**Operating Systems**

**Course Outline (SE)**

**Fall 2022**

1. **Course Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Code** | SENS3511 (Lab) | | | |
| **Course Title** | Operating Systems | | | |
| **Credit Hours** | 1 | | | |
| **Prerequisites by Course(s) and Topics** | Data Structures and Algorithms | | | |
| **Assessment Instruments with Weights** (homework, quizzes, midterms, final, programming assignments, lab work, etc.) | * **Lab (1 credit hr.)**   Final Lab Exam 40%  Lab Midterm 20%  Graded Labs 30%  Class participation 10%  **Lab Total 100%** | | | |
| **Course Instructors** |  | | | |
| **Course Coordinator** | Dr. Nauman Mazhar | | | |
| **Lab Instructors** |  | | | |
| **Plagiarism Policy** | Cheating is not only unethical; it is also against the University rules and will not be tolerated. Any form of cheating in programming assignments, homework problems, quizzes, and exams will result in strict action. Plagiarism detection tools will be used to determine who has cheated in programming assignments.  **All the parties involved will be awarded Zero in first instance. Second offense will result in three instruments being marked zero and on third offense case be forwarded to Disciplinary committee for an F grade in course.** | | | |
| **Attendance Policy** | Be in class on time, attendance will be marked and closed in the first five minutes of the class. | | | |
| **Submission Policy for assignments** | Manual assignments:   1. All manual assignments must be handwritten on a clear A4 page 2. No file or lamination should be done 3. Marks will be awarded for neatness   Coding assignments:   1. All assignments are to be submitted on Teams/Portal. 2. Email submissions will not be considered under any circumstance | | | |
| **Late submission Policy** | **24 hours late with 50% deduction in marks**  **More than 24 hours, 100% deduction** | | | |
| **Current Catalog Description** | This is designed to be the first undergraduate level course with the main objective of teaching students the concepts and principles that underlie the design and implementation of contemporary operating systems. The discussion of the concepts and principles used in UNIX, Linux, and MS-DOS/MS-Windows operating systems will be integrated throughout the course. Students will be given the opportunity to examine important UNIX and Linux data structures and use its various services through programming exercises. Programming language environment: C on UNIX/Linux | | | |
| **Grading** | All the grading will be posted on portal within a week of evaluation. Numbers posted will be deemed final after 1 week of posting(i.e. can be contest within 7 days ) | | | |
| **Textbook** (or **Laboratory Manual** for Laboratory Courses) | * Silberschatz, Galvin and Gagne, Operating System Concepts, 9th Edition. * Operating Systems Internals and Design Principles by Willam Stallings 5th edition | | | |
| **Reference Material** | * Tanenbaum, Andrew S., Modern Operating Systems, 3rd Edition, Prentice Hall. * E-book is available at <https://goo.gl/J7CkXG> | | | |
| **Course Goals** | * Students have developed a thorough understanding of the principles and concepts involved in the design of operating systems. * Students have an in-depth understanding of the various OS services for threads, inter-process communication, process synchronization, and file systems, offered as system or library calls in UNIX and Linux operating systems. | | | |
| **Learning Outcomes** | Students who complete this course successfully are expected to:   1. High-level understand and explaining what is an operating system and the role it plays 2. A high-level understanding of the structure of operating systems describing its applications, and the relationship between them. 3. Express some knowledge of the services provided by operating systems. 4. Apply and gain extensive knowledge on principles and modules of operating systems 5. State and understand key mechanisms in process & thread management 6. Discover issues in concurrent processes and threads 7. Develop and understand memory management, virtual memory concepts, 8. highlight the synchronization and resource management techniques leading to deadlocks 9. compare performance of processor scheduling algorithms 10. produce algorithmic solutions to process synchronization problems 11. use modern operating system calls such as Linux process and synchronization libraries 12. practice with operating system concepts such as process management and synchronization | | | |
| **Topics Covered in the Course, with Number of Lectures on Each Topic** (assume 15-week instruction and one-hour lectures) | Attached | | | |
| **Programming Assignments Done in the Course** | Yes | | | |
| **Class Time Spent on** (in credit hours) | **Theory** | **Problem Analysis** | **Solution Design** | **Social and Ethical Issues** |
|  | 1.5 | 1.0 | 1.0 | 0.5 |
| **Oral and Written Communications** |  | | | |

**LECTURE BREAKDOWN**

|  |  |  |
| --- | --- | --- |
| **Lab No.** | **Lab Task** | **Evaluation** |
| 01 | **Practice Lab-01:** Ubuntu Installation and Setting up Virtual Machine and Task on C, C++ Difference, Linux Sub-System for Windows. | **NG** |
| 02 | **Practice Lab-02:** Overview of Ubuntu Directories and Basic Commands | **NG** |
| 03 | **Practice Lab-03:**  C Programming Introduction.  **GRADED LAB-01:**  Graded Lab on Linux Commands | **GL01, CP01** |
| 04 | **Practice Lab-04:** Process Creation (Fork) | **CP02** |
| 05 | **GRADED LAB-02:**  Process Creation. | **GL02** |
| 06 | **Practice Lab-05:**  Inter Process Communication: Pipes, FIFO on terminal and in C. **GRADED LAB-03:**  Pipes and FIFOs | **CP03, GL03** |
| 07 | **Practice Lab-06:**  I/O Redirection, dup, dup2 on terminal and in C. | **CP04** |
|  | **Mid-Term Exam** |  |
| 08 | **Practice Lab-07:**  CPU Scheduling. |  |
| 09 | POSIX Threads. |  |
| 10 | Matrix Multiplication Threads |  |
| 11 | Producer Consumer Problem |  |
| 12 | Reader Writer Problem |  |
| 13 | Dining philosopher Problem + Semaphore |  |
| 14 | Revision Lab |  |
|  | **Final-Term Exam** |  |