Distributed Computing?

According to Wikipedia:

- Distributed Computing is a field of computer science that studies distributed systems.

Distributed System

- A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal.

A better definition*:

Also,

Specificially:

- A distributed computing application consists of multiple programs running on multiple machines that coordinate together to perform a single task.

- The computers in a distributed system are independent and do not physically share memory or processors.

- They communicate with each other using messages, pieces of information transferred from one computer to another over a network.

* From John DeNero – University of California, Berkeley
Class Exercise
Candy Count

Get ready to stand up, and move around!

Let’s take a closer look

- How did you assign the work to each of you?
- What if we have more work units that you (machines)?
- How did everyone know what to do?
- How did you know that one of you is done?
- How did you aggregate the data at the end?
- What if one of you made a mistake counting?
- What is one of you fell asleep?
- Who made all of these decisions?

What will we learn in this course?

- The components needed in a distributed system to coordinate the distributed resources to perform a task.

- These components include:
  - Network architectures
  - Message passing
  - Distributed algorithms
  - Fault-tolerant mechanisms
  - Distributed file systems

How will we learn?

- Before the lecture:
  - We’ll study existing systems

- In the lecture:
  - We’ll analyze how they work in real life
    - Architecture
    - Algorithms

- After the lecture:
  - You will build some components

- **By the end of the course**
  - We’ll coordinate our own mini-computation
Before the lecture

– You will read about existing systems.
  – Or related algorithms.
  – Or current challenges.

– It depends on the lecture topic.

– The reading is required, because all class interactions will depend on it.

During the lecture

– You will forms groups of 4, and each of you will have a role:
  – Manager: The person in charge of managing the group.
  – Reflector: The person in charge of analyzing the group dynamics.
  – Recorder: The person in charge of taking notes.
  – Speaker: The person in charge of speaking for the group.

– You must switch roles every week.
– You can change groups (if you’d like) every 4 weeks.

– More details are in the sheets in front of you!

After the lecture

– There will be at least 4 programming assignments throughout the semester.
  – Work in pairs.
  – And at least 2 written assignments.
  – Work individually.

– Starting from the Assignment 2, we’ll work on real virtual resources 😊
– All assignment management will be done through our GitHub repository.

– Don’t forget to check out Assignment 0!
  – Due this Friday.
  – It’s simply to set up your accounts on GitHub.

Grading policy

– Assignments: 40%
– Final Project: 20%
– Exam 1 (in-class): 15%
– Exam 2 (take-home): 15%
– Class Participation: 10%
Final project

- After the second exam, you’ll start working on the final project.
- Projects are done in pairs.
- It is a compilation of all the components that you have built in your assignments, to build a complete system.
- You can also pick your own project idea if you’d prefer.

What’s a process?

- It’s a piece of code executed on the machine.
- The operating systems keeps track of the instructions to be executed and the data to be manipulated by the process.
- How many processes do you think your processor can handle at a time?
Process states

Scheduling

- Pre-emptive vs non-preemptive

- Metrics:
  - Efficiency
  - Fairness
  - Turn-around time
  - Response time
  - Throughput

Group Exercise
Scheduling policies

Follow the instructions in the exercise sheet!