Mario Ponte-Garofalo

Week 08/25 - 08/29 Report

This week, we went over the general requirements for this class as well as the basics of DOS. The most important detail discussed in the first lecture is that we will be using Gleam for our projects in this class, which is a relatively new type-safe language that none of us have used before. There are limited examples and information available for this language, so we will be forced to learn it on our own through this course and develop important skills that will be necessary in the field, namely staying up to date with new languages and systems in the quickly expanding field of computer science. We will use this language to eventually create something with most of the functionality of apps like Twitter, Facebook, Reddit, etc.

We also went over how DOS applies to a lot of networking as well as communication within modern processors, especially in terms of parallelism and multithreading. Programmers s in the modern day must be able to make use of multiple independent cores since computers and servers have many of these to increase throughput and their applications are becoming more prevalent with the use of AI. However, fully distributed systems are not always the solution to every problem, so it is important to find the correct solution to the presented problem. Distributed systems are meant to be the most scalable option for communication between multiple cores and systems when compared to centralized and decentralized systems. DOS also have a lot of complex issues, with two of the biggest ones being consistency and security, keeping data the same across systems while also ensuring only authorized access to this data.

On Friday we also went over how we are not truly working on DOS, more so higher-level middleware while still being aware of operating system features. To make efficient systems, using the right middleware is extremely important and can make the difference between a fast, efficient program, and something that is completely useless for its intended purpose. This also shows why OOP is not always the solution, and for us in this class the actor model will most likely be the most important takeaway.