# Class projects Details

ECS 140a:

project 1: learned programming in go(similar to c++). Learned to create test files in order to test code

project2: Coding in lisp – the lack of loops causes the coder to only program each function with recursion

project3: coding in prolog – No loops, functions, and assignments. All we do is define a set of rules and have the computer output results of true or false given the input.

Project4: concurrency in go – The main concept is to deal with data races: Dealing with shared variables between different threads since we are unable to predict the order in which the processor executes these threads.

GitHub - Tarun-coding/ECS140A

ECS 150a:

Emulating the performance of the slotted aloha networking protocol by comparing different retransmission algorithms. In particular, we compare the performance of the slotted aloha with the p-persistent, Binary Exponential Backoff, and Linear Backoff retransmission algorithms.

Working methodology:

There are N hosts(nodes) and 1 server. Server wakes up once during in periodic intervals(we assume 1 second). Every time it wakes up, it checks for the number of nodes active during the particular slot. If there is only one node active, it will transmit the packet of that node and will be considered a successful slot. If there are more than 1 nodes active a collision will occur and the server will retransmit the packet of each of the active nodes to a different time slot decided by the retransmission algorithm. The successful/total slot ratio is what provides us the throughput.

Tarun-coding/SlottedAlohaSimulator (github.com)

ECS 171: machine learning

This group project was essentially trying to measure trends in crime, incarcerations and poverty rates. As part of the software engineering group, our job was to take our models and make them more interactive with the user. The voila feature of the jupyter notebook allowed us to create a webapp where we were able to generate linear regressions, box plots, k-means clusters etc with the input variables of states and the years provided by the user.

nahcn/ECS171-Project (github.com)

ECS 170: Artificial intelligence

Project 1: finding shortest path in 3D grid with the A* algorithm. Developed different heuristics against various cost functions to minimize the number of nodes expanded and the time to find the optimal path from the start to the goal state.

Project 2: Using the minimax algorithm to play connect 4: We looked a specified depth down the search tree(approximately 11 moves from both players) to find the path that promises us the highest utility. We also implemented alphabeta pruning optimization to save time by not growing certain parts of the tree.

Tarun-coding/ECS170 (github.com)

ECS 150-operating systems

Construction of shell: The custom shell we had build would be able to handle built in commands (exit,pwd etc.), regular commands, output redirection, piping(upto 3 pipes). For more information on the different steps in construction of the shell read the Report.md for "P1" in the link.

Thread management: In this project we deal with thread scheduling(with the help of queue datastructure) and management. Depending on what the user requires, the thread will be in the ready queue(where each thread will be executed in order), the blocked queue(essentially waiting for another thread to be finished before continuing execution), zombie queue(where thread has finished execution but is waiting to be collected by another thread). The makefile and the test files are in the "apps" folder of "P2". Report.md file in "P2" contains more detailed information.

File system: Implemented the basic Microsoft FAT(file allocation table) file system.(will come back to this later, since I can't remember how I did this).

Tarun-coding/ECS150 (github.com)

ECS 154A-computer architecture

Constructed a model CPU using logism(logical circuits): (come back to this later)

You will need to download logism to execute the .circ files

Tarun-coding/ECS-154A (github.com)

Classes to still write about:

ECS 132-probability and stats for ECS:

ECS 122A-algorithm design and analysis

ECS 120-theory computation