Operating Systems Laboratory

B.Tech. 5th Semester

Name	**:**
Roll Number	**:**
Department	**: Computer Science and Engineering**

Faculty of Engineering & Technology

Ramaiah University of Applied Sciences

Ramaiah University of Applied Sciences

Private University Established in Karnataka State by Act No. 15 of 2013

Faculty	Engineering & Technology
Programme	B. Tech. in Computer Science and Engineering
Year/Semester	2nd Year / 5th Semester
Name of the Laboratory	Operating Systems Laboratory
Laboratory Code	CSC306A

List of Experiments

1. Introduction to operating system- compiling and booting of Linux kernel
2. Programs using process management system calls
3. Programs using file management system calls
4. Programs based on multithreaded programming
5. Programs for process scheduling algorithms
6. Solution to producer consumer problem using Mutex and Semaphore
7. Solutions to Dining philosopher problem using Semaphore
8. Programs for deadlock avoidance algorithm
9. Programs for memory management algorithms

Name: ______________________________ Roll Number:______________________________

Index Sheet

No.	Lab Experiment	Performing the experiment (7)	Document (7)	Viva (6)	Total Marks (20)
1	Introduction to Operating Systems				
2	Programs using process management system calls				
3	Programs using file management system calls				
4	Programs based on multithreaded programming				
5	Programs for process scheduling algorithms				
6	Solution to Producer Consumer Problem using Semaphore and Mutex				
7	Solution to Dining Philosopher problem using Semaphore				
8	Programs for deadlock avoidance algorithm				
9	Programs for memory management algorithms				
10	Lab Internal Test conducted along the lines of SEE and valued for 50 Marks and reduced for 20 Marks				
	Total Marks				

Component 1 = Lab Internal Marks =

Signature of the Staff In-charge

Name: ______________________________ Roll Number:______________________________

Laboratory 1

Title of the Laboratory Exercise: Introduction to Operating Systems

1. Introduction and Purpose of Experiment

   Operating system provides a platform for execution several other programs in a computer system. This laboratory exercise will help the students to get familiar with the operating system by understanding its structure.

2. Aim and Objectives

   Aim

   - To understand Linux operating systems structure and services

   Objectives

   At the end of this lab, the student will be able to

   - Explain the structure of Linux operating system
   - Compile a new kernel image
   - Execute programs with the created kernel

3. Experimental Procedure

   Configure, compile and install the kernel

4. Question

   Create a new Linux kernel image for the given kernel version

5. Calculations/Computations/Algorithms
6. Presentation of Results
7. Analysis and Discussions
8. Conclusions
9. Comments

Name: ______________________________ Roll Number:______________________________

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 2

Title of the Laboratory Exercise: Programs using process management system calls

1. Introduction and Purpose of Experiment

   A system call is a programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. There are different types of system calls developed for various purposes. They are mainly classified as process management, file management, directory management. By solving the problems students will be able to apply process management system calls

   Aim and Objectives

   Aim

   - To develop programs involving process management system calls

   Objectives

   At the end of this lab, the student will be able to

   - Use different process management system calls
   - Apply different system calls wherever required
   - Create C programs using process management system calls

2. Experimental Procedure

   i. Analyse the problem statement
   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
   iii. Implement the algorithm in C language
   iv. Compile the C program
   v. Test the implemented program
   vi. Document the Results
   vii. Analyse and discuss the outcomes of your experiment

3. Questions

   Implement the following operations in C

Name: ______________________________ Roll Number:______________________________

Create 5 processes and distinguish parent and child processes. And also display process ID and its parent ID of all the created processes using process management system calls

4. Calculations/Computations/Algorithms

5. Presentation of Results

6. Analysis and Discussions

7. Conclusions

8. Comments

    1. Limitations of Experiments

    2. Limitations of Results

    3. Learning happened

    4. Recommendations

    4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 3

Title of the Laboratory Exercise: Programs using file management system calls

1. Introduction and Purpose of Experiment

   A system call is a programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. There are different types of system calls developed for various purposes. They are mainly classified as process management, file management, directory management. By solving the problems students will be able to apply file management system calls

   Aim and Objectives

   Aim

   - To develop programs involving file management system calls

   Objectives

   At the end of this lab, the student will be able to

   - Use different file management system calls
   - Apply different system calls wherever required
   - Create C programs using file management system calls

