

# CMSC 327: Distributed Systems

## Semester Project

### Overview

The semester project is an opportunity for you to build a distributed system for some interesting project of your choice. We have discussed various reasons for building systems in a distributed manner. Your application should stress one of these themes. For example, your system might:

- provide extended access over some large geographic area, for example, by providing access to remote resources or enabling communication;
- speed-up the computation via parallel processing;
- provide fault tolerance through redundancy.

You are free to choose whatever language and middleware technology that make the most sense for your project. There are few requirements:

- The project should be substantial (it is worth 15% of your entire grade).
- Each team should be composed of 3 or 4 people.
- Group work should be well-decomposed – each member should take ownership of some aspect.
- Written reports should address the following aspects of your system:
  - Design
  - Implementation
  - Evaluation

### Deadlines and Deliverables

October 25	Project Idea and Team	What? Who?	problem, proposed solution team members
November 5	Development Plan	What? Who? When?	problem, proposed solution and method of evaluation team member responsibilities a development and testing timeline
November 26	Status Report	What? How?	accomplishments; remaining work design and initial implementation
December 16	Final Project Report and Demonstration	What? Why? Who? How?	problem and proposed solution motivate problem and solution team member contributions design, implementation, evaluation

### Learning Objectives

- Design, implement and evaluate a distributed system.
- Work as a group to build a substantial piece of software.

## Project Ideas

### Implementation-Oriented

- Create a file-sharing service.
- Create a distributed hashtable.
- Create an instant messaging service.
- Create a shared whiteboard program.
- Implement a distributed simulation system.
- Create a massively multi-player online game.
- Create a n-tier web application (where  $n \geq 3$ ).
- Parallelize some computationally demanding task.
- Create a delay-tolerant networking text-message or micro-blogging system.

### Evaluation-Oriented

- Compare different group communication algorithms.
- Compare various distributed mutual exclusion algorithms.
- Compare different delay-tolerant networking algorithms in simulation.
- Empirically evaluate the performance of various middleware technologies (e.g. TCP/IP, CORBA, JavaRMI)

### Keith-Oriented

- Create a remote interface to some interesting hardware (e.g. a webcam, arduino board, a robot)
  - Collaborative control of the webcam is particularly interesting.
- Create a program to simultaneously grab webcam data from the machines in 100 to build image panoramas.
- Create an android application to control the IPRE robots via bluetooth.