4CP41: ADVANCED OPERATING SYSTEMS
CREDITS - 4 (LTP: 3, 0, 1)

Course Objective:

To impart knowledge of operating system design issues and the latest trends in advanced operating systems

Teaching and Evaluation Scheme:

Teaching Scheme (Hours per Week)			Credits	Evaluation Scheme				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE	CE	ESE	CE	
3	0	2	4	60	40	20	30	150

Course Contents:

Unit No	Topics	Teaching Hours
1	**Introduction**: Block diagram of Unix/Linux kernel, Kernel & its Data structures, Design Structures, Consistency of global data structures	**6**
2	**File System Implementations** : Buffer cache, File System related system calls and algorithms, Unix/Linux file system implementation, Issues related to file system performance, Vnode/Vfs architecture	**10**
3	**Process**: States & transitions, Context, Creation & termination, System boot & init, Unix/Linux scheduling algorithm, Clocks, System calls & algorithms. Signals and Session Management : Signal generation and handling, Unreliable signals, Reliable signals, Signals in SVR4, Signals implementation, Exceptions	**11**
4	**Memory management**: Swapping, Demand paging, Hybrid systems.	**6**
5	**I/O Subsystems**: Driver interfaces, Disk drivers, Terminal Drivers, Streams.	**3**
6	**Inter-process communication**: Shared memory and message passing mechanisms	**3**
7	**Multi Processor Systems**: Design concepts of Multi-processor Operating systems, Scheduling algorithms and Case study. **Distributed Operating Systems**: System Architectures, Design issues, Distributed scheduling, Distributed file systems	**6**
	Total	**45**

List of References:

1. Maurice J. Back, “The design of the Unix operating system”, PHI
2. Uresh Vahalia, “UNIX Internals”, Pearson Education
3. Stevens, “Advanced Programming in the UNIX Environment”, Addison Wesley
4. Tannenbaum, “Modern operating Systems”, PHI
5. Taunenbaum, “Distributed Systems: Principles and Paradigms”, PHI

Course Outcomes (COs):

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

1. Understand different design structures of operating systems.
2. Implement OS commands using system calls.
3. Design own Scheduling algorithms, Memory management algorithms and File system management algorithms.
4. Do advanced operating system programming.
5. Develop kernel modules and insert into Kernel.
6. Differentiate scheduling algorithms used for uniprocessor operating systems and multiprocessor operating systems.