OPERATING SYSTEMS

Anatomy of the Course

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

Course Details

- Subject Code : UE19CS254
- Subject Title : Operating Systems
- Subject Credits: 04
- Sections: B.Tech 4 CS - C, D, H
- Faculty: Nitin V Pujari

Course Assessments

- In Semester Assessment
  - ISA - 1: 60 Marks - Unit 1, 2, 3 => reduced to 21
  - ISA - 2: 40 Marks - Unit 4, 5 => reduced to 14
  - Assignment: 15 Marks => reduced to 15
  - Total =>50
- End Semester Assessment
  - 100 Marks - Unit 1, 2, 3, 4, 5 => Reduced to 50
  - Total =>50
- Total Marks = ISA + ESA = 50+50 => 100 Marks

OPERATING SYSTEMS

Course Syllabus

UNIT 1: Introduction and Process Management

Operating-System Structure & Operations, Kernel Data Structures,Computing Environments, Operating-System Services, OperatingSystem Design and Implementation. Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination, CPU Scheduling and Scheduling Algorithms, IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary. Case Study: Linux/Windows Scheduling Policies.

UNIT 2: Threads and Concurrency

Introduction to Threads, types of threads, Multicore Programming, Multithreading Models, Thread creation, Thread Scheduling, PThreads and Windows Threads, Mutual Exclusion and Synchronization: software approaches, principles of concurrency, hardware support, Mutex Locks, Semaphores. Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts. Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads. Deadlocks: principles of deadlock, tools for detection and Prevention.

UNIT 3: Memory Management

Main Memory: Hardware and control structures, OS support, Address translation, Swapping, Memory Allocation (Partitioning, relocation), Fragmentation, Segmentation, Paging, TLBs context switches Virtual Memory – Demand Paging, Copy-on-Write, Page replacement policy – LRU (in comparison with FIFO & Optimal), Thrashing, design alternatives – inverted page tables, bigger pages. Case Study: Linux/Windows Memory.

Unit 4 : Storage Management

Mass-Storage Structure – Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing. File System Implementation/Internals: File control Block (inode), partitions & mounting, Allocation methods. Case Study: Linux/Windows File Systems

Unit 5 : I/O Management and Security

I/O Hardware, polling and interrupts, DMA, Kernel I/O Subsystem and Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering. System Protection: Goals, Principles and Domain of Protection, Access Matrix, Access control, Access rights. System Security: The Security Problem,Program Threats, System Threats and Network Threats. Case Study: Windows 7/Windows 10

OPERATING SYSTEMS

Course Outline - Unit 1

Class No.	Chapter Title / Reference Literature	Topics to be covered	% of Portions covered	
			Reference chapter	Cumulative
1	1.1-1.2	What Operating Systems Do, Computer-System Organization?	1	21.4
2	1.3,1.4,1.5	Computer-System Architecture, Operating-System Structure & Operations	1	
3	1.10,1.11	Kernel Data Structures, Computing Environments	1	
4	2.1,2.6	Operating-System Services, Operating System Design and Implementation	2	
5	3.1-3.3	Process concept: Process in memory, Process State, Process Control Block, Process Creation and Termination	3	
6	5.1-5.2	CPU Scheduling: Basic Concepts, Scheduling Criteria	5	
7	5.3	Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling	5	
8	5.3	Scheduling Algorithms: Shortest-Job-First Scheduling (Pre-emptive), Priority Scheduling	5	
9	5.3	Round-Robin Scheduling, Multi-level Queue, Multi-Level Feedback Queue Scheduling	5	
10	5.5,5.6	Multiple-Processor Scheduling, Real-Time CPU Scheduling	5	
11	5.7	Case Study: Linux/Windows Scheduling Policies	5	
12	3.4,3.6.3	IPC – Shared Memory & Message Passing, Pipes – Named and Ordinary	3,6	

OPERATING SYSTEMS

Course Outline - Unit 2

13	4.1,4.2	Introduction to Threads, types of threads, Multicore Programming.	4	42.8
14	4.3,5.4	Multithreading Models, Thread creation, Thread Scheduling	4	
15	4.4	Pthreads and Windows Threads	4	
16	6.1-6.3	Mutual Exclusion and Synchronization: software approaches	6	
17	6.3-6.4	principles of concurrency, hardware support	6	
18	6.5,6.6	Mutex Locks, Semaphores	6	
19	6.7.1-6.7.3	Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts	6	
20	6.9	Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads.	6	
21	Handouts	Demonstration of programming examples on process synchronization		
22	7.1-7.3	Deadlocks: principles of deadlock, Deadlock Characterization.	7	
23	7.4	Deadlock Prevention, Deadlock example	7	
24	7.6	Deadlock Detection	7	

