CS 499 – Distributed Systems  
Monday, October 19th, 2009

TODO:

* <late> Get updated notes from other student

C lass Notes:

Lecture Notes:

* Deadlocks
  + What are they?
    - resources, that cannot be shared
    - process are waiting for other processes to release a resource
    - cycles processors waiting for each other
  + What can you do about it?
    - avoid
    - breaking deadlock cycles
      * abort transactions by looking at the cycles
      * time-outs
        + Are they a good idea?

for heavily accessed items they are accessed a lot so they would look like they are in a dead lock

// Not an ideal method for dealing with deadlocks

V

T

W

C

U

V

V

B

* Coordination and Agreement
  + Mutual Exclusion (ME)
    - Critical Sections (CS) – pieces of code that act on data on resources, and they need to be executed in an atomic way. If they cannot be executed in this way you will have corrupted data.
      * Actions need to be executed in a atomic way; those actions do not need to be atomic.
      * There is no concurrency when dealing with critical sections
    - Conditions:
      * ME 1 (safety): only one process is allowed to execute in the CS at a given time
      * ME 2 (liveness): all processes should have a fair chance to execute CS
        + Examples: “nice” in Linux, priority in task manager in Windows.
  + Central Server Algorithm:

4

2

p1

1. Request token

Queue of requests

2. Release token

3. Grant token

* + Ring Algorithm

token