at a time.

11. Signal is handled by one of two signal handles: default, user-defined

12. Two ways to terminating a thread before is has finished: thread to be cancelled is target thread, two approaches: Asynchronous cancellation, deferred cancellation.

Chapter 5

1. Atomic = non-interruptible
2. Counting semaphore – integer value can range over an unrestricted domain
   Binary semaphore – integer value can range only between 0 and 1, same as mutex lock.
3. Two operation: Block, wakeup
4. Deadlock, two or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes.
5. Starvation – indefinite blocking, a process may never be removed from the semaphore queue in which it is suspended. Priority Inversion – Scheduling problem when lower-priority process holds a lock needed by higher-priority process.
6.