University of Houston-Clear Lake
CSCI 5531 – 01
Advanced Operating Systems
Spring 2015

Quality Enhancement Plan (QEP)
Applied Critical Thinking for Lifelong Learning and Adaptability

Applied Critical Thinking Statement: This course has been authorized by UHCL as an Applied Critical Thinking (ACT) Course which means that in addition to learning about the specified course content, students will be engaged with some or all of the Elements of Thought and Universal Intellectual Standards of critical thinking. The objective of an ACT course is to develop the student's ability to become skilled at analysis and evaluation by applying a set of intellectual tools that may be effectively used across all disciplines (as well as to the student's personal life). Based on the Foundation for Critical Thinking model (http://www.criticalthinking.org/), critical thinking involves thinking for a *purpose*, asking *questions*, using *information*, applying *concepts*, drawing *inferences and conclusions*, identifying *assumptions*, anticipating *implications and consequences*, and recognizing *points of view*. The Universal Intellectual Standards that are applied to these Elements of Thought of critical thinking in order to develop Intellectual Traits include *clarity, accuracy, precision, relevance, depth, breadth, logic, significance,* and *fairness*.

Instructor:	Dr. Ahmed Abukmail
Email:	Abukmail@uhcl.edu
Office:	Delta Building – Room D-169
Phone:	(281)283-3888
Office Hours:	Mon. Wed.: 4:30pm – 6:30pm; and by appointment..
Meeting Time:	Tuesday: 4pm – 6:50pm.
Meeting Room:	D-241.

Teaching Assistant (TA): Neha Gorule **Email**: GoruleN6662@uhcl.edu

TA Office hours:

Mon:,7-10pm Tue: 5-9pm, , Wed: 7-10pm, Thu: 6-10pm , all held in Delta D158

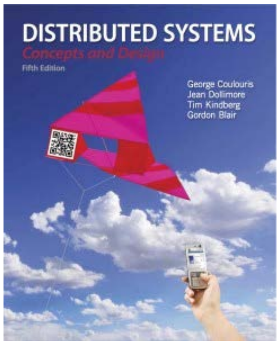
Course Description:

This course is a required course in the Master's program in Computer Science. The course covers the fundamental *concepts relevant* to distributed operating systems. Basic operating systems and single processor *concepts* are prerequisite knowledge that all students are required to *clearly* possess. Students will be exposed to more advanced *concepts and issues* in Operating Systems, especially concepts *related* to multiusers and distributed domains. We will also discuss the different implementation *issues related* to material through programming projects. Topics to be discussed include processes and threads, IPC, RPC, sockets, process synchronization, global clocks, logical clocks, distributed mutual exclusion, communication and synchronization in distributed systems, consistency and replication, security and fault tolerance in distributed systems, distributed file systems, and distributed object-based systems.

Critical thinking and its elements are paramount in this course and in computer science in general. As often a computer scientist is faced with problems that have not been confronted before to which he/she must develop a solution after analyzing the requirements, and initiating the design of the solution to the particular problem. In this course, many of the programming assignments are going to be composed of problems that students have not seen before, but they have knowledge of the tools used to solve them. The students will be taught the *concepts* of synchronization, multiprocessing, and multithreading and will be given *clear* examples in class, however at the time of the assignment, a different programming assignment will be presented that applies the knowledge gained in class. While the constructs learned are applicable to the new assignment, knowing how to apply them to new problems and make the necessary and *accurate assumptions relevant* to the new assignment is paramount in understanding these *concepts* and any *issues relating* to them.

Text:

Distributed Systems Concepts and Design 5th Ed., G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair. Addison Wesley 2012. ISBN-10: 0-13-214301-1, ISBN-13: 978-0-13-214301-1



Prerequisites:
CSCI 4534 Operating Systems [or equivalent undergraduate senior-level OS course]
Familiarity with C/C++ and Java.
Familiarity with Unix and System Calls

Student Learning Objectives (SLOs): The students who successfully complete this course will be able to:

1. Understand and apply the important *concepts clearly* in distributed systems and advanced operating systems.
2. Develop an *in-depth* understanding of the *implications and consequences* designing and implementing a distributed system. *Clearly* identify the *issues related* to how the different components such as: inter-process communication, global clocks and synchronization, consistency and replication, and fault tolerance work together.
3. Develop and *in-depth* understanding of the design and development *issues* of distributed systems components: inter-process communication, global clocks and synchronization, consistency and replication, and fault tolerance.

