Instructor Information:

Instructor: Dr. Nihat Altiparmak
Office: DC 209
Phone: 502-852-7533
E-mail: nihat.altiparmak@louisville.edu
Office Hours: Tue & Thu 1:00 PM - 2:00 PM in-person (DC 209), or by appointment (in-person or online through MS Teams).

TA Information:

TA: Jacob Higdon
Office: DC 242
Phone: 502-852-0462
E-mail: jacob.higdon@louisville.edu
Office Hours: Mon & Wed Noon - 1:00 PM in-person (DC 242), or by appointment (in-person or online through MS Teams).

Course Information:

Number: CSE-629-75-4252
Title: Distributed System Design
Format: In-Person
Meeting Times: Tue & Thu 11:00 AM - 12:15 PM
Meeting Location: WS Speed 113
Credit Hours: 3
Class Website: http://cecs.louisville.edu/nihat/teaching/cse629s25

Technology and Logistics Requirements:

- A working computer equipped with a webcam, speakers, and microphone.
- UofL's version of the Respondus Lockdown Browser software installed on the computer. Usage and installation instructions are provided in BlackBoard.
- A decent Internet connection.
- A desk located in a private room, where you can be alone by yourself and close the door.

Use of Generative AI:

- The use of Generative AI is not permitted in this course for any assignment.

Textbooks:

- Distributed Systems, 4th Edition by Maarten Van Steen and Andrew Tanenbaum. ISBN-13: 9789081540636. (Required - Free in PDF form)
  - Request Free Copy: https://www.distributed-systems.net/index.php/books/ds4/
- Distributed and Cloud Computing, 1st Edition by Kai Hwang; Geoffrey C. Fox; Jack Dongarra. ISBN-13: 9780123858801. (Recommended)
- The Go Programming Language, 1st Edition by Alan Donovan and Brian Kernighan. ISBN-13: 9780134190440. (Recommended)
- Computer Networking: A Top-Down Approach, 8th Edition by James Kurose and Keith Ross. ISBN-13: 9780135928615. (Recommended)

Prerequisites:

CSE 420: Design of Operating Systems or equivalent

Course Description and Topics Covered:

This course covers general concepts in design and implementation of distributed systems, by visiting foundational topics in communication, synchronization, consistency, replication, and fault tolerance. Discussions of research papers will reveal recent advancements in the field and design of modern dis-

tributed systems currently used in large-scale companies such as Google, Facebook, Amazon, and Microsoft. The GO programming language and the AWS cloud computing platform will be used in projects. This course is composed of the following components:

- **Introduction:** GO and AWS review, review of necessary computer networks background for distributed systems, introduction to distributed systems, goals, and architectures.
- **Processes:** Review of processes and threads in an operating system, roles of processes and threads in a distributed system design, virtualization, and code migration.
- **Communication:** Socket programming, Remote Procedure Call (RPC), and message passing based distributed communication techniques.
- **Synchronization:** Clock synchronization, logical lamport and vector clocks, mutual exclusion techniques in a distributed system including centralized, decentralized, distributed, and token-based solutions, and election algorithms.
- **Consistency & Replication:** Consistent ordering, consistency models including eventual, monotonic read/write, read your writes, and writes follows reads models, replica placement, and consistency protocols including the primary-backup protocol.
- **Fault Tolerance:** Failure models, paxos, distributed commit techniques including two-phase and three-phase commit, CAP theorem, and recovery from failures via checkpointing and message logging.
- **Distributed Computing and Storage:** Google File System and MapReduce programming paradigm.
- **Paper Readings/Discussions:** Selected among Google's BigTable, Megastore, Chubby, Spanner, Borg; Facebook's photo storage/caching, existential consistency; Amazon's Dynamo; Microsoft's Dyrad, and others including but not limited to Bitcoin, Spark, Ceph, and The Tail at Scale.

Course Schedule:

Please see the class website for a detailed schedule of the course:

http://cecs.louisville.edu/nihat/teaching/cse629s25

Student Learning Objectives:

- Gain familiarity with the networking fundamentals necessary to understand and develop distributed systems.
- Learn distributed system design goals and system architectures including centralized, decentralized, and hybrid forms.
- Understand the roles of processes and threads in distributed systems, the concept of virtualization, and process migration.
- Demonstrate understanding of different communication models in distributed systems including remote procedure calls, massage passing, and various multicasting techniques such as flooding and gossiping.
- Understand how processes can synchronize and coordinate in a distributed system using relative ordering, mutual exclusion, and election algorithms.
- Demonstrate understanding of consistency and replication models in distributed systems including data-centric and client-centric consistency models, replica placement and management, as well as consistency protocols including the primary-backup technique.
- Learn fault tolerance terminology, the CAP theorem, and develop understanding of techniques for making distributed systems fault tolerant including process groups, paxos, distributed commit protocols, check pointing, and message logging.
- Demonstrate understanding of modern distributed computing models (MapReduce) and distributed storage techniques (GFS).