2. Experimental Procedure

   i. Analyse the problem statement
   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
   iii. Implement the algorithm in C language
   iv. Compile the C program
   v. Test the implemented program
   vi. Document the Results
   vii. Analyse and discuss the outcomes of your experiment

3. Questions

   Implement the following command in C

Name: ______________________________ Roll Number:______________________________

Implement copy command (cp) to copy a file content to other file using file management system calls

4. Calculations/Computations/Algorithms
5. Presentation of Results
6. Analysis and Discussions
7. Conclusions
8. Comments

   1. Limitations of Experiments

   2. Limitations of Results

   3. Learning happened

   4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 4

Title of the Laboratory Exercise: Programs based on multithreaded programming

1. Introduction and Purpose of Experiment

   Multithreading is the ability of a processor or a single core in a multi-core processor to execute multiple threads concurrently, supported by the operating system. By solving students will be able to manipulate multiple threads in a program.

2. Aim and Objectives

   Aim

   - To develop programs using multiple threads.

   Objectives

   At the end of this lab, the student will be able to

   - Identify multiple tasks
   - Use threads constructs for creating threads
   - Apply threads for different/multiple tasks

3. Experimental Procedure

   i. Analyse the problem statement

   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code

   iii. Implement the algorithm in C language

   iv. Compile the C program

   v. Test the implemented program

   vi. Document the Results

   vii. Analyse and discuss the outcomes of your experiment

4. Questions

   Create multithreaded programs to implement the following

Name: ____________________________ Roll Number:____________________________

a) Display “Hello World” message by 3 different threads

b) Display thread IDs by each thread

c) Create three threads;

- Thread1 add marks out of 10 from subject1 to subject5 of student 1, Thread2 adds from Subject1 to subject 5 of student 2 and Thread3 takes the sum from the Thread1 and Thread2 and decides who scored more marks. Display the total marks of the student (highest score) in parent process.
  Instructions: Use a global variable to keep track of the sum from Thread1 and Thread2 and update it. Thread3 needs to wait until both Thread1 and Thread2 are done. Finally, parent process needs to wait until Thread3 is done. Get the return value from Thread3, and print the return value.

5. Calculations/Computations/Algorithms
6. Presentation of Results
7. Analysis and Discussions
8. Conclusions
9. Comments

   1. Limitations of Experiments
   2. Limitations of Results
   3. Learning happened
   4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 5

Title of the Laboratory Exercise: Programs for process scheduling algorithms

1. Introduction and Purpose of Experiment

   A Process Scheduler schedules different processes to CPU based on particular scheduling algorithms. There are various scheduling algorithms present in each group of operating system. By solving these problems students will be able use different scheduling algorithms as part of their implementation

2. Aim and Objectives

   Aim

   - To develop programs to implement scheduling algorithms

   Objectives

   At the end of this lab, the student will be able to

   - Distinguish different scheduling algorithms
   - Apply the logic of scheduling algorithms wherever required
   - Create C programs to simulate scheduling algorithms

3. Experimental Procedure

   i. Analyse the problem statement

   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code

   iii. Implement the algorithm in C language

   iv. Compile the C program

   v. Test the implemented program

   vi. Document the Results

   vii. Analyse and discuss the outcomes of your experiment

4. Questions

Name: ____________________________ Roll Number:____________________________

Write program to simulate the following process scheduling algorithms. Calculate average waiting time and average turnaround time for processes under each scheduling algorithm.

Instructions: Assume all the processes arrive at the same time. For FCFS and SJF scheduling algorithms, read the number of processes/jobs in the system and their CPU burst times. For round robin scheduling algorithm, read the number of processes in the system, their CPU burst times and the size of the time slice. For priority scheduling algorithm, read the number of processes in the system, their CPU burst times and the priorities.

a) First Come First Served
b) Shortest Job First
c) Priority
d) Round Robin

5. Calculations/Computations/Algorithms
6. Presentation of Results
7. Analysis and Discussions
8. Conclusions
9. Comments

   1. Limitations of Experiments
   2. Limitations of Results
   3. Learning happened
   4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 6

Title of the Laboratory Exercise: Solution to Producer Consumer Problem using Semaphore and Mutex

1. Introduction and Purpose of Experiment

   In multitasking systems, simultaneous use of critical section by multiple processes leads to data inconsistency and several other concurrency issues. By solving this problem students will be able to use Semaphore and Mutex for synchronisation purpose in concurrent programs.

