EVALUATION METHOD TO BE USED:

Continuous assessment	Mid term	End Semester
15(T) + 25 (P)	20	40

CO - PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓		✓	✓				✓		✓	✓
CO2	✓	✓	✓	✓	✓				✓			✓
CO3	✓	✓	✓	✓	✓				✓			✓
CO4	✓	✓	✓	✓	✓				✓			✓
CO5	✓	✓	✓	✓	✓				✓			✓
CO6	✓	✓										✓

CS 6108 OPERATING SYSTEMS

Prerequisites for the course: None

OBJECTIVES:

- To learn the basic concepts and functions of operating systems
- To learn the mechanisms of operating systems to handle processes and threads and their communication
- To know the components and management aspects of concurrency management
- To study the basic components of scheduling mechanism
- To learn the mechanisms involved in memory management in contemporary OS
- To appreciate the emerging trends in Operating Systems
- To learn programmatically to implement simple OS mechanisms

OPERATING SYSTEMS	L	T	P	EL	TOTAL	CREDITS
	3	0	4	3		6

MODULE I INTRODUCTION TO OPERATING SYSTEMS	L	T	P	EL
	4	0	4	4

Introduction to OS – Operating System Services – Operating System Operations – Virtualization – User and Operating System Interface – System Calls – Operating System Structures - Building and Booting an Operating System

SUGGESTED ACTIVITIES:
PRACTICAL:
l - Shell programming assignments

EL

1. Shell programming
2. Read the history of Unix/Linux/Windows
3. Know the operating system in your phone/laptop
4. System boot up process of Windows / Linux

SUGGESTED EVALUATION METHODS:

- Quiz on understanding of Linux and shell programming

MODULE II INTRODUCTION TO PROCESSES	**L**	**T**	**P**	**EL**
	6	**0**	**8**	**6**

Process Concept – Process Scheduling – Context Switch – Operations on Processes – Inter-process Communication – IPC in Shared-Memory Systems – IPC in Message-Passing Systems Examples of IPC Systems – POSIX shared memory

SUGGESTED ACTIVITIES :

Practical:

1. Use of ps, ps lx, ps tree, ps –aux commands
2. Use of top command to display resource usage statistics of processes
3. Use of the fork, clone, exec, wait, exit system calls
4. Inter-process communication using pipes, shared memory

EL: Learn to write a makefile, to use gdb and to use grep

SUGGESTED EVALUATION METHODS:

- Implementation evaluation
- EL assignment to be appropriately evaluated

MODULE III THREADS	**L**	**T**	**P**	**EL**
	3	**0**	**4**	**3**

Threads – Overview – Multithreading models – Pthreads

SUGGESTED ACTIVITIES :

Practical:

Implement multi-threading using the Pthread library

EL: Java threads

SUGGESTED EVALUATION METHODS:

- Evaluation of the implementation of multi-threading

MODULE IV CPU SCHEDULING	**L**	**T**	**P**	**EL**
	3	**0**	**4**	**3**

Basic Concepts of CPU Scheduling – Scheduling Criteria – Scheduling Algorithms

SUGGESTED ACTIVITIES :

Practical:

Simulation of CPU scheduling algorithms

EL:

Assignment problems on CPU scheduling algorithms

SUGGESTED EVALUATION METHODS:

- Assignments to be appropriately evaluated.

MODULE V PROCESS SYNCHRONIZATION	**L**	**T**	**P**	**EL**
	6	**0**	**8**	**6**
The Critical-Section Problem - Peterson's Solution – Hardware Support for Synchronization – Mutex Locks – Semaphores – Monitors				

SUGGESTED ACTIVITIES:

Practical:

1. Solutions to Syncronization problems using semaphores
2. Introduction to xv6: download and build
3. Run the kernel inside QEMU gdb

EL:

Reading details about xv6 operating system

SUGGESTED EVALUATION METHODS:

- Implementation evaluation
- Quiz on the understanding of the different concepts in this module

MODULE V STORAGE MANAGEMENT	**L**	**T**	**P**	**EL**
	4	**0**	**4**	**4**
File Concept – Access Methods – Directory Structure – Protection – Directory Implementation – Allocation Methods – Free-Space Management – Disk Structure – Disk Scheduling				

SUGGESTED ACTIVITIES:

Practical:

1. Use of system calls like creat, open, read, write, close, dup, readdir and scandir
2. Read the file xv6/fs.h to understand how a directory entry, a superblock and the contents of an inode are implemented in xv6
3. Read the file xv6/fs.c to understand how a new entry is added to a directory and explain the functions involved.

