Syllabus

CSC435: Distributed Systems I

Quarter: Fall 2023

Lecture Time: Tuesday 5:45pm- 9:00pm

Location: CDM202 / Online (Async)

Professor: Alexandru Iulian Orhean (aorhean@depaul.edu)

Office Hours:

- Tuesday 4:30pm-5:30pm (CDM843)
- Friday 3:00-4:00pm (Online D2L/Zoom)

Course Description

This course covers the general principles and the fundamental concepts behind distributed systems. The focus is on issues and concepts that are critical to the application of distributed systems and computer networks, such as: the definition and organization of distributed systems, distributed processing, inter-process communication, process coordination, distributed data consistency and replication and fault tolerance. The course will be mostly devoted to basic concepts and techniques but will also include assorted topics in advanced distributed systems and in distributed programming. CSC435 is a good foundational course for many other courses, such as: CSC464 Virtualization and Cloud Computing, CSC536 Distributed Systems II and CSC551 Distributed Database Systems.

Prerequisites

CSC403 Data Structures II and CSC407 Systems II.

Course Book (REQUIRED)

Marteen van Steen and Andrew Tanenbaum, “Distributed Systems” 4th Edition

https://www.amazon.com/Distributed-Systems-Maarten-van-Steen/dp/9081540637

Course Schedule

Week 1: Introduction to Distributed Systems and the Linux Ecosystem
Week 2: Distributed Systems Architectures
Week 3: Processes, Threads and Virtualization
Week 4: Remote Procedure Call and Remote Method Invocation
Week 5: Message-oriented and Stream-oriented Communication
Week 6: Synchronization and Coordination
Week 7: Consistency and Replication
Week 8: Fault tolerance
Week 9: Fault tolerance
Week 10: Cloud Systems, Network File Systems, Distributed File Systems

Programming Assignments

There will be 4 programming assignments, each worth 10% of the total grade and each taking between 2 and 3 weeks to complete. There will be 2 short extra credit programming assignments, that are also worth 10% each and take only 1 week to complete. The assignments must be completed individually. They will require knowledge of Java and C++, and it is expected that the students already know the basics of these languages. The solutions to the programming assignments must run on a Linux environment, on which they will also be graded.

Computing Resources

To complete the programming assignments, the students will use their personal computers and Chameleon Cloud. Instructions for using Chameleon Cloud will be provided during class.
https://www.chameleoncloud.org/

Exams and Quizzes

There will be 8 quizzes (10-min long), each of them covering the lecture content of the previous week and each of them worth 2.5% of the total grade. The quizzes must be completed individually. In-class students must take the quizzes in class. Online students must take the quiz online in a 1-hour window once the quiz is announced (typically at 7:30pm CST). The quizzes are open note, open book and open Internet, and calculators are allowed.

There will be 1 final exam, that will cover the materials from the entire quarter. The exam must be completed individually, and the students will be allowed to use their textbooks and any notes they have on paper. No electronic phones, eReaders, tablets or laptops are allowed during the final exam. Pocket calculators will be allowed, as long as they do not have a browser (e.g., phones, tablets, laptops). The final exam will be worth 30% of the total grade. In-class students must take the exam in class. Online students can take the exam at an official testing center.

There will be no makeup exams or quizzes.

Late Policy

Programming Assignment will be due at 11:59pm on the day of the due date. There will be a 1-hour grace period. Any late submission beyond the grace period will be penalized 20% every day it is late (an assignment that is more than 5 days late will receive zero credit). Quizzes and exams cannot be taken late.