- Read, understand, and evaluate influential research papers on distributed systems including but not limited to Google's MapReduce, GFS, and Spanner, Amazon's Dynamo, and Facebook's Photo Storage.
- Develop hands-on experience developing distributed systems using the GO programming language and the AWS cloud platform.

Grading:

- 30% Test
- 30% Projects (3 programming projects, 10% each)
- 15% Paper Presentation
- 15% Paper Reaction Write-ups (one review for each paper discussed, equally weighted)
- 10% Homeworks (8 HWs, equally weighted)

The letter grade is calculated as follows: $100 \geq$ A+ $\geq 97 >$ A $\geq 94 >$ A- $\geq 90 >$ B+ $\geq 87 >$ B $\geq 84 >$ B- $\geq 80 >$ C+ $\geq 77 >$ C $\geq 74 >$ C- $\geq 70 >$ D+ $\geq 67 >$ D $\geq 64 >$ D- $\geq 60 >$ F ≥ 0.

Announcements:

All announcements will be posted on BlackBoard and will also be immediately emailed to you.

MS Teams:

This course will be utilizing MS Teams to facilitate class discussion. Rather than emailing questions to the course staff (instructor/TA), please post your questions on MS Teams. The course staff will monitor MS Teams closely and you will usually get a quick response. If you know the answer to a question, you are encouraged to help your classmates by replying to their posts, which will improve your virtual class participation and it is highly recommended! **You should ask your questions directly to the course staff only if your question might reveal part of your solution to an assignment.** MS Teams is the most effective way to communicate with the course staff. Please avoid email if MS Teams will do.

Attendance:

Attendance will not be taken in this class.

Projects:

Projects will be mostly programming based and will be assigned and submitted through Blackboard as scheduled in the class website. All programming will be performed **in AWS using the GO programming language only** (not C, C++, Java, Python, etc.).

Academic Integrity and Plagiarism:

1. This course assumes that assignments submitted by students will be generated by the students themselves, working individually or in groups as directed by class assignment instructions. This policy indicates the following constitute violations of academic honesty: a student has another person/entity do the work of any portion of a graded assignment for them, which includes purchasing work from a company, hiring a person or company to complete an assignment or exam, **and/or using generative AI tools (such as ChatGPT).**
2. All submitted assignments should be done individually unless explicitly stated as a group assignment. **Except your group members (if a group project), you are not allowed to go over your friends' code, and your friends cannot see your code.** You are only allowed to make high-level verbal discussions on the projects with other students to make sure what is being asked for. Please note that high-level discussions do not translate into specific algorithms/code implemented in your assignments - such discussions would be considered low-level. Such low-level discussions specific to the implementation of the assignment can only be made with the course staff, and those discussions cannot be shared with other students!

3. Posting assignments and/or solutions online is not permitted. You cannot publish your code partially (in forums or other sites for asking questions) or completely (in public source code repositories). **For instance, you cannot post your code on your public GitHub account unless you make it private!**
4. **You are not allowed to share your code with the future students of this class.**
5. **You are not allowed to use Chegg.com, CourseHero.com, MyAssignmentHelp.com, or any other sites to post any course material, including syllabus, homeworks, exams, projects, slides, etc., and/or to find solutions to assignments.** Instead, post your questions on MS Teams if they do not reveal your answer, or email them to the course staff if your answer might be revealed on MS Teams. In addition, the course staff has dedicated office hours, and are also available via appointment for one-on-one help. Seek immediate help through these means when you need help!

Not complying with these collaboration and code sharing rules will put you under the risk of plagiarism for this semester or the following semesters, including the cases where future students of this class copying your code from online resources (forums, GitHub, etc.). **WE USE ADVANCED COPY CHECKERS! Cheating/copying of assignments (including Internet resources) will be reported to the dean's office for plagiarism and a grade of F will be recorded for the entire course. NO EXCEPTIONS WILL BE MADE!!!** The following procedure will be followed to deal with potential plagiarism cases:

https://engineering.louisville.edu/academicdishonesty/

To clarify even further, here we provide some examples of plagiarism:

- My friend promised to only check, not submit my code, so I emailed him/her my solution.
- My friend and I sat side-by-side and did the project together.
- I did not see my friend's code and s/he did not see my code, but I gave line-by-line instruction on how to solve part of the project. (*Clarification: You can only have high-level discussions of the assignment to make sure what is being asked for. High-level discussions do not translate into specific algorithms or code. Any low-level discussions specific to the implementation of the project can only be made with the course staff, and those discussions cannot be shared with other students!*)
- My friend asked me to debug his code so I went through his code and helped him to debug before/after the deadline. (*Clarification: Only the course staff can go through your code for debugging help; however, you can post the errors you get on MS Teams without revealing your source code/solution!*)
- I shared my solution with a friend only **after** the deadline. (*Clarification: Sharing your code after the deadline is still plagiarism.*)
- The project asked me to implement my version of X, I googled it, went through specific implementations, and copied some code partially/completely in my project. (*Clarification: You can check Internet/textbooks for textual descriptions of algorithms, if applicable, but you cannot check their implementations - pseudo or actual code - and/or copy any code partially or completely from other sources - even though you change some variable names or move some code around!*)
- I googled project description (fully/partially), found some code online, and used it in my solution.
- I asked ChatGPT to solve the project (fully/partially) and used code it generated in my solution.
- I shared my code publicly on GitHub (or other sites) after the course ended. (*Clarification: Sharing your code after the course ended deadline is still plagiarism.*)

Please note that plagiarism scenarios are not limited to the ones discussed above but there is no need to get stressed about it! Just check with the instructor in advance if you are not sure. A simple MS Teams chat with the instructor would clarify it very easily!

