UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 537
Introduction to Operating Systems

Andrea C. Arpaci-Dusseau
Remzi H. Arpaci-Dusseau

Dining Philosophers & Monitors

Questions answered in this lecture:

- How to synchronize dining philosophers?
- What are monitors and condition variables?
- What are the differences between Hoare and Mesa specifications?

Two Classes of Synchronization Problems

Uniform resource usage with simple scheduling constraints

- No other variables needed to express relationships
- Use one semaphore for every constraint
- Examples: thread join and producer/consumer

Complex patterns of resource usage

- Cannot capture relationships with only semaphores
- Need extra state variables to record information
- Use semaphores such that
  - One is for mutual exclusion around state variables
  - One for each class of waiting

Always try to cast problems into first, easier type

Today: Two examples using second approach

Dining Philosophers

Problem Statement:

- N Philosophers sitting at a round table
- Each philosopher shares a chopstick with neighbor
- Each philosopher must have both chopsticks to eat
- Neighbors can't eat simultaneously
- Philosophers alternate between thinking and eating

Each philosopher/thread i runs following code:

```
while (1) {
   think();
   take_chopsticks(i);
   eat();
   put_chopsticks(i);
}
```