|  |  |  |
| --- | --- | --- |
| utdbw83x35 | **Course** | **CS/CE/TE 6378.001.17S, Advanced Operating Systems** |
| **Professor** | Ravi Prakash |
| **Term** | Spring 2017 |
| **Meetings** | TR 8:30 am – 9:45 am, ECS North 2.110 |

**Professor’s Contact Information**

|  |  |
| --- | --- |
| **Office Phone** | (972) 883-2289 |
| **Other Phone** | (972) 883-2185 (CS Department Phone Number) |
| **Office Location** | ECS South 4.210 |
| **Email Address** | ravip@utdallas.edu |
| **Office Hours** | Tuesdays and Thursdays 10:00 am – 11:00 am |
| **Other Information** | The best way to communicate with me is through UTD email. Please make an appointment if you wish to meet me at times other than my office hours. |

**General Course Information**

|  |  |
| --- | --- |
| **Pre-requisites, Co-requisites, & other restrictions** | Pre-requisites:CS 5348 Operating Systems or equivalent, programming skills in C/C++ and/or in Java, socket programming skills, and working knowledge of a UNIX-based operating system. |
| **Course Description** | Concurrent processing, inter-process communication, process synchronization, deadlocks, introduction to queuing theory and operational analysis, topics in distributed systems and algorithms, checkpointing, recovery, multiprocessor operating systems. |
| **Learning Outcomes** | 1. Ability to understand the concepts of concurrent and distributed execution in modern operating systems and networks of systems. 2. Ability to understand the notion of time and clocks in a system with no global time-keeper. 3. Ability to understand the concept of causal ordering of events and deadlocks. 4. Ability to understand the concept of distributed mutual exclusion and resource management, including processor, memory and file systems. 5. Ability to design new algorithms/protocols for resource management. 6. Ability to understand the concept of process failure and approaches to build fault-tolerance in a distributed execution environment including checkpointing, voting protocols and replication. 7. Ability to design and conduct simulation experiments to quantitatively evaluate various distributed algorithms, and analyze and interpret the data. 8. Ability to communicate and work as a group on a team software project. |
| **Required Texts & Materials** | We will use a collection of research papers, some new and some old. The list will be distributed in class and will be available on eLearning. |
| **Suggested Texts, Readings, & Materials** | None. |

**Assignments & Academic Calendar**

*[Topics, Reading Assignments, Due Dates, Exam Dates]*

|  |  |
| --- | --- |
| **Number of Lectures** | **Topic** |
| **1** | Introduction |
| **6** | Theoretical Foundations (Causality, logical time, scalar and vector clocks, causally ordered message delivery, snapshot collection, termination detection). |
| **4** | Distributed Mutual Exclusion (Lamport’s algorithm, Ricart-Agrawala algorithm, Roucairol-Carvalho optimization, Maekawa’s algorithm, Raymond’s tree-based algorithm). |
| **3** | Physical clock synchronization |
| **2** | Agreement Protocols (Fischer paper). |
| **1** | Distributed Shared Memory |
| **1** | Distributed Scheduling |
| **6** | Recovery and Fault Tolerance (Model for recovery, Koo-Toueg algorithm, 2- and 3-phase commit, replica consistency, static and dynamic voting algorithms). |
| **4** | File Systems (Google file system, Dynamo, MapReduce, BigTable, Chubby: as time permits). |
| **Important Dates &**  **Times** | Last day of class: Thursday, April 27  Midterm Examination: Tuesday, February 28 (during class);  Final Examination: As per schedule determined by Registrar |
|  |  |

**Course Policies**

|  |  |
| --- | --- |
| **Grading (credit) Criteria** | Midterm exam: 25%, Final exam: 30%, Programming Project: 30%, Quizzes: 15%  In order to obtain an “A” or “A-” grade a student must perform well in the examinations, as well as in the programming projects. This is the minimum requirement, and satisfying this requirements does not guarantee an A or A- grade. |
| **Make-up Exams** | Make-up examinations will be offered only if the student has a valid medical reason and produces a doctor’s letter.  If a student has to be absent for several classes because of job related obligations, he/she will not be eligible for an incomplete grade. In such instances the student is advised to drop the course. |
| **Extra Credit** | No extra credit work will be assigned. |
| **Late Work** | Programming projects and homeworks submitted after the due date will be penalized at the rate of 10% of the total credit for that project/homework for every day (not including weekends and holidays) by which they are late. Late submissions will not be accepted once the solution has been discussed in class and the graded submissions have been returned. |
| **Class Attendance** | As per the Department of Computer Science policy, three consecutive absences leads to one letter grade drop. Four consecutive absences leads to an F. |
| **Classroom Citizenship** | The instructor encourages students to take active part in class discussions. No question is too simple/stupid to be asked. So, do not hesitate. |
| **Comet Creed** | *This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:*  *“As a Comet, I pledge honesty, integrity, and service in all that I do.”* |
| **UT Dallas Syllabus Policies and Procedures** | *The information contained in the following link constitutes the University’s policies and procedures segment of the course syllabus.*  *Please go to* [*http://go.utdallas.edu/syllabus-policies*](http://go.utdallas.edu/syllabus-policies) *for these policies.* |

***These descriptions and timelines are subject to change at the discretion of the Professor.***