4. Gain exposure to the main research *issues* as they *relate* to the design of distributed systems. Students will also be prepared for reading, understanding, and participating in research *relevant* to distributed and advanced operating systems.
5. Understand the *significance* of technologies and principles such as Virtualization, Cloud Computing, Pervasive Computing, Distributed and Web-Based Systems as they *relate* to designing and *modeling* distributed systems.
6. Practice and develop applications that *solve* the various *concepts related* to the advanced and distributed operating systems.

Fundamental and Powerful Concepts: This course includes a set of concepts that are essential to understanding advanced and distributed operating systems. These concepts are:

1. Clear understanding of concurrent programming and system calls.
2. The ability to relate the fundamental communication and design concepts to address issues in the construction of distributed systems.

Major Activities:
There will be four major activities within this course all of which are highly relevant to critical thinking and to enhance the students' ability to develop and in-depth clear understanding of course material and achieve the course goals. During the class activities such as programming assignments, quizzes, and examinations, all of the SLOs will be addressed and will be assessed accordingly. Each SLO is going to map to multiple exam questions and/or multiple assignments.

In particular we will assess the following SLOs with respect to critical thinking:

1. **SLO 2:** This is a *significant* SLO during which students will be required to write programs that model synchronization and interprocess communications by *solving* different scenarios presented to them in the problem description. Additionally, *problems* such as global clock, consistency, replication and fault tolerance are going to be covered *in-depth* and evaluated on *related* quiz questions as well as on the examinations.
2. **SLO 4:** A chapter on current *issues* in distributed systems, in particular mobile and ubiquitous computing will *broadly* discussed, during which *relevant* research *issues* in the topic will be addressed. Student's understanding will be measured by answering a specific targeted quiz and exam questions.
3. **SLO 6:** Applying concepts of advanced operating systems such as multithreading to *solve related* concurrency and synchronization mechanisms will be addressed through a programming assignment on which the students will develop *precise solutions.*

Grading:

Homework Quizzes:	20% of the grade.
Programming Assignments:	20% of the grade.
Midterm Examination:	25% of the grade.
Final Examination:	35% of the grade.

Grading Scale:

93+ = A; 90+ = A-; 87+ = B+; 83+ = B; 80+ = B-; 77+ = C+;
73+ = C; 70+ = C-; 67+ = D+; 63+ = D; 60+ = D-; 0+ = F

Assessment Method:
During this semester there will be graded assignments and examinations. While all SLOs will be assessed in this course. The particular SLOs we decided to assess will be the focus of this assessment method. The assessment criteria will be based on the grade of a particular question on the exam/quiz, or a particular programming assignment used to measure the understanding of a particular concept, or a combination of both exam questions and programming assignments. An overall assessment for the course will be deduced by combining the three individual assessments from the critical thinking SLOs. The following will be used for each SLO:

1. **SLO 2:** A combined exam/quiz grade for the particular questions of 80-100% will be considered excellent. A grade of 65-79% will be considered acceptable. Below 65% will be considered unacceptable. The assignment will consist of a variation of the reader/writer problem during which the use of process creation with semaphores used for synchronization.
2. **SLO 4:** A combined exam/quiz grade for the particular question of 85-100% will be considered excellent. A grade of 70-84% will be considered acceptable. Below 70% will be considered unacceptable. A discussion of mobile computing with test questions relating to the topic will be on the final examination.
3. **SLO 6:** An assignment grade of 90-100% will be considered excellent. A grade of 70-84% will be considered acceptable. Below 70% will be considered unacceptable. Students will be able to relate to their understanding of multithreading through writing a program that utilizes multithreading to solve a particular problem

Measuring the 4 C's of critical thinking:
With respect to the 4 C's of critical thinking (Curiosity, Connections, Creativity, and Communication), this course will focus on **Creativity** in finding a solution to the programming assignments, and **Communication** through team projects.