OPERATING SYSTEMS

Course Outline - Unit 3

25	8.1	Main Memory: Hardware and control structures, OS support, Address translation,	8	64.2
26	8.2	Swapping, Memory Allocation (Partitioning, relocation), Fragmentation,	8	
27	8.4	Segmentation	8	
28	8.5	Paging	8	
29	8.5	TLBs context switches	8	
30	8.6	Structure of page tables	8	
31	8.6.3-8.7	design alternatives - inverted page tables, bigger pages	8	
32	9.1-9.2	Virtual Memory - Demand Paging,	9	
33	9.39.4.1-9.4.3	Copy-on-Write, Page replacement: Basic page replacement (FIFO page replacement and optimal page replacement)	9	
34	9.4.4, 9.5	Page replacement, Allocation of frames	9	
35	9.6	Thrashing	9	
36	9.10	Case Study: Linux/Windows Memory	9	

OPERATING SYSTEMS

Course Outline - Unit 4

37	12.1	Mass-Storage Structure – Mass-Storage overview	12	82.1
38	12.4	Disk Scheduling	12	
39	12.4	Disk Scheduling	12	
40	12.6-12.7	Swap-Space Management, RAID structure	12	
41	10.1-10.2	File Concept, Access Methods	10	
42	10.3	Directory and Disk Structure	10	
43	10.4-10.6	File-System Mounting, File Sharing, Protection	10	
44	11.1-11.3	File-System Structure, File-System Implementation, Directory Implementation	11	
45	11.4	Allocation methods	11	
46	16.7,11.8	Case Study: Linux/Windows File Systems	11,16	

OPERATING SYSTEMS

Course Outline - Unit 5

47	13.1-13.2	I/O Hardware, polling and interrupts, DMA	13	100
48	13.4	Kernel I/O Subsystem	13	
49	13.5	Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering.	13	
50	14.1-14.3	System Protection: Goals, Principles and Domain of Protection.	14	
51	14.3.2	Domain of Protection: Unix, MULTICS examples	14	
52	14.4	Access Matrix	14	
52	14.5	Implementation of Access Matrix, Access control, Revocation of Access rights.	14	
54	15.1	System Security: The Security Problem	15	
55	15.2	Program Threats.	15	
56	15.317.5	System Threats and Network ThreatsCase Study: Windows 7/Windows 10	15,17	

Course Objectives

Course Objectives

- The course focuses on fundamental operating system concepts.
- The course provides an understanding of various components of the operating system.
- The course delves deeper into various algorithms and associated trade-offs for efficient resource management such as process, disk, and memory management.
- The course will introduce design principles and trade-offs in the design of Operating Systems.
- The course will also introduce the concepts such as security, protection and virtualization

Course Outcomes

Course Outcomes

- Gain extensive knowledge on principles and modules of Operating Systems.
- Understand the design of various algorithms for scheduling and their relative performance.
- Design pieces of the operating systems such as process management, concurrent processes and threads, memory

OPERATING SYSTEMS

Course References

Operating System Concepts

Abraham Silberschatz, Greg Gagne, and Peter Baer Galvin, Ninth Edition, 2013

Operating Systems - Internals and Design Principles,

William Stallings,9th edition, pearson, 2018

Operating Systems: Three Easy Pieces

Remzi Arpaci, Self-publishing, 2016

Rerence Book to understand DMA concepts in detail

Advanced Programming in the Unix Environment

Richard Stevens and Stephen A Rago, 3rd edition, Pearson,2017

To understand unix system calls like fork(), exec(), wait(), memory layout of a c program, system calls related to files.

