MANIPAL UNIVERSITY JAIPUR

School of Computer Science & Engineering

Department of Artificial Intelligence and Machine Learning (AIML)

Course Hand-Out

A. BASIC DETAILS:

Programme Name:	BTech AI&ML
Course Name:	Operating System Lab
Course Code:	AIM2231
LTPC (Lecture Tutorial Practical Credits):	0 0 2 1
Session:	Jan 25 – May 25
Class:	B.Tech. 2nd Year IV Semester
Course Coordinator:	Dr. Uddalak Chatterjee
Course Instructor(s):	Ms Simran, Dr Upendra Singh, Dr Shail Saharan, Dr. Varun Tiwari, Ms Surabhi Agrawal, Dr. Harish Kumar Shakya, Ms. Stuti Pandey.
Additional Practitioner(s) (Industry Fellow/ Visiting Faculty/ Adjunct Faculty, etc.):	To be identified and appointed later

B. INTRODUCTION:

The objective of Operating System lab is to provide students practical knowledge of Unix Commands, various scheduling, page replacement, deadlock handling, etc. algorithms and to familiarize the students with the fundamental concepts, methods, and implementation details of operating systems. Participation in this course will enable students to compare the working behavior and functions of different operating systems.

C. COURSE OUTCOMES:

Course outcomes for the Operating System Lab (AIM2231) course focus on students being able to understand and apply fundamental concepts of operating systems, including process management, memory management, file systems, and concurrency control, through hands-on lab exercises. Students will gain practical experience in implementing and manipulating operating system functionalities through programming assignments and lab projects. They will be proficient in developing system-level software components and understanding their interactions within the OS environment. Here are some potential course outcomes for the Operating System Lab (AI1230) course:

AIM2231.1. Explain basic Unix commands and write shell Scripts.

AIM2231.2. Build Skills to develop system programs using file and process system calls.

AIM2231.3. Compare various algorithms used for process scheduling.

AIM2231.4. Describe concepts related to concurrency and achieve the same for cooperating processes, apply various deadlock handling strategies to solve resource allocation problems.

AIM2231.5. Evaluate the performance of different memory management techniques and page replacement algorithms.

Information about attainment levels:

Attainment (%)	Level
< 60 %	0
≥ 60% < 70%	1
≥ 70% < 80%	2
≥ 80	3

D. VISION, MISSION, PROGRAM EDUCATIONAL OBJECTIVES, PROGRAM OUTCOMES, PROGRAM SPECIFIC OUTCOMES

VISION:

To achieve Excellence in Computer Science and Engineering (Artificial Intelligence and Machine Learning) Education for Global Competency with Human Values.

MISSION:

1. Provide innovative Academic & Research Environments to develop competitive Engineers in the field of Artificial Intelligence and Machine Learning
2. Develop Problem-solving and Project Management Skills through Student Centric Activities and industry collaboration.
3. Nurture the Students with Social & Ethical Values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

1. Graduates will Demonstrate Professional Skills on Global Platform in Computer Science Engineering and Integrated Domains.
2. Graduates will Enhance Knowledge and Skills through Higher Studies and Lifelong Learning of New Computing Technologies
3. Graduates will Provide Innovative Solutions to Drive Societal Advancement through Technology Entrepreneurship and Start-up.

PROGRAM OUTCOMES (POS):

[PO.1]. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

[PO.2]. Identify, formulate, research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

[PO.3]. Design solutions for complex engineering 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.

[PO.4]. 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.

[PO.5]. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

[PO.6]. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

[PO.7]. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

[PO.8]. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practices.

[PO.9]. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

[PO.10]. Communicate effectively on complex engineering activities with the engineering 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.

[PO.11]. Demonstrate knowledge and understanding of the engineering 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.

[PO.12]. 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.

E. ASSESSMENT PLAN:

Criteria	Description	Maximum Marks
Internal Assessment (Summative)	Lab Record: 20 Lab Performance: 10 Lab Quiz: 10 Viva: 10 Attendance: 10	60
End Term Exam (Summative)	End Term Exam Lab Performance: 10 Written: 10 Viva: 20	40
	Total	100
Attendance (Formative)	A minimum of 75% Attendance is required to be maintained by a student to be qualified for taking up the End Semester examination. The allowance of 25% includes all types of leaves including medical leaves.	
Makeup Assignments (Formative)	As per the guidelines from the Academic Section	
Homework/ Home Assignment/ Activity Assignment (Formative)	As per the guidelines from the Academic Section	

F. SYLLABUS:

Basic Linux commands: Illustration of shell functions, wild cards, redirection, pipes, sequencing, grouping, background processing, command substitution, sub shells, Shell programming. System Calls: File and process, I/O Redirection, IPC using Pipe and Signals. PThread API: Multithreaded programs, Synchronization programs using PThreads and Semaphores, CPU Scheduling, Deadlock, Memory Management. Creating a Virtual Machine: Virtual Machine Files and Snapshots, Virtual Machine Cloning and Exporting.

G. TEXTBOOK:

1. W. R. Stevens and S. A. Rago, Advanced Programming in the UNIX Environment, (3e), Addison-Wesley, 2017.

REFERENCE BOOKS

2. S. Das, Unix Concepts and Applications, (4e), McGraw Hill, 2017.
3. K. A. Robbins and S. Robbins, Unix Systems Programming: Communication, Concurrency, and Threads, (2e), Prentice Hall, 2015.