General Course Outline and expected course schedule:

1. Related programming material which includes but not limited to: Processes, Threads, System V/POSIX IPCs, Shared memory, Berkeley Sockets, Semaphores, Message Queues, and pipes. (Weeks 1-3)
2. Characterization of Distributed Systems. (Ch. 1) (Week 4)
3. System Models (Ch. 2) (Week4, Week 5)
4. Networking and Internetworking. (Ch. 3) (Week 5, Week 6)
5. Interprocess Communication (Ch. 4) (Week 7)
6. Remote Invocation (Ch. 5) (Week 8)
7. Indirect Communication (Ch. 6) (Week 9)
8. Operating System Support (Ch. 7) (Week 9, Week 10)
9. Peer-to-Peer Systems (Ch. 10) (Week 10)
10. Distributed File Systems (Ch. 12) (Week 11)
11. Name Services (Ch. 13) (Week 11)
12. Time and Global States (Ch. 14) (Week 12)

13. Coordination and Agreement (Ch. 15) (Week12, 13)
14. Replication (Ch. 18) (Week 13)
15. Mobile, Ubiquitous, and Cloud Computing (Ch. 19) (Week 14)

Exam Dates:

Midterm Exam: Tuesday March 10, 2015
Final Exam: Lecture time during finals week, Tuesday May 5, 2015.

Other Important Dates: (You must confirm on the University website)

First Day of classes:	January 20, 2015.
Spring Break:	March 16-22, 2015.
Last day to Drop/Withdraw-Regular 15-Week Session:	April 14, 2015.

Class Attendance:

This course is a face-to-face course that will have lectures and in-class presentations. Attendance is expected for all students, and is mandatory. It is the student's responsibility to obtain all the material covered in class, any announcements or handouts, and anything else conducted during class.

Exam and Assignment Policies:

Exam and quiz dates have already been determined. Every student must attend each exam/quiz on the set date and time. Assignments will be announced in class and posted on Blackboard and their due dates will be included in the assignments. There will be a 25% off penalty per day for late submission of each assignment for a maximum of two days per assignment. After the two days, the assignment will not be accepted and a grade of zero will be assigned.

Makeup exams and assignments are very restricted, and are only allowed in the case of a documented legitimate emergency (appropriate documentations must be provided). Verbal excuses are not acceptable under any circumstances.

Disability Accommodations:

Students that require special needs and/or have a disability need to contact Disability Services Office at 281-283-2627 website: www.uhcl.edu/disability as well as the instructor as soon as possible.

Academic Honesty:

HONESTY CODE of UHCL states: **I will be honest in all my academic activities and will not tolerate dishonesty.** Academic dishonesty will not be tolerated in this course. Students and faculty alike are required to adhere to the honesty code of the University of Houston Clear Lake! See the UHCL catalog for more details. You must be familiar with the UHCL academic dishonesty policy found in the student handbook. Copying the work of others and allowing others to copy your own work is not acceptable and is considered academic dishonesty. Also, sharing the course material after finishing this course is not allowed. Once you submit an exam, a quiz, or an assignment, this submission will be considered 100% your own work. If it is found not be 100% your own original work, then this will be considered an

academic dishonesty. This includes but not limited to accessing resources found online or any other source of information. Any violation of the dishonesty rules will result in a grade of zero, a 10% grade reduction of the total course grade, and filing an Academic Honesty Code Violation Form with the Dean of students for each incident and for all students involved in the incident.
Academic Honesty Policy:
http://prtl.uhcl.edu/portal/page/portal/DOS/Documents_and_Forms/Academic_Honesty_Policy.pdf

Use of Class Products in Assessment
The University of Houston–Clear Lake may use your work in this class to generate assessment data. Any works used will be used only for educational purposes.