1. 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

2. Study of Basic Linux Commands: echo, ls, read, cat, touch, test, loops, arithmetic comparison, conditional loops, grep, sed etc.

3. 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.

4. Implement the C program in which main program accepts an array. Main program uses the FORK system call to create a new process called a child process. Parent process sorts an 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 which display array in reverse order.

5. Implement the C program for CPU Scheduling Algorithms: Shortest Job First (Preemptive) with different arrival time.

6. Implement the C program for CPU Scheduling Algorithms: Round Robin with different arrival time.

7. Thread synchronization using counting semaphores. Application to demonstrate: producer- consumer problem with counting semaphores and mutex.

8. Thread synchronization and mutual exclusion using mutex. Application to demonstrate: Reader- Writer problem with reader priority.

9. Implement the C program for Deadlock Avoidance Algorithm: Bankers Algorithm.

10. Implement the C program for Page Replacement Algorithm: FCFS for frame size as minimum three.

11. Implement the C program for Page Replacement Algorithm: LRU for frame size as minimum three.

12. Implement the C program for Page Replacement Algorithm: Optimal for frame size as minimum three.

13. **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.

14. **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.

15. Implement the C program for Disk Scheduling Algorithms: SSTF considering the initial head position moving away from the spindle.

16. Implement the C program for Disk Scheduling Algorithms: SCAN considering the initial head position moving away from the spindle.

17. Implement the C program for Disk Scheduling Algorithms: C-Look considering the initial head position moving away from the spindle.