CSE 535 Asynchronous Systems
Lecture 1 – 30th Aug 2013
Instructor: Annie Liu
Notes by: Madhurima Roy

General Notes:

- No textbooks.
- Class notes to be followed.
- Three questions to be asked: what, why and how

Definition of Concurrent systems

- Asynchronous is one symptom of concurrent and distributed systems.
- Concurrent is
  > Multiple processing elements
  > Multiple storage elements
- Communication is Important. Common things like shared memory are to be communicated. Then problem is synchronization.

Kinds of Concurrent Application:

- Physical terms
  > Corporate/enterprise, Air traffic control systems, banking systems

Why and how

- Better quality of life
- Ease of modeling
- Parallelization makes it fast
- Applications are always measured in terms of
  1. Correctness
  2. Efficiency.

Examples of concurrency:

- Mutual exclusion
- Leader election
- 100 Prisoners and light bulb problem
- coordinated attack
- muddy children

Challenges:

1. Parallel Systems:
   - Like in banks, searching locations, etc.
   - processes are <u>independent</u>
   - performance needs to be rechecked
2. Concurrent:
   - they are <u>not independent</u>
   - We have to apply atomicity.

3. Distributed:
    - only way of communication is message passing.
    - The biggest challenge - when message is sent, the link fails/process fails/slow.
    - This is collaborative work where multi systems are working towards one goal.
4. Multiple player games:
    - competitive

Process communication requirement

- Data: to compute
- Control : is for synchronization

Types of Communication paradigm/model

- Shared memory : is concurrent
- Message passing : is distributed
    - In Distributed shared memory DSM, message passing concurrency is achieved using **Erlang**.
    - It needs interleaving.

Fundamental issues

1. Order of the events
    - Logical clocks: Say there is no physical clock, one msg received and then next is sent. Here order needs to be maintained for mutual exclusion. This is logical clock application.
    - !Clock synchronization!

2. Global state
    i. In distributed system, (snapshot algo,) how to know the global state?
    ii. Global predicate

3. Failure model and fault taularance
    i. process failure / crash failure – after failure, it is assumed that we stop. Issue can be when alive.
    ii. link failure – channel failure. Like in the coordinated attack problem , the messenger dies.
    iii. byzantine failure – any arbitarary failure.

    Fault tolerance
    - First thing to do is Fault isolation
    - Checkpointing is to be set on whole state and when reached, message logging
    - Redundancy, replication
    - Deadlock detection, failure detection

4. Security - Access control

Tools

1) Models and languages.
    a) It can be through shared memory(SM) or message passing(MP)
    SM : impertative/ procedure programming. Languages example tuple space, object space, java space
    MP: Functional programming. Eg Process algebra, Petri net, IO automata, TLA+ -> PLUSCAL

2) Implementation, simulation
    a) Message passing with shared memory: send msg to shared memory

Correctness conditions for Asynch. System

- safety(no 2 are in at the same time)
- liveness(somebody is in at a time)
- fairness

Complexity/performance for Dist. system

- message complexity/communication complexity
- round complexity is when it is to be broadcasted or singularly sent

For submisssion:

Describe a problem and submit