ITE2002	Operating Systems	L	T	P	J	C
		3	0	2	0	4
Pre-requisite	**ITE1004**	**Syllabus version**				
		1.0				

Course Objectives:

- To impart major operating system components and its design principles.
- To provide an in-depth exposure to process, memory, device and file management techniques.
- To impart knowledge on various security challenges related to operating systems.
- To design applications for PC based operating systems and mobile based operating systems.

Expected Course Outcome:

1) Demonstrate the knowledge on fundamental concepts of operating systems.
2) Analyse and provide solution to process management.
3) Develop solution for process synchronization in multiprocessing system and handle deadlocks
4) Apply methods to support and manage main memory, virtual memory and secondary memory
5) Use and apply file access, file mounting and file allocation concepts.
6) Analyse disk management concepts.
7) Develop applications targeted for windows and mobile operating systems.
8) Develop and implement the various OS concepts in Linux operating system.

Student Learning Outcomes (SLO):	**2, 5,17**
[2]	Having a clear understanding of the subject related concepts and of contemporary issues
[5]	Having design thinking capability
[17]	Having an ability to use techniques, skills and modern engineering tools necessary for engineering practice

Module:1	**Fundamentals**	**5 hours**

Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Operating-System Services. User and Operating-System Interface, System Calls, Types of System Calls, System Programs.

Module:2	**Process and Thread Management Basics**	**7 hours**

Process Concept, Process Scheduling, Operations on Processes, Inter-process communication, Multicore Programming, Multithreading Models.

Scheduling:

Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

Module:3	**Mutual Exclusion**	**7 hours**
The Critical-Section Problem, Peterson's Solution, Semaphores, Classic Problems of Synchronization. Deadlock : Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection.		
Module:4	**Main Memory, virtual and Secondary storage Management**	**7 hours**
Swapping, Contiguous Memory Allocation. Segmentation, Paging, Structure of the Page Table Demand Paging, Page Replacement, Allocation of Frames, Thrashing.		
Module:5	**File Systems**	**7 hours**
File Concept, Access Methods, File-System Mounting, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods.		
Module:6	**Disk Management**	**4 hours**
Disk Structure, Disk Attachment, Disk Scheduling.		
Module:7	**Windows Operating System**	**6 hours**
History, Design Principles, System Components, Terminal Services and Fast User Switching, File System, Networking, Programmer Interface Mobile operating system –An introduction to Android and its versions, iOS, Windows Phone.		
Module:8	**Contemporary issues:**	**2 hours**
	Total Lecture hours:	**45 hours**

Text Book(s)	
1.	Silberschatz, P.B. Galvin & G. Gagne, Operating System Concepts, John Wiley, Ninth Edition, 2013.
Reference Books	
1.	William Stallings, Operating Systems – Internals and Design Principles, Seventh Edition, Prentice Hall, 2011.
List of Challenging Experiments (Indicative)	
1.	**Shell programming** a. Identify the command to print the home directory of each user. b. Develop an interactive grep script that asks for a word and a file name and then finds the number of occurrences of that word in the file. c. Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else. d. Write a shell script that determines the period for which a specified user is working

	on the system. e. Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on. f. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
2.	**Program to illustrate various methods for process and thread handling** a. Assume that you have given a complex program that contains large number of instructions. The program takes more time to execute if it is executed as a single thread of execution. Analyze the role of the system calls given below and restructure the program using it, so that the execution time of the program can be minimized considerably. Fork(), exec(), getpid(), exit(), wait(), close(), stat(), opendir(), readdir(). b. Programs using the I/O system calls of UNIX operating system (open, read, write, etc) c. Program to create processes, child processes and orphan process. d. Program to create a thread to find the factorial of a natural number n. e. The Collatz conjecture concerns what happens when we take any positive integer n and apply the following algorithm: n = n/2, if n is even n = 3 × n + 1, if n is odd The conjecture states that when this algorithm is continually applied, all positive integers will eventually reach 1. For example, if n = 35, the sequence is 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1.Write a C program using the fork () system call that generates this sequence in the child process. The starting number will be provided from the command line. For example, if 8 is passed as a parameter on the command line, the child process will output 8, 4, 2, 1. Because the parent and child processes have their own copies of the data, it will be necessary for the child to output the sequence. Have the parent invoke the wait () call to wait for the child process to complete before exiting the program. Perform necessary error checking to ensure that a positive integer is passed on the command line.
3.	a. Assume that two processes named client and server running in the system. It is required that these two processes should communicate with each other using shared memory concept. The server writes alphabets from a..z to the shared memory .the client should read the alphabets from the shared memory and convert it to A…Z. Write a program to demonstrate the above mentioned scenario. b. Design a program using ordinary pipes in which one process sends a string message to a second process, and the second process reverses the case of each character in the message and sends it back to the first process. For example, if the first process sends the message Hi There, the second process will return hI tHERE. This will require using two pipes, one for sending the original message from the first to the second process and the

