

Threads

CSE 794R/ECE 694R

1

Multiple Threads

- UnicastRemoteObject object is not replicable (one remote object serves all clients): each client executes as a separate thread
- GUIs
- Parallelizable tasks (multiple CPUs?)

CSE 794R/ECE 694R

2

Concurrency

- **Concurrency** is a property of systems which consist of computations that execute overlapped in time, and which may permit the sharing of common resources between those overlapped computations.
- Concurrent use of shared resources is the source of many difficulties. *Race conditions* involving shared resources can result in unpredictable system behavior. The introduction of *mutual exclusion* can prevent race conditions, but can lead to problems such as *deadlock*, and *starvation*.

Race Conditions

- A **race condition** is a flaw in a system or process where the output exhibits unexpected critical dependence on the relative timing of events.

Mutual Exclusion

- **Mutual exclusion** algorithms are used in concurrent programming to avoid the concurrent use of un-shareable resources by pieces of computer code called *critical sections*.
- A **critical section** is a piece of code that can only be executed by one process or thread at a time, e.g., a section of code that accesses a common data structure.

Deadlock & Starvation

- A **deadlock** is a situation wherein two or more competing actions are waiting for the other to finish, and thus neither ever does.
- **Starvation** is a concurrency-related problem, where a process is perpetually denied necessary resources. Without those resources, the program can never finish its task.