

# Concurrent & Distributed Software Systems

**CS 475 Spring 2003**  
**Prof. Sanjeev Setia**

## About this Class

- ⌘ Distributed systems are ubiquitous
- ⌘ Focus: designing and writing moderate-sized concurrent & distributed applications
- ⌘ Prerequisites:
  - ☒ CS 471 (Operating Systems)
  - ☒ High level of competence in C/C++/Java

## What you will learn

"I hear and I forget, I see and I remember, I do and I understand" – Chinese proverb

- ⌘ Issues that arise in the development of distributed software
- ⌘ Foundations of concurrent and distributed software
- ⌘ Concurrent Programming
  - ☒ Threads, semaphores, condition variables...
- ⌘ Middleware technology
  - ☒ Sockets, RPC/RMI
  - ☒ Web services

## Logistics

- ⌘ Grade: 65% projects, 35% exams (15% midterm, 20% final)
  - ☒ midterm in March
  - ☒ final probably a take-home exam
- ⌘ Slides, assignments, reading material on class web page  
<http://www.cs.gmu.edu/~setia/cs475/>

## Logistics cont'd

- ⌘ 3 or 4 small (2 week) programming assignments + 1 larger project
  - ☑ concurrent programming, sockets, RPC/RMI
  - ☑ project can be done in a group of two whereas assignments have to be done individually
- ⌘ Use any UNIX platform; all the necessary software will be available on the Sun workstations in the IT&E lab
  - ☑ DO NOT USE OSF1/MASON2
  - ☑ You can use Windows platforms subject to my approval
- ⌘ Strict enforcement of honor code

CS 475 - Spring 2003

5

## Schedule

- ⌘ Concurrent/Multi-threaded Programming
- ⌘ Distributed Systems – Overview
- ⌘ Application-level network protocols
  - ☑ network programming using sockets
- ⌘ Client-server application design
- ⌘ Middleware technology
  - ☑ RPC/RMI/CORBA
  - ☑ Web Services aka using SOAP/XML for distributed applications
- ⌘ Advanced Topics (depends upon time available)
  - ☑ Parallel programming, P2P systems

CS 475 - Spring 2003

6

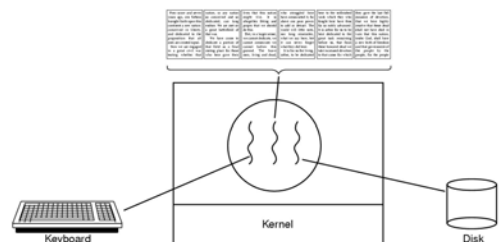
## Concurrent applications

- ⌘ Multi-threaded Programs
  - ☑ Processes/Threads on same computer
  - ☑ Window systems, Operating systems
- ⌘ Distributed applications
  - ☑ Processes/Threads on separate computers
  - ☑ File servers, Web servers

CS 475 - Spring 2003

7

## Multi-threaded application



A word processor with three threads

CS 475 - Spring 2003

8

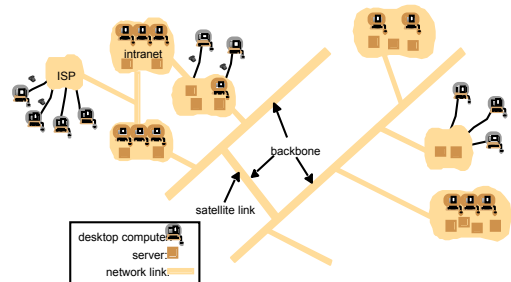
## Distributed systems

- ⌘ "Workgroups"
- ⌘ ATM (bank) machines
- ⌘ WWW
- ⌘ Computing landscape will soon consist of ubiquitous network-connected devices
- ☑ "The network is the computer"

CS 475 - Spring 2003

9

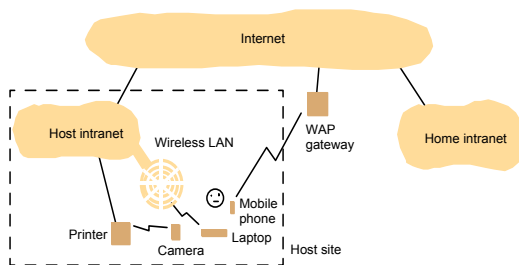
## A typical portion of the Internet



CS 475 - Spring 2003

10

## Portable and handheld devices in a distributed system



CS 475 - Spring 2003

11

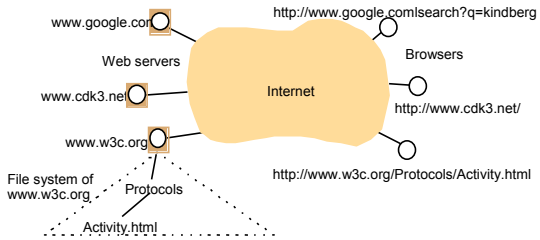
## Distributed applications

- ⌘ Applications that consist of a set of processes that are distributed across a network of machines and work together as an ensemble to solve a common problem
- ⌘ In the past, mostly "client-server"
  - ☑ Resource management centralized at the server
- ⌘ "Peer to Peer" computing represents a movement towards more "truly" distributed applications

CS 475 - Spring 2003

12

## Web servers and web browsers



CS 475 - Spring 2003

13

## Benefits

- ⌘ Resource sharing
- ⌘ Fault tolerance and availability
- ⌘ Performance
  - ☑ Parallel computing can be considered a subset of distributed computing
- ⌘ Scalability

CS 475 - Spring 2003

14