	other for sending the modified message from the second to the first process. You can write this program using either UNIX or Windows pipes.
4.	Consider a corporate hospital where we have n number of patients waiting for consultation. The amount of time required to serve a patient may vary, say 10 to 30 minutes. If a patient arrives with an emergency, he /she should be attended immediately before other patients, which may increase the waiting time of other patients. If you are given this problem with the following algorithms how would you devise an effective scheduling so that it optimizes the overall performance such as minimizing the waiting time of all patients. [Single queue or multi-level queue can be used]. • Consider the availability of single and multiple doctors • Assign top priority for patients with emergency case, women, children, elders, and youngsters. • Patients coming for review may take less time than others. This can be taken into account while using SJF. a. FCFS b. SJF (primitive and non-pre-emptive)
5.	Apply the following algorithms for the above case and determine the variations in the resulting parameters. a. Priority b. Round robin.
6.	a. Write a program to calculate the below mentioned parameters and write your inference on implementing future knowledge algorithm [which starts scheduling only after fixed amount of time, even if processes have arrived]. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. [use non pre-emptive scheduling] Process Arrival Time Burst Time P1 0.0 8 P2 0.4 4 P3 1.0 1 b. Calculate the average turnaround time for these processes with the FCFS and SJF scheduling algorithm. c. The SJF algorithm is supposed to improve performance, but notice that we chose to run process P1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and then SJF scheduling is used. Remember that processes P1 and P2 are waiting during this idle time, so their waiting time may increase. [This type of algorithm is called as future knowledge algorithm]. d. Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context-switching overhead is 0.1 Milli second and that all processes are long-running tasks. Write a program to calculate the CPU utilization for a round-

	robin scheduler when: • The time quantum is 1 millisecond • The time quantum is 10 milliseconds
7.	Many CPU-scheduling algorithms are parameterized. For example, the RR algorithm requires a parameter to indicate the time slice. Multilevel feedback queues require parameters to define the number of queues, the scheduling algorithm for each queue, the criteria used to move processes between queues, and so on. These algorithms are thus really sets of algorithms (for example, the set of RR algorithms for all time slices, and so on). One set of algorithms may include another (for example, the FCFS algorithm is the RR algorithm with an infinite time quantum). What (if any) relation holds between the following pairs of algorithm sets? Implement the below mentioned algorithms for the data given below and determine the efficiency of each algorithm. 1. Priority and SJF 2. Multilevel feedback queues and FCFS 3. Priority and FCFS 4. RR and SJF
8.	a. Write a program to find the Fibonacci series using multi-threaded concept. b. Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers 90 81 78 95 79 72 85 The program will report The average value is 82 The minimum value is 72 The maximum value is 95 The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited.
9.	A pair of processes involved in exchanging a sequence of integers. The number of integers that can be produced and consumed at a time is limited to 100. Write a Program to implement the producer and consumer problem using POSIX semaphore for the above scenario.
10.	a. Write a Program to implement the solution for dining philosopher's problem. b. Servers can be designed to limit the number of open connections. For example, a server may wish to have only N socket connections at any point in time. As soon as N connections are made, the server will not accept another incoming connection until an existing connection is released. Write a program to illustrate how semaphores can be used by a server to limit the number of concurrent connections.
11.	a. Write a Program to implement banker's algorithm for Deadlock avoidance

	b. Consider the following snapshot of a system: Allocation Max A B C D A B C D P0 3 0 1 4 5 1 1 7 P1 2 2 1 0 3 2 1 1 P2 3 1 2 1 3 3 2 1 P3 0 5 1 0 4 6 1 2 P4 4 2 1 2 6 3 2 5 Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the processes may complete. Otherwise, illustrate why the state is unsafe. a. Available = (0, 3, 0, 1) b. Available = (1, 0, 0, 2)
12.	Consider a memory hole of size 1kb initially. When a sequence of memory request arrives as following, illustrate the memory allocation by various approaches and calculate the total amount memory wasted by external fragmentation and internal fragmentation in each approach. a. First fit; b. Best fit c. Worst fit
13.	Write a program to implement the page replacement algorithms. a. FIFO b. LRU c. OPT
14.	Write a program that implements the FIFO, LRU, and optimal pager replacement algorithms. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames can vary from 1 to 7. Assume that demand paging is used.
15.	Consider a file of size 1 MB. The size of a disk block is 512Bytes. Assume any number of available free blocks in the disk contiguously or non-contiguously. Implement the following algorithms to perform file allocation. Determine the efficiency of each file allocation strategies. a. Sequential b. Indexed c. Linked

		Total Laboratory Hours	30 hours
Recommended by Board of Studies	05-03-2016		
Approved by Academic Council	No. 40	Date	18-03-2016