H. LAB PLAN

Lab No.	Topics	Session Outcome	Mode of Delivery	Corresponding CO	Evaluation
1-3	Introduction- Linux Operating System, Unix Commands and Shell Scripts	Define basic terminology related to OS.	Lecture based teaching-learning	AIM2231.1	Continuous Evaluation End Term Examination
		List and demonstrate various basic Unix and shell script commands.	Individual learning/ self-study		
		Illustrate use of Unix and Shell scripts commands in writing shell scripts.	Inquiry based learning		
4	System Calls	Program writing using file system related system calls.	Lecture based teaching-learning	AIM2231.2	Continuous Evaluation End Term Examination
5-6	Process Control	Illustrate process creation and its termination. (Using fork and kill)	Lecture based teaching-learning	AIM2231.2	Continuous Evaluation End Term Examination
		Illustrate Inter-Process communication using pipes.	Expeditionary learning		
		Illustrate Zombie and Orphan Process.	Expeditionary learning		
7	Thread	Implementation of concept of Multi-Threading using PThread in Linux OS.	Expeditionary learning	AIM2231.2	Continuous Evaluation End Term Examination
8	Process Scheduling	Apply knowledge of CPU scheduling algorithms in Implementing various CPU Scheduling Algorithms viz. FCFS, SJF, Priority, and Round Robin.	Lecture based teaching-learning	AIM2231.3	Continuous Evaluation End Term Examination
9-10	Process Synchronization	Implementation of Reader-Writer, Producer-Consumer, Synchronization Problems using Semaphores	Learning through problem-solving	AIM2231.4	Continuous Evaluation End Term Examination
11	Deadlock	Apply Banker algorithm for Deadlock Avoidance.	Learning through problem-solving	AIM2231.4	Continuous Evaluation End Term

					Examination
12-13	Memory Management Policies	Illustration of Page Replacement Algorithms: FIFO, Optimal and LRU	Learning through problem-solving	AIM2231.5	Continuous Evaluation End Term Examination
		Illustration of memory allocation strategies: First Fit, Best Fit, Next Fit and Worst Fit	Learning through problem-solving		
12-13	Memory Management Policies	Illustrate Page Replacement Algorithms: FIFO, Optimal and LRU Illustrate memory allocation strategies: First Fit, Best Fit, Next Fit and Worst Fit	Learning through problem-solving	AIM2231.5	Continuous Evaluation / End-Term Examination

I. COURSE ARTICULATION MATRIX: (MAPPING OF COS WITH POS)

CO	Statement	CORRELATION WITH PROGRAM OUTCOMES												CORRELATION WITH PROGRAM SPECIFIC OUTCOMES	
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2
1	Explain basic Unix commands and write shell Scripts.	**3**				**2**				**1**			**2**	**2**	**2**
2	Build skills to develop system programs using file and process system calls and P.Thread API.	**2**		**2**		**3**				**1**			**2**	**2**	
3	Compare various algorithms used for process scheduling.	**2**		**2**						**1**	**1**	**1**	**2**	**1**	
4	Describe concepts related to concurrency and achieve the same for cooperating processes, apply various deadlock handling strategies to solve resource allocation problems.	**2**		**2**			**1**	**1**	**1**	**1**	**1**	**1**	**2**	**1**	
5	Evaluate the performance of different memory management techniques and page replacement algorithms.	**2**		**2**			**1**	**1**		**1**	**1**	**1**	**2**	**1**	

1- Low Correlation; 2- Moderate Correlation; 3- Substantial Correlation

J. LIST OF EXPERIMENTS:

1. Basic Linux Commands
2. Illustration of shell functions such as wild cards, redirection, pipes, sequencing, grouping, background processing, command substitution, and subshells.
3. Write shell scripts with the use of shell variables, command line arguments, interactive use of read, loops (for, while), and conditional statements (if else, case).
4. Write C programs using file system calls. i.e. open, read, write, close, lseek.
5. Write C programs using process system calls. i.e. fork, exec, dup, exit, pause, wait.
6. Write C programs for interprocess communication using pipe.
7. Write multithreaded programs using POSIX Thread API.
8. Write programs to implement CPU scheduling algorithms.
9. Write a program to show solutions to the producer-consumer problem and Readers-Writer Problem.
10. Write a program to implement the Bankers algorithm for deadlock avoidance.
11. Write a program to illustrate memory allocation strategies (First-Fit, Next-Fit, Best-Fit, and Worst-Fit).
12. Write a program to illustrate page replacement algorithms (FIFO, Optimal, and LRU).

K. GUIDELINES TO STUDENTS:

1. Equipment in the lab for the use of the student community. Students need to maintain proper decorum in the computer lab. Students must use the equipment with care. Any damage caused is punishable.
2. Students are required to carry their observation/program book with completed exercises while entering the lab.
3. Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab noticeboard.
4. The lab can be used in free time/lunch hours by the students who need to use the systems should get prior permission from the lab in charge.
5. Lab records need to be submitted on or before the date of submission.
6. Students are not supposed to use pen drives/CDs.

L. HOW TO WRITE PROGRAM IN THE LAB RECORD:

Students will need to write the program in the following format.

1- **Aim:** write the complete aim of the program to be developed.
2- **Source Code:** write complete source code with the name of the file in the middle of the sheet and the program should be written in proper indentation.
3- **Output:** write the complete output with a set of inputs entered by the user during execution.

Example Code :

1. **Aim**: Write a program in C to display “Hello World” on the computer Screen.

2 **Source Code :**

```
#include<stdio.h>

void main() {

      printf(“Hello World”)

 }
```

3. **Output:**

```
Hello World
```

M. LAB RECORD FORMAT:

1- Physics/Chemistry file.
2- For every lab (from 1-13), students need to perform one of the programs as per the list of experiments, get the output checked by the lab instructor take his/her signature in the observation notebook. Write that experiment and get it checked on the same day or in the next lab.
3- Every question in the lab record must start with a new page.
4- Every question will have as shown in the image.
   a. Aim
   b. Source Code in C
   c. On the plane side output will be there.