Homeworks:

Homeworks will be assigned and submitted through BlackBoard as scheduled in the class website. You will need to upload a PDF to BlackBoard including your answers. You have multiple options to generate the PDF:

1. You can print the provided HW and write your answers on the printed copy, or write your answers on a blank sheet of paper. Once you are done, you will need to take a picture or scan your completed solution using a scanner or a scanning app (Genius Scan, Scannable, Microsoft Lens, etc.), and upload the generated PDF to BlackBoard.
2. You may type your HW (using Word, LibreOffice, LaTeX, etc.) and save your file as a PDF.
3. You may write your solutions using a tablet device and save as a PDF.

Exam and Makeup Policy:

Exam will be administered online on BlackBoard using Blackboard's Lockdown Browser and Respondus Monitor feature. Online exam date and time is provided in the class website; and a grade of zero will be recorded for missed exam unless prior arrangements are made **with a valid proof of excuse (such as doctor's note)** at least a week in advance, in which case your exam will be administered at an earlier time than the scheduled time for the rest of the class, and it will include a different set of questions. If you have not used Blackboard's Lockdown Browser and Respondus Monitor feature yet, please make sure to read the provided "Online Test Guide" and "Using LockDown Browser and Monitor for Students" documents in BlackBoard for specific instructions on installing necessary software and taking an online exam using Lockdown Browser and Respondus Monitor. Additionally, you can also watch the following introductory videos explaining how Lockdown Browser and Respondus Monitor work:

- http://youtu.be/e-QRHkoF8Xg
- http://youtu.be/hv2L8Q2NpO4

Please note that to take the exam with LockDown Browser and Respondus Monitor:

1. You will need a working computer equipped with a webcam and microphone, as well as a decent Internet connection.
2. You will be alone in a room where you can close the door.

More details on the exam are as follows:

- Exam will be closed books and notes. You will be responsible for only what is covered in lecture videos and slides. You will not be allowed to use the Internet, calculators, cellphones, or other electronics during the exam.
- You are not allowed to leave your desk during the entire exam; otherwise you will be flagged by the Respondus system. Leaving your desk in the middle of the exam will risk your exam grade and might cause you to get a 0. Also, make sure to show enough in the environment video, where tilting your web cam down and back up will not really show your environment. You should try your best to show your surroundings.
- A practice testing environment will be provided on BlackBoard for you to get used to the online testing using Lockdown Browser and Respondus Monitor. Please test your system with the provided practice before the exam to make sure all works fine.
- Make sure to answer all questions before you exit the system.

Disability Resource Center (DRC) Statement:

The University of Louisville is committed to providing access to programs and services for qualified students with disabilities. If you are a student with a disability and require accommodation to participate and complete requirements for this class, notify me immediately and contact the Disability Resource Center (Stevenson Hall 119, 502-852-6938, askdrc@louisville.edu) for verification of eligibility and determination of specific accommodations.

Computer Issues and IT Support:

Speed IT staff are available by appointment from 9:00 am to 4:00 pm to assist you with your technology needs. You may schedule an appointment by sending a detailed email including any relevant error codes and screen snips at SPDHelp@Louisville.edu (preferred) or 502-852-7620. You can also seek help from the course TA or the CSE IT Staff (2nd floor Duthie) if you need assistance with your Ubuntu installation. Make sure you set up an appointment in advance, preferably early in class before the first project is assigned.

Title IX/Clery Act Notification:

Sexual misconduct (including sexual harassment, sexual assault, and any other nonconsensual behavior of a sexual nature) and sex discrimination violate University policies. Students experiencing such behavior may obtain **confidential** support from the PEACC Program (852-2663), Counseling Center (852-6585), and Campus Health Services (852-6479). To report sexual misconduct or sex discrimination, contact the Dean of Students (852-5787) or University of Louisville Police (852-6111).

Disclosure to University faculty or instructors of sexual misconduct, domestic violence, dating violence, or sex discrimination occurring on campus, in a University-sponsored program, or involving a campus visitor or University student or employee (whether current or former) is **not confidential** under Title IX. Faculty and instructors must forward such reports, including names and circumstances, to the University's Title IX officer.

For more information, see http://louisville.edu/hr/employeerelations/sexual-misconduct-brochure.

The instructor reserves the right to make changes in the syllabus when necessary. Such changes will be announced via BlackBoard.