Operating System Lab
(PCC CS592)

Laboratory Instructor’s Manual

Last Revised

July, 2022

Dept. of CSE
Techno Main, Salt Lake

GENERAL INSTRUCTIONS FOR STUDENTS

1. Do not enter the Laboratory without prior permission.
2. Switch off your mobile phones during Lab class and maintainsilence.
3. Save your files only on the specific destination folders as instructed.
4. Do not play games, watch movies, chat or listen to music during the class.
5. Do not change desktop setting, screen saver or any other systemsettings.
6. Do not use any external storage device without prior permission.
7. Do not install any software without prior permission.
8. Do not browse any restricted, illegal or spamsites.

GENERAL ADDRESS FOR LABORATORY TEACHERS

1. Submission of documented lab reports related to completed lab assignments should be done during the following lab session.
2. The promptness of submission should be encouraged by way of marking and evaluation patterns as reflected in the lab rubric which eventually will benefit the students**.**

Program Outcomes (PO)

PO1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis:** Identify, formulate, research literature, and analyze engineering problemsto arrive at substantiated conclusions using first principles of mathematics, natural and engineering sciences.

PO3. **Design/Development of solutions:** Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety and the cultural societal and environmental considerations.

PO4. **Conduct investigations of complex problems:** Use research based knowledge including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage:** Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.

PO8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineeringpractice.

PO9. **Individual and team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

PO10. **Communications:** Communicate effectively with the engineering community and with the society at large. Be able to comprehend and write effective reports documentation. Make effective presentations and give and receive clear instructions.

PO11. **Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

PO12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Program Specific Outcomes (PSO)

PSO1: Abilityto develop the solutions for scientific, analytical and research-oriented problems in the area of Computer Science and Engineering.

PSO2: Ability to apply suitable programming skills integrated with professional competence to develop applications catering to the industrial and societal needs in the field of Computer Science and Engineering and its allied areas.

NAME OF THE PROGRAM: *CSE*	**DEGREE:** *B.Tech*
COURSE NAME: *Operating System*	**SEMESTER:** *5th*
COURSE CODE: *PCC-CS592*	**COURSE CREDIT:** *2*
COURSE TYPE: *LAB*	**CONTACT HOURS:** *4P*

SYLLABUS

1. Shell programming [6P]: creating a script, making a script executable, shell syntax (variables, conditions, control structures, functions, commands).

2. Process [6P]: starting new process, replacing a process image, duplicating a process image, waiting for a process, zombie process.

3. Signal [9P]: signal handling, sending signals, signal interface, signal sets.

4. Semaphore [6P]: programming with semaphores (use functions semctl, semget, semop, set_semvalue, del_semvalue, semaphore_p, semaphore_v).

5. POSIX Threads [9P]: programming with pthread functions (viz. pthread_create, pthread_join, pthread_exit, pthread_attr_init, pthread_cancel)

6. Inter-process communication [9P]: pipes (use functions pipe, popen, pclose), named pipes(FIFOs, accessing FIFO)

NAME OF THE PROGRAM: *CSE*	**DEGREE:** *B.Tech*
COURSE NAME: *Operating System Lab*	**SEMESTER:** *5th*
COURSE CODE: *PCC-CS592*	**COURSE CREDIT:** *2*
COURSE TYPE: *LAB*	**CONTACT HOURS:** *4P*

Course Outcomes (CO)

After this course students will be able to:

CO1: Experiment with various Linux commands and shell scripting.

CO2: Analyze various process management techniques using Unix/Linux system calls.

CO3: Apply different programming tools for writing and executing programs in an efficient way.

CO4: Perform effectively as an individual to solve different operating systems problems.

CO5: Prepare appropriate documentation containing detailed reports for all solved problems.

NAME OF THE PROGRAM: ***CSE***	**DEGREE:** ***B.Tech***
COURSE NAME: ***Operating System Lab***	**SEMESTER:** ***5th***
COURSE CODE: ***PCC-CS592***	**COURSE CREDIT:** ***2***
COURSE TYPE: ***LAB***	**CONTACT HOURS:** ***4P***

Exp. No.	**List of Experiments**	**Week**
1.	a) Write Shell script to find out Factorial of a given number. b) Write Shell script to determine a given year is leap year or not. c) Write Shell script to find out sum of digits of a given number. d) Write Shell script to generate Fibonacci series up to N-th term. e) Write a shell script to compute the Common Logarithm of an integer N. The value of N should be taken as command-line argument. **Note**: the resulting value should not contain more than two digits after the decimal point.	Week 1
2.	a) Write a shell script for Summation of N natural numbers, where the values of N natural numbers are given as command line arguments. b) Write a shell script that sorts an array of integer using any well- known sorting algorithm c) Write a shell script to check an input string is a valid user or not	Week 2
3.	a) Write a menu driven script to do the following: i) Show the permissions of a file or directory; ii) Show the number of files and directories under current directory separately; iii) Show the last modification time of a file in current directory; iv) Exit from menu. b) Write a menu driven shell script with three options: i) Accept your personal details along with a password and store them in a data file (e.g. personal.dat); ii) Display the personal details stored in the data file after successfully matching the password stored in the file; iii) Exit from menu. **Hint**: Use ls, wc, grep, cut, pipe (|) and redirection (<, >, >>).	Week 3
4.	a) Create a student file (Students.dat) containing four fields, Name, Roll, Phone and Address. The field separator is '|'. Now, write a menu driven script to do the following: (i) Sort the records in reverse order of Roll number and store the output in a new file (Sorted.dat). (ii) Replace lower case letters with the upper case letters in the file 'Sorted.dat'. Take file input using redirection. (iii) Write commands to find out the records with unique name in the file 'Sorted.dat'. Append the output records to 'Students.dat'. (iv) Display only the 2nd and 3rd lines from the above file (Students.dat). b) Create a marks file (Marks.dat) containing five fields, Name, Roll, Subject_1, Subject_2 and Subject_3. The field separator is '~'. Also, write a shell script to find the name of the student who has got the highest marks in total. The file name should be supplied as command line argument. Also, check for sufficient number of command line arguments passed and show appropriate error message.	Week 4

