MODEL ENGINEERING COLLEGE ERNAKULAM

CS 334 NETWORK PROGRAMMING LAB

CYCLE I

Expt No:1 **<date>** **LAB 1** **<4/2/2019>** **&** **<8/2/2019>**	**STUDY OF SYSTEM CALLS FOR OS PROGRAMMING** **AIM:** To study the system calls – create(), open(), read(), write(), close(), sleep(), exit(), unlink(), kill(), getpid(), getppid(), getuid(), getgid(), fork(), pipe(), fifo(), execl() **creat() system call** <Header files, syntax and description> **open() system call** <Header files, syntax and description> **read() system call** <Header files, syntax and description> **write() system call** <Header files, syntax and description> **close() system call** <Header files, syntax and description> **sleep() system call** <Header files, syntax and description> **exit() system call** <Header files, syntax and description> **unlink() system call** <Header files, syntax and description> **kill() system call** <Header files, syntax and description> **Program No: (i):** To get the process id, parent process id, real user id, real group id, effective user id, effective group id. <Header files, syntax and description of getpid(), getppid(), getuid(), getgid(), geteuid(), getegid()> Program, Execution Steps, Output **Program No: (ii):** Familiarization of fork() system call <Header files, syntax and description> Program, Execution Steps, Output

	Program No: (iii): Familiarization of pipe() system call <Header files, syntax and description Program, Execution Steps, Output **Program No: (iv):** To create a FIFO (named pipe) <Header files, syntax and description> Program, Execution Steps, Output **Program No: (v):** Familiarization of execl() system call <Header files, syntax and description> Program, Execution Steps, Output
Expt No:2 **<date>** **LAB 2** **<11/2/2019>** **&** **<15/2/2019>**	**FAMILIARISATION OF POSIX THREAD FUNCTIONS** **AIM:** To study the basic posix thread functions – pthread_create, pthread_join, pthread_self, pthread_detach, pthread_exit <Header files, syntax and description> Program, Execution steps, Output
Expt No:3 **<date>** **LAB 3** **<18/2/2019>** **&** **<25/2/2019>**	**INTERPROCESS COMMUNICATION USING PIPES** **AIM:** To implement interprocess communication using two pipes <Header files, syntax and description> Program, Execution steps, Diagram,Output
Expt No:4 **<date>** **LAB 3** **<18/2/2019>** **&** **<25/2/2019>**	**INTERPROCESS COMMUNICATION USING FIFO** **AIM:** To implement interprocess communication using fifo <Header files, syntax and description> Program, Execution steps, Output
Expt No:5 **<date>** **LAB 4** **<01/3/2019>** **&** **<11/3/2019>**	**INTERPROCESS COMMUNICATION USING POSIX MESSAGE QUEUES** **AIM:** To implement interprocess communication using Message Queues **.** <Header files, syntax and description> Program, Execution steps, Output

Expt No:6 **<date>** **LAB 5** **<18/3/2019>** **&** **<22/3/2019>**	**INTERPROCESS COMMUNICATION USING POSIX SHARED MEMORY** **AIM:** Write a program to create an integer variable using shared memory concept and increment the variable simultaneously by two processes. Use semaphores to avoid race conditions <Header files, syntax and description> Program, Execution steps, Output
Expt No: 7 **<date>** **LAB 6** **<25/3/2019>** **&** **<29/3/2019>**	**READERS-WRITERS PROBLEM** **AIM:** A Program to implement the Readers-Writers problem using Semaphores and shared memory <Header files, syntax, algorithm and description> Program, Execution steps, Output
	CYCLE II
Expt No:8 **<date>** **LAB 7** **<01/4/2019>** **&** **<05/4/2019>**	**STUDY OF SYSTEM CALLS FOR NETWORK PROGRAMMING** **AIM:** To study the system calls - Socket(), bind(), listen(), accept(), connect(), <Header files, syntax and description> Program, Execution Steps, Output
Expt No:9 **<date>** **LAB 8** **<08/4/2019>** **&** **<12/4/2019>**	**SOCKET PROGRAMMING USING TCP** Implement client server communication using socket programming and TCP as transport layer protocol <Header files, syntax,and description> Program, Execution steps, Output
Expt No:10 **<date>** **LAB 8** **<08/4/2019>** **&** **<12/4/2019>**	**SOCKET PROGRAMMING USING UDP** Implement client server communication using socket programming and UDP as transport layer protocol <Header files, syntax, and description> Program, Execution steps, Output

Expt No:11 **<date>** **LAB 9** **<22/4/2019>** **&** **<26/4/2019>**	**MULTICLIENT CHAT SERVER USING TCP** Implement a multi client chat server using TCP as transport layer protocol <Header files, syntax, and description> Program, Execution steps, Output
Expt No:12 **<date>** **LAB 10** **<22/4/2019>** **&** **<26/4/2019>**	**SIMPLE MAIL TRANSFER PROTOCOL** Implement a simple mail transfer protocol <Header files, syntax, and description> Program, Execution steps, Output