Operating Systems

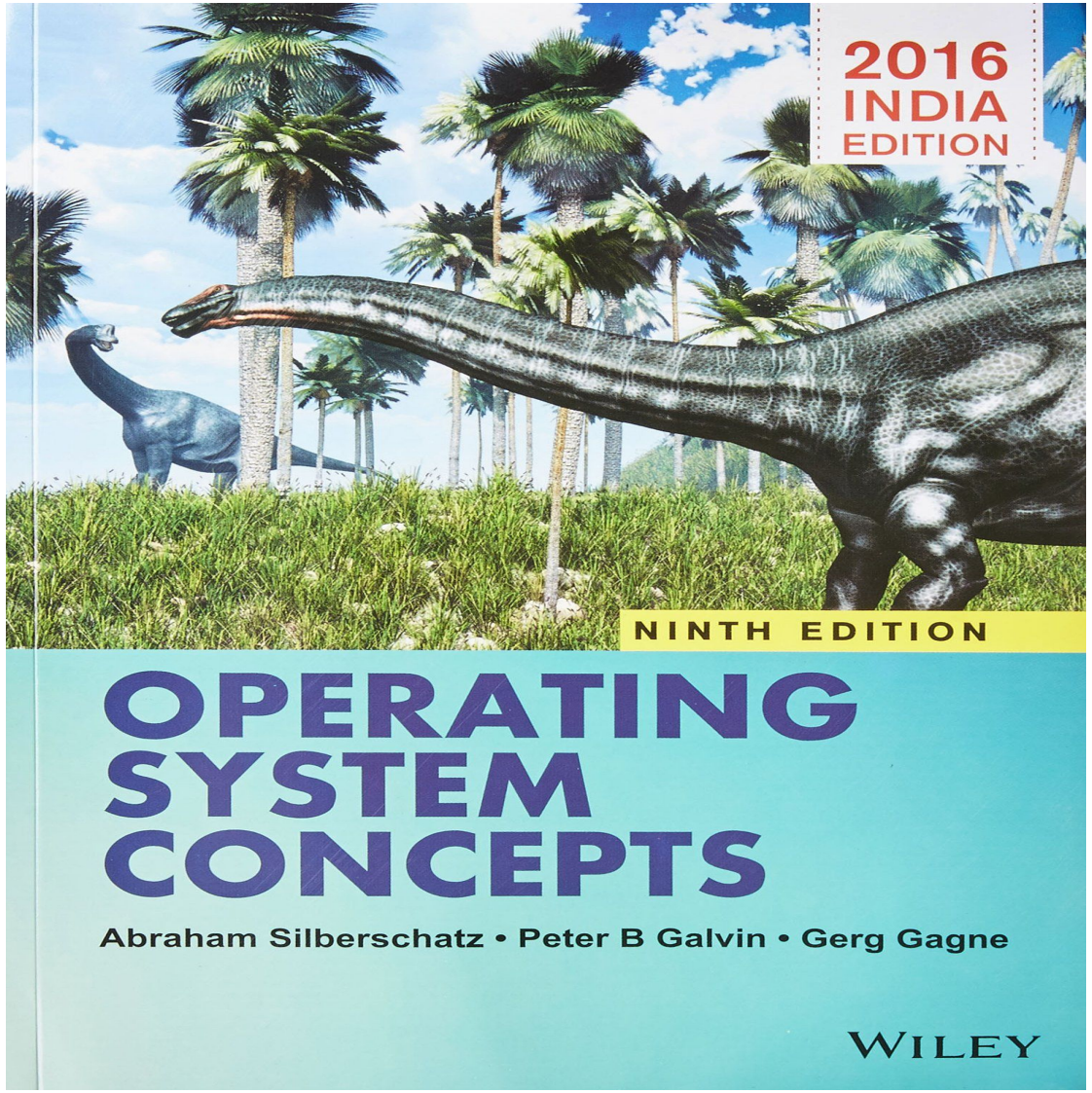
Harvey Deitel, Paul Deitel, David Choffnes,3rd edition, Prentice Hall,2004

Modern Operating Systems

Andrew S Tanenbaum, 3rd edition, Pearson, 2007

OPERATING SYSTEMS

Text Book



PES UNIVERSITY ONLINE

For Course Deliverables by the Anchor Faculty click on

www.pesuacademy.com

My Course Link on Teams

https://teams.microsoft.com/l/team/19%3a0e45de38a46c4c66bd0bd058c59f9056%40thread.tacv2/conversations?groupId=42c21380-afc4-47a3-bc3e-17af7ae78ae9&tenantId=e290fb02-d184-4a8c-ae49-c83b04485909

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com