Exp. No.	List of Experiments	Week
5.	a) Write a C program to create a child process and print the PID & PPID from both the parent and child processes. Also verify the output of your program with the 'ps' command. b) Write a C program to create an Orphan process and verify the output of your program with the 'ps' command. c) Write a C program to create a Zombie process and verify the output of your program with the 'ps' command.	Week 5
6.	a) Implement IPC between parent and child process where parent will print a message received from the child, who will take the message as user input. Use unnamed pipe for IPC. b) Implement IPC between two processes where procwess-1 will take two strings as user input and send them to process- 2.Process-2 will compare them and print the result (SAME OR NOT SAME).Use FIFO for IPC.	Week 6
7.	a) Write a Program to demonstrate the use of signal. The process will print a message infinitely until an interrupt signal occurs. It will handle the signal and will print a message along with the signal number that it has got. b) Write a program to demonstrate the use of signal. Parent process will stop until an alarm received from child process. c) Write a Program to create a child process. The parent will send a signal to the child every 5 seconds and the child will handle the signal and check if an input number is a leap year or not.	Week 7
8.	Write a ‘C’ program in LINUX to create a thread that determines the summation of N natural numbers using POSIX thread.	Week 8
9.	a) WAP using semaphore which two process will synchronize each other to print ***baabbaabbaab*** pattern. b) WAP where a newly created thread will check and number is prime or not, which number is entered in main thread.	Week 9
10.	Simulate Producer-Consumer Problem using multi-threading and Semaphore.	Week 10
11.	Take one string as input from user in process P1 and store it in a shared memory. Process P2 will read the string from the shared memory and will check if it is a PALINDROME or not and return 1 (for Palindrome) or 0 (for NOT Palindrome) back to P1, which will print the result.	Week 11
12.	a) Simulate the Reader-Writer problem using multiple processes. b) Simulate the Dining Philosopher problem using multiple processes.	Week 12

NAME OF THE PROGRAM: *CSE*	**DEGREE:** *B.Tech*
COURSE NAME: *Operating System*	**SEMESTER: 5th**
COURSE CODE: *PCC-CS592*	**COURSE CREDIT:** *2*
COURSE TYPE: *LAB*	**CONTACT HOURS:** *4P*

Rubrics for Lab

Score **Criteria**	**Excellent (100%)**	**Good (80%)**	**Average (60%)**	**Poor (40%)**	**Absent (0%)**	**CO Mapping**	**PO PSO Mapping**
1. Lab Participation	Students are able to identify the problem/ analyze the problem/Design the solutions and solve the problem applying various algorithms with appropriate test cases; students are able to include boundary conditions in the test cases; students are able to modify the program or design as per requirement of the outcomes from boundary conditions (if any).	Students are able to identify the problem/ analyze the problem/Design the solutions and solve the problem applying various algorithms with appropriate test cases; students are able to include boundary conditions in the test cases.	Students are able to identify the problem/ analyze the problem/Design the solutions and solve the problem applying various algorithms with appropriate test cases.	Student is not able to understand/analyze/design the problem or interpret the problem into specified language		CO1, CO2	PO1, PO2, PSO1, PSO2

2. Effective utilization of the modern tools and their properties, compilers	Students are able to exploit the full potential of the tool/property/topic under consideration for the specified language	Students are able to exploit the important features of the tool/property/topic under consideration for the specified language	Students are able to use specified tool/property/topic as per the problem requirement only under consideration for the specified language	Students are not able to use tool/property/topic under consideration for the specified language		CO3	PO5
3. Individual or team work	Students are able to work effectively, sincerely and ethically as an individual or in a member of a team	Students are able to work ethically as an individual or in a member of a team	Students are able to work as an individual or in a member of a team	Students are not able to work effectively, sincerely and ethically as an individual or in a member of a team		CO4	PO9
4. Documentation	Students will prepare effective documentation of lab classes mentioning problem statement, input-output, appropriate test cases with boundary conditions	Students will prepare effective documentation of lab classes mentioning problem statement, input-output, test cases	Students will prepare effective documentation of lab classes mentioning problem statement, input-output	Students will not prepare effective documentation of lab classes mentioning objective, input-output, test cases, boundary conditions		CO5	PO10