**d-**

**OPERATING SYSTEMS and UNIX**

Course Code : IS632 Credits : 4:0:0

Prerequisites: NIL Contact Hours : 56

Course coordinator: Dr.Megha. P. Arakeri

Course objectives:

1. Introduce various concepts of operating system
2. Understand Process synchronization and deadlock by the operating systems
3. Explore he memory management function of operating system
4. Realize the role of operating system in file management.
5. Understand system protection and acquire knowledge on modern operating system.

Course Contents:

UNIT -1

Introduction: Operating System Structure, Operating System Operations. Process Concept: Process Scheduling, Operations on Processes, Interprocess Communication; Process Scheduling: Scheduling Criteria, Scheduling Algorithms. UNIX Architecture, Unix Commands- Directory, file, Protection and Security, Communication, Information, Process management, I/O redirection & Piping and Filter Commands.

UNIT -2

Synchronization: The Critical Section Problem, Peterson’s Solution, Synchronization Hardware, Semaphores, Classical Problems of Synchronization, Monitors, Synchronization Examples; Dead locks: System Model, Deadlock Characterization, Methods for handling Deadlocks, Deadlock Prevention, Deadlock Avoidance and detection, Recovery from Deadlock.

UNIT -3

Memory Management Strategies: Background, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation; Virtual Memory Management: Background, Demand Paging, Copy on Write, Page Replacement, Allocation of frames, Allocating Kernel Memory.

UNIT -4

File System: File Concept, Access Methods, Directory Structure, Protection; Implementing File Systems: File System Structure, File System Implementation, Directory Implementation, Allocation Methods, Free Space Management, Efficiency and Performance, Recovery; Secondary-Storage Structure: Disk structure, Disk Attachment, Disk Scheduling Methods. Disk management. Swap-Space Management.

**UNIT -5**

System Protection: Goals of Protection, Principle of Protection, Access Matrix, Domain of Protection, Access Matrix and its implementation. Case Studies: The Linux System: Design Principles, Kernel Modules, Process Management, Scheduling, Memory Management, File Systems, Interprocess Communication.

**Text Books:**

1. Silberschatz, Galvin, Gagne, Operating System Concepts, 7/e, John Wiley and sons, 2007.

**Reference Books:**

1. Dhamdhere D. M., Operating Systems, A Concept Based Approach, 2/e, Tata McGraw Hill, 2006.

2. Tanenbaum A. S., Modern Operating Systems, 3/e, Pearson Education, 2008.

**Course Outcomes:**

The students will be able to

**CO1:** Discuss operating system concepts and analyze CPU scheduling algorithms. **(PO- a,b,d)**

**CO2:** Describe the various methods for accessing shared data and handling deadlocks.**(PO-a,b,d)**

**CO3:** Employ the techniques for memory management during process execution.

**(PO-a,b,d)**

**CO4:** Describe the file system and secondary storage structure.**(PO-a,b,c,d)**

**CO5:** Articulate the need for information protection mechanisms in OS and illustrate the working of modern operating system.**(PO –a,b,c,l)**

**OPERATING SYSTEMS and UNIX LAB**

Course Code : IS632L Credits : 0:0:1

Prerequisites: NIL Contact Hours : 28

Course coordinator: T. Tamilarasi

Course Objective:

1) Implement Unix system calls

2) Implement the scheduling algorithms and analyze their performance.

3) Implement the strategies used for concurrency management.

4) Implement the various schemes used for memory management.

5) Implement the different methods used for handling files.

Course Contents:

Implement the following:

1) Simulate all file related, protection and security commands

2) Simulate Process, I/O Redirection and piping and filters commands.

(The above two programs will not be included for Final exam.

