

**Course Name:** Operating System Lab

**Course Code:** T7511

**Faculty:** Engineering

**Course Credit:** 1

**Course Level:** 2

**Sub-Committee (Specialization):** Computer Science and Engineering

**Learning Objectives:**

**The students will be able to:**

* 1. Compute Waiting/Turnaround time of processes using various Scheduling Algorithms using C program.
  2. Understand basic concepts of Linux and Fedora Operating System.
  3. Practice different Linux shell commands and interpret their results
  4. Relate File Operations of Linux to basic structure of File Organization of Linux.
  5. Produce results using shell commands using various arithmetic commands for the given problem
  6. Illustrate how deadlocks can be detected using Bankers algorithm.
  7. Show how various looping conditional statements can be implemented in Linux.
  8. Demonstrate memory allocation for processes by Page Replacement Algorithm using C program.

**Program Outcome (PO):**

1. Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, societal, and environmental considerations.
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.
5. Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Function effectively as an individual and as a member or leader indiverse teams and in multidisciplinary settings.
10. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
11. Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological change.
12. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

**Program Specific Outcomes (PSO):**

1. To apply the concepts of computer engineering and practical knowledge in analysis, design and development of computing systems and applications to multi-disciplinary problems.
2. To provide a concrete foundation to the students in the cutting edge areas of CSE and further help them in excelling in chosen areas of advanced computer science like Machine Learning, Algorithms, Data Science, Internet of Things, Computing and Data Security and Privacy.

**Pre-Requisites:** None

**Books Recommended:**

|  |  |  |
| --- | --- | --- |
| **Book** | **Author** | **Publisher** |
| Advanced UNIX Programming ,2nd Edition, 2005 | W. R. Stevens | Addison-Wesley Professional, ISBN: 8131700054 |
| Fedora Linux: A Complete Guide to Red Hat's Community Distribution ,1st Edition,2007 | Chris Tyle | O'Reilly Publication, ISBN:978059652682 5. |
| Linux in a Nutshell, 6th Edition,2009 | E Siever, S Figgins, R Love, A Robbins | O'Reilly Publication, ISBN: 9780596154486. |
| Linux vs. Windows: A Comparison of Application and Platform Innovation Incentives for Open Source and Proprietary Software Platforms | The Economics of Open Source Software Development | NA |
| Mobile OS and efforts towards open standards | Dotcom Infoway | NA |
| Scheduling Algorithms and Operating Systems Support for Real-Time Systems | NA | NA |
| Ted Comparing Operating Systems using robustness benchmarks | Marz | NA |
| Unix Shell Programming, 1st Edition, 2008 | Y Kanetka | BPB Publications, ISBN: 978-81-7029-753-2. |

**Course Outline:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No.** | **Topic** | **Actual Teaching Hours** | **Contact Hours Equivalence** |
| 1 | To implement basic commands on Ubuntu and ls commands with all options. | 2 | 2 |
| 2 | To implement C program using Vi editor. | 2 | 2 |
| 3 | To implement all process management commands of UNIX. | 4 | 4 |
| 4 | To write a program to demonstrate process creating (Parent and Child). | 2 | 2 |
| 5 | To write C program to implement FCFS CPU scheduling algorithm. | 2 | 2 |
| 6 | To write C program to implement SJF CPU scheduling algorithm | 2 | 2 |
| 7 | To write C program to implement Priority based CPU scheduling algorithm. | 4 | 4 |
| 8 | To implement Banker’s Algorithm. | 4 | 4 |
| 9 | To implement basic shell commands and shell scripts. | 4 | 4 |
| 10 | To write a shell script to demonstrate if-else conditional statement. | 4 | 4 |
| **Virtual Lab Experiments (open ended experiments)** | | | |
| 11 | To write a shell script to demonstrate loop control structures | 2 | 2 |
| 12 | To write a shell script to perform read and write operations on file. | 2 | 2 |
| Total | | 34 / 30 | 34 /30 |

**Evaluation:**

**Pedagogy:**

1. Continuous Assessment: Lab Assignments, Open Ended Problems, virtual lab
2. End Semester Examination: Viva-Voce Lab Exam

Hands on Session Project based Learning

HackerRank/ Github/ Virtual Lab

**CO-PO-Mapping:**

The mapping is typically represented using a scale (1, 2, 3) where:

1: Low Contribution

2: Medium Contribution

3: High Contribution

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| **CO1** | **3** | **3** | **1** | **1** | **1** | **-** | **-** | **-** | **-** | **1** | **-** | **2** | **3** | **3** |
| **CO2** | **3** | **3** | **2** | **2** | **2** | **-** | **-** | **-** | **-** | **1** | **-** | **2** | **3** | **3** |
| **CO3** | **3** | **3** | **3** | **2** | **2** | **-** | **-** | **-** | **-** | **1** | **-** | **2** | **3** | **3** |
| **CO4** | **3** | **2** | **2** | **2** | **3** | **-** | **-** | **-** | **-** | **2** | **-** | **2** | **3** | **3** |
| **CO5** | **1** | **1** | **2** | **2** | **2** | **-** | **-** | **-** | **3** | **3** | **2** | **2** | **3** | **3** |