MIT | Academy of Engineering (An Autonomous Institute)			COURSE SYLLABI (2016 - 2020)		
DEPARTMENT OF COMPUTER ENGINEERING			**W.E.F.**	:	2018-19
TY BTECH			**COURSE NAME**	:	Operating System
			COURSE CODE	:	CS301
			COURSE CREDITS	:	4
RELEASE DATE	:	**01/06/2018**	**REVISION NO.**	:	1.0

TEACHING SCHEME :		**EVALUATION SCHEME :**					
LECTURE	**PRACTICAL**	**THEORY**			**PRACTICAL**	**PRESENTATION/ DEMONSTRATION**	**TOTAL**
		ICE	**ECE**	**IA**			
3	2	30	40	30	25	25	125

PRE-REQUISITE:

1.CS 201-Data and File Structures

COURSE OBJECTIVES:

1. CS301.CEO.1: To describe the services of an operating system.
2. CS301.CEO.2: To introduce the concepts of a process, its life cycle and threads.
3. CS301.CEO.3: To explore inter process communication and CPU scheduling.
4. CS301.CEO.4: To understand memory management in operating system.
5. CS301.CEO.5: To understand the deadlock handling methods.
6. CS301.CEO.6: To discuss idea of file-system and its implementation

COURSE OUTCOMES:

The students after completion of the course will be able to

1. CS301.CO.1: State the basic principles of operating systems and its computational resources.(L1)
2. CS301.CO.2: Discuss various scheduling algorithm.(L2)
3. CS301.CO.3: Recognize deadlock to resolve the related issues.(L2)

4. CS301.CO.4: Solve or interpret problems regarding memory management.(L3)
5. CS301.CO.5: Analyze the efficiency of File System.(L4)

THEORY:

Unit I	**Introduction**	**8 Hours**

App/System/Case study: Linux Booting and Login Process

Content:
Basics of Operating Systems, Linux vs Windows, Abstract View of computer System Components, Types of Operating Systems, Functions of Operating System ,System Calls and its types. Booting and Shutting Down, Bootstrapping, Booting PCs, GRUB: The GRand Unified Boot loader, Booting to single-user mode. Working with Startup scripts, Rebooting and Shutting down.

Self -Study: Open Source Operating Systems-Fedora.

Further Reading: Special Purpose Systems.

Unit II	**Process and Threads**	**6 Hours**

App/System/Case study: Multitasking In Mobile Systems.

Contents: Process: Concept, Operation, Scheduling**,** Thread Overview: Multicore Programming, Multithreading Models, Thread Libraries, and Implicit Threading.

Self-Study: Threading Issues.

Further Reading: Operating System Generation & Debugging.

Unit III	**CPU Scheduling and Inter-Process Communication**	**6 Hours**

App/System/Case study: Multi process-Any Web Browser

Contents: Basic Concepts of CPU scheduling, Scheduling criteria, Scheduling Algorithm, Thread Scheduling, Multiple Processor Scheduling, Inter-process Communication, Shared-Memory Systems, Message-Passing Systems.

Self-study: Real Time Scheduling

Further Reading: Examples of IPC Systems.

Unit IV	**Process Synchronization and Deadlock**	**8 Hours**

App/System/Case study: Java Monitor.

Contents: Process Synchronization overview, The critical Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic problem of synchronization, Monitors, Deadlock, Methods for Handling Deadlocks, Modified Deadlock.

Self- Study: Synchronization examples

Further Reading: Alternative Approaches.

Unit V	**Memory Management**	**8 Hours**

App/System/Case study: ARM architecture.

Contents: Main memory-Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of page table, Virtual Memory-Demand Paging, Page Replacement, Allocation of frames, Thrashing Memory Mapped Files, Allocating Kernel Memory.

Self- Study: Linux memory management schemes.

Further Reading: Examples of Intel 32 and 64 bit Architecture.

Unit VI	**File and IO Management**	**8 Hours**

App/System/Case study: File locking in Java, Permission in UNIX.

File Concepts, Access Methods, Directory and Disk Structure, File System Mounting, File Sharing, Protection. File System Structure, File System Implementation, Directory Implementation, Allocation Methods, Free Space Management, Efficiency and Performance, Recovery, I/O Hardware, Application I/O Subsystem, Transforming I/O request to hardware operations.

Self -Study: WAFL File System.

Further reading: NFS

PRACTICAL: Perform following experiments using Open source tools

Practical No. 1	Title	Hours
1	Basic Shell Programming. (Basic System Calls and Shell Scripting)	**6 Hours**
2	Implement Process scheduling algorithm in C/C++/Java for following algorithm 1.FCFS,2.SJF,3.Round Robin, Priority based algorithm	**4 Hours**
3	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 Hello Friends, the second process will return hELLO fRIENDS.	**4 Hours**
4	In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes	**4 Hours**

	come and go, and new resources are bought and added to the system. If deadlock is controlled by the banker's algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances? a. Increase Available (new resources added). b. Decrease Available (resource permanently removed from system).	
5	Implement Page Replacement for following algorithm 1.LRU, 2.FIFO, 3.Optimal	**4 Hours**
6	Write a LINUX/UNIX C++ program to simulate the following file organization techniques: a) Single level directory b) Two level directory c) Hierarchical.	**6 Hours**
Mini projects	The Course Mini Project work will be started in Semester V. The work of the mini projects will be starting at beginning of term in alignment with laboratory assignments. It may be done by a groups of 3 students. However if project is done in groups, each student will be given a responsibility for a distinct module and the progress of individual modules is independent of others and performance of individual modules will be tracked periodically. The final evaluation will be done at the end of term through presentation, project demonstration and report.	**8 Hours**

TEXT BOOK:

1. Silberschatz, Galvin, Gagne, Operating System Concepts: International Student Version, 9th Edition, Paperback: 992 pages Publisher: Wiley; Eighth edition (20 April 2009) Paperback – 20 Apr 2009, Language: English, ISBN-10: 8126520515, ISBN-13: 978-8126520510.

2. Tanenbaum, Modern Operating Systems, 4th Edition, Paperback: 1136 pages, Publisher: Pearson Education India; Fourth edition (31 August 2016), Language: English, ISBN-10: 9332575770, ISBN-13: 978-9332575776

REFERENCES:

1. Gary Nutt, Operating Systems, 3rd Edition, Publisher: Pearson Education Singapore Pvt. Ltd (2004), ISBN-10: 8131723593, ISBN-13: 978-8131723593, ASIN: B007YTM00I.

2. Ann McHoes and Ida M. Flynn, Understanding Operating Systems, 6th edition, Mendel Rosenblum and John K. Ousterhout, Paperback: 590 pages, Publisher: Cengage; 6 edition (1 December 2013), Language: English, ISBN-10: 8131521567, ISBN-13: 978-8131521564.

3. Tanenbaum, Operating Systems Design and Implementation, Paperback: 1080 pages, Publisher: Pearson Education India; 3 edition (2015), Language: English, ISBN-10: 9332550514,ISBN-13: 978-9332550513.

4. Deitel, Operating System, 3rd Edition, Paperback: 1270 pages, Publisher: Pearson Education India; edition (2007), Language: English, ISBN-10: 8131712893, ISBN-13: 978-8131712894.