Grades

Grading Policies:

- **Programming Assignments (4 + 2)** 40% + 20% extra credit
- **Quizzes (8)** 20%
- **Exam (1)** 40%

Plagiarism Policy

DePaul University is a learning community that fosters the pursuit of knowledge and the transmission of ideas within a context that emphasizes a sense of responsibility for oneself, for others and for society at large. Violations of academic integrity, in any of their forms, are, therefore, detrimental to the values of DePaul, to the students' own development as responsible members of society, and to the pursuit of knowledge and the transmission of ideas. Violations include but are not limited to the following categories: cheating; plagiarism; fabrication; falsification or sabotage of research data; destruction or misuse of the university's academic resources; alteration or falsification of academic records; and academic misconduct. Conduct that is punishable under the Academic Integrity Policy could result in additional disciplinary actions by other university officials and possible civil or criminal prosecution. Please refer to your Student Handbook or visit the Academic Integrity website for further details.

In this class, plagiarism includes submitting, as your own work, a computer program that was written by someone else, or directly derived from someone else. A program is directly derived from someone else's program if it is identical to someone else's program except for minor changes such as reformatting, change of variable names, etc. We will use D2L's built-in plagiarism detection tool and MOSS.

Moss Results

Tue Sep 8 23:29:31 PDT 2015

Options -l python -d -m 10

[How to Read the Results | Tips | FAQ | Contact | Submission Scripts | Credits]

File 1	File 2	Lines Matched
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (99%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/kt[redacted]/ (99%)	86
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/k[redacted]/ (76%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (66%)	91
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (81%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (82%)	69
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (70%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]3/ (61%)	70
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (69%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (40%)	71
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/k[redacted]/ (56%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]4/ (50%)	43
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (62%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]9/ (55%)	67
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (55%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (48%)	40
/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/k[redacted]/ (54%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/6/raw/[redacted]/ (55%)	40

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]/ (68%)	/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]/ (73%)
4-71	2-66
95-111	90-106
74-91	69-86
115-132	110-127

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]

```
>>>> file: LongJump.py
#[redacted]a

print("***** Long Jump Information System *****")
print("Please enter the names of competitors. (Press return when done.)")
print("Competitor no. 1:")
competitor = input()
b,c,g,h,d,k = 1,0,0,0,[],0
maxi,competitors = [],[competitor]
while True:
    b += 1
    print("Competitor no. "+str(b)+":")
    competitor = input()
    if competitor == "":break
    else:
        competitors.append(competitor)
print("Please enter the distances for each competitor.")
for each in competitors:
    print("Competitor " each ":" sep="")
    at1 = input("Attempt 1:\n")
    at2 = input("Attempt 2:\n")
    at3 = input("Attempt 3:\n")
    x = (at1+at2+at3).lower()
    if (at1+at2+at3).find("oul") != -1:
        x = (at1+at2+at3).lower()
    d.append(at1)
    d.append(at2)
    d.append(at3)
    if x.find("foul") != -1:
        maxi.append(max(eval(at1),eval(at2),eval(at3)))
```

/home/ubuntu/Projects/work/2015/uct-csc1010h/tutorials/4/raw/[redacted]

```
>>>> file: LongJump.py

print("***** Long Jump Information System *****")
print("Please enter the names of competitors. (Press return when done.)")
print("Competitor no. 1:")
competitor = input()
b,c,g,h,d,k = 1,0,0,0,[],0
maximums,competitors = [],[competitor]
while True:
    b += 1
    print("Competitor no. "+str(b)+":")
    competitor = input()
    if competitor == "":break
    else:
        competitors.append(competitor)
print("Please enter the distances for each competitor.")
for each in competitors:
    print("Competitor " each ":" sep="")
    attempt1 = input("Attempt 1:\n")
    attempt2 = input("Attempt 2:\n")
    attempt3 = input("Attempt 3:\n")
    g = (attempt1+attempt2+attempt3).lower()
    if (attempt1+attempt2+attempt3).find("oul") != -1:
        g = (attempt1+attempt2+attempt3).lower()
    d.append(attempt1)
    d.append(attempt2)
    d.append(attempt3)
    if g.find("foul") != -1:
        maximums.append(max(eval(attempt1),eval(attempt2),eval(attempt3)))
    else:
        d.remove("foul")
    if not "foul" in d:
```