3) Write a shell programs using Command line arguments, command Substitution and Positional Parameters

4) Write a Shell program using set, shift, Control structures and Looping structures

5) Write a C program using Process system call(fork, exit) and executing another program using exec system call

6) Write a C Program using File System call

7) Demonstrate directory system call

8) Write a C program implement Orphan and Zombie process, security related calls

9) Process scheduling using FCFS and RR

10) Process handling using deadlock mechanisms

11) Process Synchronization using semaphores

12) Memory management using page replacement

Text Books:

2. Silberschatz, Galvin, Gagne, Operating System Concepts, 7/e, John Wiley and sons, 2007.

Reference Books:

3. Dhamdhere D. M., Operating Systems, A Concept Based Approach, 2/e, Tata McGraw Hill, 2006.

4. Tanenbaum A. S., Modern Operating Systems, 3/e, Pearson Education, 2008.

47 of 69

Course Outcome: At the end of the course, the students will be able to

CO1: Develop simple command level program for file, process, redirection, piping, protection and security. (PO a, PO b, PO c. PO k)

CO2: Demonstrate the usage of shell using shell positional parameters and command substitution.(PO a, PO e, PO g, PO k)

CO3:Create a Child process using fork system call and execute other programs. (PO a, PO b, PO e, PO k)

CO4 :Apply scheduling, synchronization and deadlock mechanisms on processes.

(PO-a,d,l)

CO5: Manage main memory through the paging schemes.(PO-a,d,l)

SOFTWARE ENGINEERING

Course Code : IS433 Credits : 3:0:0:0

Prerequisites: NIL Contact Hours : 42L

Course coordinator(s): Rajaram M Gowda

Course objectives: Introduce concept of software engineering and understand SDLC processes. Identify the software requirements. Explore various design concepts. Identify Software Quality Assurance and testing concepts. Elucidate Project Management concepts and metrics.

Course Contents:

Unit – 1

Introduction: FAQs about software engineering, Professional and ethical responsibility, Software processes: Software process models, process iteration, process activities, the Rational Unified process, Computer Aided Software Engineering (CASE), Rapid software development: Agile methods, Extreme programming, Rapid application development and software prototyping.

Unit – 2

Software Requirements: Functional and Non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. Requirements Engineering Processes: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management. Case studies: ATM, LIBSYS. Object oriented analysis.

Unit – 3

Design Concepts: Design within the context of software engineering, the design process, design concepts, the design model. Architectural Design: Software architecture, Architectural genres, A brief taxonomy of Architectural styles. Component-level design: what is a component? Designing Class-based components. User Interface design: The golden rules, User interface analysis and design, design issues, Static and Dynamic modeling.

Unit – 4

Software Quality Assurance: Background issues, Elements of software quality assurance, SQA tasks, goals, and metrics, Formal approaches to SQA, Statistical software quality assurance. Software Testing Strategies: A strategic approach to software testing, strategic issues, test strategies for conventional software, validation testing, system testing, the art of debugging, Object oriented testing.

Unit – 5

Project management concepts: The management spectrum, people, the product, the process. Process and Project metrics: Metrics in the process and project domains, software measurement, metrics for software quality, integrating metrics within the software process, metrics for small organizations, establishing a software metrics program, CMM, CMMI, PCMM.

**Text Books**:

1. Roger S.Pressman, Software Engineering-A Practitioners approach, Seventh Edition, McGraw-Hill, 2007.

2. Ian Sommerville, Software Engineering, Eighth Edition, Pearson Education, 2007.

**Reference Books**:

1. Shari Lawrence Pfleeger, Joanne M. Atlee, Software Engineering Theory and Practice, Third Edition, Pearson Education, 2006.

2. Waman S Jawadekar, Software Engineering Principles and Practice, Tata McGraw Hill, 2004.

3. Douglas Bell, Software Engineering for Students, A Programming Approach, 4th Edition, Pearson Education.

**Course outcomes**: The students will be able to

**CO1:**Describe software development life cycle processes.**(PO 1, 2, 8) (PSO 3)**

**CO2:** Analyze software requirements and generate SRS.**(PO 1, 2, 3, 5, 9 ,10) (PSO 1)**

**CO3:** Describe design concepts and develop design document. **(PO-1, 2, 5, 9 , 10) (PSO 2, 3) CO4**: Describe SQA tasks, goals, and metrics, and test strategies. **(PO 1, 2, 5) (PSO 1, 3)**

**CO5:** Explain Project management concepts and metrics.**(PO 1, 10, 12) (PSO 1, 3)**