

PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE – 43.

SCHEDULE OF LAB EXPERIMENTS

ACADEMIC YEAR: 2019-20

DEPARTMENT: INFORMATION TECHNOLOGY

Date: 15/06/19

CLASS: T.E.

SEMESTER: I

SUBJECT: SOFTWARE LABORATORY - II

Lab Expt. No.	Problem Definition/Statement	Last Date for Completion
A	Basic Linux Commands (Content Beyond Syllabus)	3 rd Week of June
1.	Shell programming Write a program to implement an address book with options given below: a) Create address book. b) View address book. c) Insert a record. d) Delete a record. e) Modify a record. f) Exit.	1 st Week of July
2.	Process control system calls: The demonstration of FORK, EXECVE and WAIT system calls along with zombie and orphan states. a. Implement the C program in which main program accepts the integers to be sorted. Main program uses the FORK system call to create a new process called a child process. Parent process sorts the integers using sorting algorithm and waits for child process using WAIT system call to sort the integers using any sorting algorithm. Also demonstrate zombie and orphan states. b. Implement the C program in which main program accepts an integer array. Main program uses the FORK system call to create a new process called a child process. Parent process sorts an integer array and passes the sorted array to child process through the command line arguments of EXECVE system call. The child process uses EXECVE system call to load new program that uses this sorted array for performing the binary search to search the particular item in the array.	3 rd Week of July
3.	Implement multithreading for Matrix Multiplication using pthreads	4 th Week of July

4.	Thread synchronization using counting semaphores. Application to demonstrate producer-consumer problem with counting semaphores and mutex.	1 st Week of Aug
5.	Thread synchronization and mutual exclusion using mutex. Application to demonstrate Reader-Writer problem with reader priority.	2 nd Week of Aug
6.	Deadlock Avoidance Using Semaphores: Implement the deadlock-free solution to Dining Philosophers problem to illustrate the problem of deadlock and/or starvation that can occur when many synchronized threads are competing for limited resources.	3 rd Week of Aug
7.	Inter-process communication in Linux using following. a. Pipes: Full duplex communication between parent and child processes. Parent process writes a pathname of a file (the contents of the file are desired) on one pipe to be read by child process and child process writes the contents of the file on second pipe to be read by parent process and displays on standard output. b. FIFOs: Full duplex communication between two independent processes. First process accepts sentences and writes on one pipe to be read by second process and second process counts number of characters, number of words and number of lines in accepted sentences, writes this output in a text file and writes the contents of the file on second pipe to be read by first process and displays on standard output.	4 th Week of Feb
8.	Inter-process Communication using Shared Memory using System V. Application to demonstrate: Client and Server Programs in which server process creates a shared memory segment and writes the message to the shared memory segment. Client process reads the message from the shared memory segment and displays it to the screen.	2 nd Week of Sept
9.	Implement an assignment using File Handling System Calls (Low level system calls like open, read, write, etc).	3 rd Week of Sept
10.	Implement a new system call in the kernel space, add this new system call in the Linux kernel by the compilation of this kernel (any kernel source, any architecture and any Linux kernel distribution) and demonstrate the use of this embedded system call using C program in user space.	4 th Week of Sept

Prof. V. R. Jaiswal
(Subject Coordinator)

Head of Department
(Information Technology)