

## CS460 Exam 2 Review

### Threads - Chapter 4

#### Threads

how are threads different from a process?

#### Usage Examples/Benefits

#### Models:

Many to one  
one to one  
many to many

#### User level vs Kernel level

usage scenarios: why user level? why kernel level?

expected speed up: CPU bound multi-threaded code user level threads  
expected speed up: CPU bound multi-threaded code kernel level threads

What is a race condition?

How does a mutex work?

How does a semaphore work?

What is the Dining Philosophers problem?

What is the bounded buffer problem?

What is a Monitor?

Define: Mutual Exclusion/Progress/Bounded Waiting.

Define: Deadlock/Starvation/Priority Inversion

Why is Peterson's solution not actually used today?

What is TestAndSet? CompareAndSwap?

CPU Burst/IO Burst

Scheduling goals

FCFS, SJF, RR, Priority

Draw charts

Burst length prediction

Multilevel Queues: why?

Starvation/Aging

Multiprocessor Scheduling  
how is it different  
affinity

Solaris/Linux (O(1) and CFS)

What is a kernel module?

Why might you want one?

What security issues are raised by writing a kernel module?