CSCI 344/715
Lecturer: Dr. Simina Fluture

Lecture #5

Topics: **Signal and Exit Monitor**

- **Reader – Writer**

Signal-and-Continue Monitor

- **Savage-Cook problem**

Readings: **Web and class lectures**
[SH] ch5 (5.7)

Reader-Writer problem

There are two types of threads (processes) that are accessing a shared database: readers and writers.
The readers query the database, the writers alter the database. Readers may access the database concurrently but the writers require exclusive access.

The Reader-Writer **monitor solution** is a starvation free solution.
A continual stream of arriving readers cannot delay for an arbitrary amount of time a writer from writing.
An arriving reader checks first if there is a waiting Writer. If yes, the arriving reader will wait.
Whenever a writer finishes it sweeps into the database all waiting readers even those that arrived after any waiting writer. The writer gives priority to the waiting readers against the waiting writers.

We will go over the monitor solution and a possible sequence that checks the problem conditions.

CSCI 344/715
Lecturer: Dr. Simina Fluture

Signal and Continue Monitors

Savage-Cook problem
A tribe of savages eats dinner from a large pot that can hold M servings. When a savage wants to eat, he helps himself from the pot unless it is empty. If the pot is empty, the savage wakes-up the cook and waits until the cook has refilled the pot.

Develop code for the actions of the savage and the cook using a signal-and-continue monitor for synchronization.

Fig. SavageCook monitor

thread *pseudo-execution code*:

Condition variables:
The savage waits for ***Servings***.
The cook waits for ***EmptyPot***.

Initialization:

Pseudo – Service Method code: