Course Lead: Dr. Krishna Kant Agrawal

Name of the Course	**Operating System**	**L**	**T**	**P**	**S**	**C**
		3	**0**	**1**	**0**	**4**

Course Title	**Operating System**			**Course Type**			**Integrated**		
Course Code				**Class**			**B. Tech.**		
Instruction delivery	**Activity**	**Credits**	**Credit Hours**	**Total Number of Classes per Semester**				**Assessment in Weightage**	
	Lecture	**3**	**3**						
	Tutorial	**0**	**0**	**Theory**	**Tutorial**	**Practical**	**Self-study**	**CIE**	**SEE**
	Practical	**1**	**2**						
	Self-study	**0**	**0**						
	Total	**4**	**5**	**45**	**0**	**30**	**0**	**50%**	**50%**
Names Course Instructors	**Course Lead**		**Dr. Krishna Kant Agrawal**						
	Theory			**Practical**					
	Dr. Krishna Kant Agrawal, Professor SCSE.								

Course Overview:

This course will introduce the student to the basic concepts involved in the design and implementation of an operating system. Students will be made familiar to the important modules of operating systems like process management, memory management, file systems, synchronization primitives and exception handling. Important data structures used in the design of these modules will be introduced. The accompanying lab course is intended to give students an illustration of the concepts introduced in the theory course.

Course Objectives:

1. To understand the services provided by and the design of an operating system.
2. To understand the structure and organization of the file system.
3. To understand what a process is and how processes are synchronized and scheduled.
4. To understand different approaches to memory management.
5. To understand system calls for managing processes, memory and the file system.
6. To understand the data structures and algorithms used to implement an OS.

Course Outcomes:

After completion of this course, student will be able to

• **CO1:** Understanding the role of operating system with its function and services.

• **CO2:** Understand various algorithm used for CPU Scheduling, Memory management and Disk Scheduling.

• **CO3:** Apply various concepts related with Deadlock to solve Problems.

• **CO4:** Analyse Protection and Security Mechanism in Operating System.

Program Outcomes:

PO1 Computing Science knowledge: Apply the knowledge of mathematics, statistics, computing science and information science fundamentals to the solution of complex computer application problems.

PO2 Problem analysis: Identify, formulate, review research literature, and analyze complex computing science problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and computer sciences**.**

PO3 Design/development of solutions: Design solutions for complex computing 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.

PO4 Conduct investigations of complex problems: 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.

PO5 Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern computing science and IT tools including prediction and modeling to complex computing activities with an understanding of the limitations.

PO6 IT specialist and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional computing science and information science practice.

PO7 Environment and sustainability: Understand the impact of the professional computing science solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the computing science practice.

PO9 Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 Communication: Communicate effectively on complex engineering activities with the IT analyst 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.

PO11 Project management and finance: Demonstrate knowledge and understanding of the computing science 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.

PO12 Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSO's)

PSO1: Have the ability to work with emerging technologies in computing requisite to Industry 4.0.

PSO2: Demonstrate Engineering Practice learned through industry internship and research project to solve live problems in various domains.

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	–	–	–	–	–	–	–	–	–	–	–	1	1
CO2	2	1	1	–	–	–	–	–	–	–	–	–	–	1
CO3	3	2	2	2	1	–	–	–	–	–	–	–	2	2
CO4	3	2	2	2	2	–	–	–	–	–	–	–	1	2

SNo	Assessment Tools	CIE				Total CIE marks	SEE
		CAT1	CAT2	LAB	LAB EXAM		
2	Integrated	A1	A2	A3	A4		
		30	30	20	20	100	100

Course Content (Theory)

Introduction : Operating system and functions, Classification of Operating systems- Batch, Interactive, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.

Concurrent Processes: Process Concept, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.

CPU Scheduling: Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Process Control Block (PCB), Process address space, Process identification information, Threads and their management, Scheduling Algorithms, Multiprocessor Scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.

Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

I/O Management and Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, File system implementation issues, File system protection and security.

Session	Topics	Skills to be Learned
1.	Introduction of Operating system ,Operating system and functions	Understanding different types of Operating system and their classification, structure and system concept.
2.	Classification of Operating systems, Batch, Interactive, Time sharing	
3.	Real Time System, Multiprocessor Systems	
4.	Multiuser Systems, Multi process Systems, Multithreaded Systems	
5.	Operating System Structure-Layered structure	
6.	System Components	
7.	Reentrant Kernels, Monolithic and Microkernel Systems.	
8.	Process Concept, Process States, Process Transition Diagram	
9.	Schedulers, Process Control Block (PCB)	
10.	Process Scheduling, Operations on Processes	Understand the process concepts of Operating system, process concepts of Operating system
11.	Inter Process Communication models and Schemes,	
12.	Threads and their management, Multithreading Models	
13.	CPU Scheduling Concepts, Scheduling Criteria, Schedulers	
14.	Scheduling Algorithms (FCFS, SJF)	
15.	Scheduling Algorithms (Priority Scheduling Algorithm	
16.	Scheduling Algorithms (RR, Multilevel queue)	
17.	Scheduling Algorithms (Multilevel queue and Multilevel feedback Queue)	
18.	Multiprocessor Scheduling.	
19.	Process Synchronization, Principle of Concurrency	Know how to prevent deadlock in Operating system, know how to avoid, detect and recover deadlock in Operating system
20.	Producer / Consumer Problem, Mutual Exclusion	
21.	Critical Section Problem, Dekker's solution, Peterson's Problem	
22.	Semaphores, Test and Set operation	

23.	Classical Problem in Concurrency-Dining Philosopher Problem	
24.	Sleeping Barber Problem	
25.	Deadlock: System model, Deadlock characterization, Prevention	
26.	Avoidance and detection, Recovery from deadlock.	
27.	Memory Management: Basic bare machine, Resident monitor	
28.	Multiprogramming with fixed partitions	
	Multiprogramming with variable partitions	
29.	Protection schemes, Paging	Understand about memory management concept, concept of paging, concept of I/O Management and Disk Scheduling
30.	Segmentation, Paged segmentation	
31.	Virtual memory concepts	
32.	Demand paging, Performance of demand paging	
33.	Page replacement algorithms	
34.	Page fault	
35.	Thrashing, Cache memory organization	
36.	Locality of reference.	
37.	I/O Management and Disk Scheduling: I/O devices	Understand the concept of File System, File organization and its mechanism,
38.	I/O subsystems, I/O buffering,	
39.	Disk storage and disk scheduling	
40.	RAID. File System: File concept	
41.	File organization and access mechanism,	
42.	File directories, and File sharing	
43.	File system implementation issues	
44.	Files system Protection and securities	
45.	Recent Trends and Development	

Course Content (Practical)

1. Simulate the following CPU scheduling algorithms. a) FCFS b) SJF c) Round Robin d) Priority.
2. Write a C program to simulate producer-consumer problem using Semaphores
3. Write a C program to simulate the concept of Dining-philosophers problem.
4. Simulate MVT and MFT.
5. Write a C program to simulate the following contiguous memory allocation Techniques a) Worst fit b) Best fit c) First fit.
6. Simulate all page replacement algorithms a)FIFO b) LRU c) OPTIMAL
7. Simulate all File Organization Techniques a) Single level directory b) Two level directory
8. Simulate all file allocation strategies a) Sequential b) Indexed c) Linked.
9. Write a script to translate the string from capital letters to small and small letters to capital using awk command.
10. Write a script to do the sorting of given numbers (use command line argument).
11. Write a program for process creation using C. (Use of gcc compiler).
12. Design a boot loader - to load a particular OS say TinyOS/ KolibriOS image - code to access from BIOS to loading the OS - involves little assembly code may use QEMU/virtual machines for emulation of hardware. 3hours
13. Allocate/free memory to processes in whole pages, find max allocatable pages, incorporate address translation into the program. 3hours
14. Create an interrupt to handle a system call and continue the previously running process after servicing the interrupt. 3hours
15. Write a Disk driver for the SATA interface. Take care to check readiness of the controller, locked buffer cache, accept interrupts from OS during the period, interrupting the OS again once done and clearing buffers. 3hours
16. Demonstrate the use of locks in conjunction with the IDE driver. 3hours
17. Run an experiment to determine the context switch time from one process to another and one kernel thread to another. Compare the findings 3hours
18. Determine the latency of individual integer access times in main memory, L1 Cache and L2 Cache. Plot the results in log of memory accessed vs average latency. 3hours
19. Compare the overhead of a system call with a procedure call. What is the cost of a minimal system call? 3hours
20. Compare the task creation times. Execute a process and kernel thread, determine the time taken to create and run the threads. 3hours
21. Determine the file read time for sequential and random access based of varying sizes of the files. Take care not to read from cached data - used the rawdevice interface. Draw a graph log/log plot of size of file vs average per-block time.
22. Study of Basic commands to understand the system and working of Linux
23. Write a script to reverse a number and string given by user.
24. Write a script to find the smallest of three numbers as well as largest among three numbers.
25. Write script that prints names of all sub directories present in the current directory.
26. Write a script to reverse the contents of a file.
27. Write a script to check entered string or a number is palindrome or not

28. Write a menu driven shell script for Copy a file, remove a file, Move a file in Linux.
29. Shell Script to make a menu driven calculator using case in UNIX / Linux /Ubuntu.
30. Write a script to display the digits which are in odd position in a given 6 digit number in Linux

BIBLIOGRAPHY

Text Book:

1. Silberschatz, Galvin and Gagne, "Operating Systems Concepts", Wiley, Ninth Edition, 2013.

Reference Book

1. Modern Operating Systems, Andrew S. Tanenbaum, Herbert Bos, Pearson Education India; Fourth edition 2016. ISBN-13:978- 9332575776
2. Operating Systems: A Design-Oriented Approach, Charles Crowley, International edition, McGraw-Hill Education (ISE Editions). ISBN-13 978 0071144629
3. Operating Systems: Internals and Design Principles William Stallings, Pearson Education India; 7 edition (2013). ISBN-13: 978-9332518803
4. D M Dhamdhere, "Operating Systems: A Concept based Approach", McGraw Hill Education, edition, 2012

Supplementary Resources:

1. http://williamstallings.com/OS/Animation/Animations.html
2. http://codex.cs.yale.edu/avi/os-book/OS9/slide-dir

MOOC/ NPTEL Courses:

1. http://nptel.ac.in/courses/106106144/
2. http://nptel.ac.in/courses/106108101/