EL:

Read about the contents of a superblock, a directory entry, and an inode in UNIX-like operating systems

SUGGESTED EVALUATION METHODS:

- Quizzes

MODULE VI MEMORY MANAGEMENT	**L**	**T**	**P**	**EL**
Contiguous Memory Allocation – Paging – Structure of the Page Table – Segmentation – Paging with segmentation	**6**	**0**	**8**	**6**

SUGGESTED ACTIVITIES :

Practical:

1. Read and understand appropriate files in xv6 related to process scheduling and memory management

EL:

Assignment problems on memory management

SUGGESTED EVALUATION METHODS:

- Quiz on xv6 study files

MODULE VII VIRTUAL MEMORY MANAGEMENT	L	T	P	EL
	3	0	4	3
Demand Paging – Page Replacement – Allocation of Frames – Thrashing				

SUGGESTED ACTIVITIES

Practical:

- Implementation of at least one of the page replacement policies
- Implementation of a new system call in xv6

EL:

- Assignments on page replacement algorithms

SUGGESTED EVALUATION METHODS

- Evaluation of the coding assignments
- Quiz on the different parts of the module

OUTCOMES:

Upon completion of the course, the students will be able to:

- Articulate the main concepts, key ideas, strengths and limitations of Operating Systems
- Analyze the structure and basic architectural components of OS
- Elaborate and design various scheduling algorithms
- Discuss various memory management schemes and design them
- Point out the various aspects of storage management

TEXT BOOK:

1. Abraham Silberschatz, Greg Gagne and Peter B. Galvin. "Operating System Concepts", 10th Edition, Wiley, 2018.

REFERENCES:

1. Andrew S. Tanenbaum. "Modern Operating Systems", Addison Wesley, Fourth Edition, 2014.
2. D. M. Dhamdhere. "Operating Systems: A Concept-Based Approach", 3rd. Edition, Tata McGraw-Hill, 2017.
3. William Stallings. "Operating Systems: Internals and Design Principles", 7th Edition, Prentice Hall, 2013.
4. Russ Cox, Frans Kaashoek and Robert Morris. "xv6: A Simple, Unix-like Teaching Operating System", Revision 8. (Free and can be downloaded)

SOURCE CODE

The xv6 source code is available via : *git clone git://pdos.csail.mit.edu/xv6/xv6.git*

EVALUATION METHOD TO BE USED:

Category of Course	Continuous Assessment	Mid – Semester Assessment	End Semester
Theory Integrated with Practical	**15(T) + 25 (P)**	**20**	**40**

CO - PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓					✓				✓	✓
CO2	✓	✓		✓	✓	✓					✓	✓
CO3	✓	✓	✓	✓	✓		✓				✓	✓
CO4	✓	✓	✓	✓	✓	✓					✓	✓
CO5	✓	✓				✓	✓				✓	✓

CS 6109 COMPILER DESIGN

OBJECTIVES:

- To know about the various transformations in the different phases of the compiler, error handling and means of implementing the phases
- To learn about the techniques for tokenization and parsing
- To understand the ways of converting a source language to intermediate representation
- To have an idea about the different ways of generating assembly code
- To have a brief understanding about the various code optimization techniques

COMPILER DESIGN	L	T	P	EL	CREDITS
	3	0	4	3	6

MODULE I :	L	T	P	EL
	3	0	4	3
Phases of the compiler – compiler construction tools – role of assemblers, macroprocessors, loaders, linkers.				
SUGGESTED ACTIVITIES : • EL – Constructs of programming languages - C, C++, Java • LEX tool tutorial				
SUGGESTED EVALUATION METHODS: • Tutorial problems				