ICT 2227 : OPERATING SYSTEMS [4 0 0 4]

Course Objectives

- Get familiarized with the basic functionality and the evolution of different types of operating systems.
- Learn various algorithms related to CPU scheduling, deadlocks, memory management, and storage management.
- Understand the basic aspects of real time operating systems

Abstract

Introduction to Operating systems, Operating System Services, Operating system Structure, System calls, Process management: Process concept, Threads, Inter-process communication, CPU Scheduling, Process synchronization, Handling deadlocks: Deadlock Characterization, Deadlock Detection, Prevention, Avoidance and Recovery, Memory management: Main memory, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Virtual memory: Demand Paging, Page Replacement, Thrashing, Allocating Kernel Memory, Storage Management: File management, Disk scheduling, Case study on Unix based Operating system: Design Principles, Kernel Modules, Basic concepts of Real time operating systems: Classification of Real Time Systems, Microkernels, Scheduling.

Syllabus

Introduction

Operating system structure, Operating system operations, Distributed systems, Special purpose systems, Computing environments, Open source operating systems.

[3 hours]

CPU Scheduling

Process concepts: Process states, Process control block, Scheduling queues, Schedulers, Context switch, Multi-threaded programming: Overview, Multithreading models, Threading issues, Process scheduling: Basic concepts, Scheduling criteria, scheduling algorithms.

[**7 hours]**

Process Synchronization

Critical section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, monitors.

[6 hours]

Deadlocks

Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

[8 hours]

Memory management

Memory management strategies, Swapping, Contiguous memory allocation, Paging, Structure of the page table, Segmentation.

[6 hours]

Virtual Memory

Demand paging, copy on write, page replacement, allocation of frames, thrashing.

[7 hours]

Storage Management

File concept, Access methods, directory structure, file system structure, directory implementation, allocation methods, free space management, disk structure, and disk-scheduling

[5 hours]

Case study on UNIX based Operating system

Design principles, Kernel modules, Process management, Memory management.

[2 hours]

Real time operating systems

Characteristics of Real time operating systems, classification of real time systems, Micro kernels and RTOS, scheduling in RTOS, Rate monotonic scheduling, EDF, Priority inversion.

[4 hours]

Course Outcomes

At the end of this course, the students will be able to

1. Illustrate the design principles and functionalities of different operating systems.
2. Demonstrate the working of various algorithms for CPU scheduling, synchronization and deadlocks.
3. Assess the memory management and storage management techniques and their suitability in different operating systems.
4. Apply the concepts of Real Time Operating Systems in application development.

References

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, *Operating System Concepts*, 9th edition, Wiley, 2012.
2. William Stallings, *Operating Systems: Internals and Design Principles*,9th edition, Pearson, 2017.
3. Phillip A Laplante, Seppo J Ovaska, *Real time systems design and analysis*, 4th edition, Wiley, 2013.
4. Rajib Mall, *Real time systems: Theory and Practice*, 2nd edition, Pearson, 2009.

ICT 2244 OPERATING SYSTEMS LAB [0 0 3 1]

Course Objectives

- Execute shell scripts in UNIX based operating system.

- Implement inter process communication using system calls.
- Implement algorithms for CPU scheduling as well as process synchronization

Abstract

UNIX based operating system commands, executing shell scripts, inter process communication using system calls, implementing CPU scheduling algorithms, memory and deadlock management.

Course Outcomes

At the end of this course, students will be able to

1. Demonstrate the working of UNIX based operating systems
2. Demonstrate process management in operating systems
3. Implement CPU scheduling as well as synchronization algorithms
4. Implement algorithms used to understand the functionality of modern operating systems

References

1. Blurn R.& Bresnahan C., *Linux Command Line Shell Scripting Bible,* 3rd edition, *Wiley,* 2015.
2. Silberschatz A., Galvin P.B.& Gagne G., *Operating System Concepts*, 9th edition, Wiley, 2012.