2. Aim and Objectives

   Aim

   - To implement producer consumer problem using Semaphore and Mutex

   Objectives

   At the end of this lab, the student will be able to

   - Use semaphore and Mutex
   - Apply semaphore and Mutex in the required context
   - Develop multithreaded programs with Semaphores and Mutex

3. Experimental Procedure

   i. Analyse the problem statement
   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
   iii. Implement the algorithm in C language
   iv. Compile the C program
   v. Test the implemented program
   vi. Document the Results
   vii. Analyse and discuss the outcomes of your experiment

4. Questions

   Implement producer consumer problem by using the following

   a) Semaphore

Name: ____________________________ Roll Number:____________________________

b) Mutex

5. Calculations/Computations/Algorithms

6. Presentation of Results

7. Analysis and Discussions

8. Conclusions

9. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 7

Title of the Laboratory Exercise: Programs for deadlock avoidance algorithm

1. Introduction and Purpose of Experiment

   Deadlocks can be avoided if certain information is available in advance. By solving these problems students will become familiar to avoid deadlock in advance with the available resource information

2. Aim and Objectives

   Aim

   - To develop Bankers algorithm for multiple resources for deadlock avoidance

   Objectives

   At the end of this lab, the student will be able to

   - Verify a problem to check that whether deadlock will happen or not for the given resources
   - Implement the bankers algorithm for multiple resources

3. Experimental Procedure

   i. Analyse the problem statement

   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code

   iii. Implement the algorithm in C language

   iv. Compile the C program

   v. Test the implemented program

   vi. Document the Results

   vii. Analyse and discuss the outcomes of your experiment

4. Questions

   Implement a Bankers algorithm for deadlock avoidance

5. Calculations/Computations/Algorithms

Name: ______________________________ Roll Number:______________________________

6. Presentation of Results

7. Analysis and Discussions

8. Conclusions

9. Comments

   1. Limitations of Experiments

   2. Limitations of Results

   3. Learning happened

   4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 8

Title of the Laboratory Exercise: Programs for memory management algorithms

1. Introduction and Purpose of Experiment

   In a multiprogramming system, the user part of memory must be further subdivided to accommodate multiple processes. This task of subdivision is carried out dynamically done by the operating system known as memory management. By solving these problems students will become familiar with the implementations of memory management algorithms in dynamic memory partitioning scheme of operating system.

2. Aim and Objectives

   Aim

   - To develop a simulator for memory management algorithms

   Objectives

   At the end of this lab, the student will be able to

   - Apply memory management algorithms wherever required
   - Develop simulators for the algorithms

3. Experimental Procedure

   - Analyse the problem statement
   - Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
   - Implement the algorithm in C language
   - Compile the C program
   - Test the implemented program
   - Document the Results
   - Analyse and discuss the outcomes of your experiment

4. Questions

Name: ______________________________ Roll Number:______________________________

Implement a simulator for the memory management algorithms with the provision of compaction and garbage collection

a) First fit

b) Best fit

c) Worst fit

5. Calculations/Computations/Algorithms

6. Presentation of Results

7. Analysis and Discussions

8. Conclusions

9. Comments

1. Limitations of Experiments

2. Limitations of Results

3. Learning happened

4. Recommendations

Name: ______________________________ Roll Number:______________________________

Laboratory 9

Title of the Laboratory Exercise: Solution to Dining Philosopher problem using Semaphore

1. Introduction and Purpose of Experiment

   In multitasking systems, simultaneous use of critical section by multiple processes leads to data inconsistency and several other concurrency issues. By solving this problem students will be able to use semaphore for synchronisation purpose in concurrent programs.

2. Aim and Objectives

   Aim

   - To develop concurrent programs using semaphores

   Objectives

   At the end of this lab, the student will be able to

   - Use semaphore
   - Apply appropriate semaphores in different contexts
   - Develop concurrent programs using semaphores

3. Experimental Procedure

   i. Analyse the problem statement

   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code

   iii. Implement the algorithm in C language

   iv. Compile the C program

   v. Test the implemented program

   vi. Document the Results

   vii. Analyse and discuss the outcomes of your experiment

4. Question

Name: _____________________________ Roll Number:_____________________________

Implement the Dining Philosopher problem using POSIX threads

5. Calculations/Computations/Algorithms

6. Presentation of Results

7. Analysis and Discussions

8. Conclusions

9. Comments

    1. Limitations of Experiments

    2. Limitations of Results

    3. Learning happened

    4. Recommendations