CS 499 – Distributed Systems  
Wednesday, September 02, 2009

Presentation:

Joe Flieger gave a quick presentation of his design.  
 Provides for chat rooms and connection to a larger network

Lecture Notes:

* in depth discussion on communication primitives
* message primitives could be seen from two different perspectives
  + blocking
  + non-blocking
  + synchronous
  + asynchronous
* First picture
  + sender / receiver
  + four scenarios
  + blocking – def – do you wait till sending complete to terminate connection
    - else you initially send then move on to a new process
  + Synchronous – def- waiting for the receiver to send back a signal that they got the message
    - else you continue to different process without waiting for confirmation
  + // Since most of the class is behind in course involvement we will postpone this lecture
* Challenges: System Perspective
  + API for communications
    - “ease of use”
  + Synchronization
    - how do you sync a world that does not have a singular world clock?
  + Data storage and Access
    - where do you host it
    - how do you make it available
    - how do you maintain it
  + Consistency and Replication
    - how do you org replication
    - how to keep replica consistent
      * do you need to keep it consistent
    - difference between a strict and loose system
  + Fault Tolerance
    - how are you going to organize fault tolerance
    - how do you organization fault tolerance in reality of nodes failing and links failing
  + Scalability and Modularity
    - how do you organize your system so it is scalable easily
    - is the user aware of shortages or not
    - higher load 🡪 add more resources to maintain the same
  + Naming
    - how do you name resources so they are readable, discoverable, searchable
  + Processes
    - code migration
    - mobile agents
    - software that moves from machine to machine
    - A process is a container that …
  + There will be more challenges that will either be discussed later or are there for your own independent research
* A Model for Distributed Computations
  + “state”
  + What is a distributed program?
    - Def: composed of a set of an asynchronous processes (p1, p2, …, pN)
  + Process execution and message transfers are asynchronous
    - notation:
      * Cij 🡪 channel from Pi to Pj
      * Nij 🡪 message from Pi to Pj
    - message transmissions delay are finite but unpredictable
* A Model of Distributed Execution
  + “distributed program”
  + a processes can consist of actions (in the end)
    - “a process can just execute some actions”
    - processes in the classical UNIX sense of the work
  + Actions are atomic and indivisible
    - “atomic and indivisible” 🡪 bank transactions
    - “atomic and divisible” 🡪 database management system
    - modeled as three types of events
      * interval events
      * sent message events
      * receive message events

Side Notes:

* Majority of students have never had CS 460 before
* the Friday after next, students that know about sockets should not come to this class
* then after that we will discuss more about socket programming