OPERATING SYSTEMS & INTERNALS LAB

Course Code: 20CS1103

L	T	P	C
0	0	3	1.5

COURSE OUTCOMES:

At the end of the Course the student shall be able to

CO 1: Analyse CPU scheduling algorithms. (L4)

CO 2: Implement system calls and process coordination. (L3)

CO 3: Distinguish various memory management techniques. (L4)

CO 4: Implement different File System techniques. (L3)

CO 5: Analyse Disk Scheduling algorithms. (L4)

LIST OF EXPERIMENTS:

(Any 12 experiments of the following to be performed)

1. Understanding and practical exposure towards Basic Linux commands.
2. Collect the basic information about your machine using proc in Linux.
3. Implementation of write () and read () system calls.
4. Implementation of open (), fork () system calls.
5. Implement a program using fork () system call to create a hierarchy of 3 process such that P2 is the child of P1 and P1 is the child of P.
6. Implement the following:
   i) Program to create an Orphan process.
   ii) Create two child process C1 and C2. Make sure that only C2 becomes an Orphan process.
7. Implement the following:
   i) Program to create threads in Linux. Thread prints 0-4 while the main process prints 20-24
   ii) Program to create a thread. The thread prints numbers from zero to n, where value of n is passed from the main process to the thread. The main process also waits for the thread to finish first and then prints from 20-24.

8. Implement non-pre-emptive/pre-emptive CPU scheduling algorithms to find turnaround time and waiting time (minimum 2 from all process scheduling algorithms)
9. Implement process synchronization using Semaphores (use any one real time example application)
10. Implement Banker's algorithm for the purpose of Deadlock avoidance
11. Implement the MVT and MFT Memory Management techniques
12. Implement the following Contiguous Memory Allocation techniques
    a) Worst-fit b) Best-fit c) First fit
13. Implement Page Replacement algorithms. (minimum two)
14. Implement File Allocation strategies. (minimum two)
15. Implement File Organization techniques. (minimum two)
16. Implement Disk Scheduling algorithms. (minimum two)

Note: From program 8 to 16 it is the faculty choice to decide which algorithms to be implemented

Case Study: Build process management, memory management on any shell interpreter

Reference link for case study:

http://www.cs.ecu.edu/sartipi/courses/OS/f12/3.LabProjects/2.Stallings-TextBook-Projects/html/

REFERENCES:

1. William Stallings, *Operating Systems – Internal and Design Principles*, 9th Edition, Pearson education/PHI,2018.
2. D.M. Dhamdhere, *Operating systems - A Concept based Approach*, 3rd Edition, TMH,2017.
3. Charles Crowley, *Operating Systems - A Design Approach*, 1st Edition, TMH,2017.
4. Andrew S Tanenbaum, *Modern Operating Systems*, 3rd Edition, Pearson/PHI,2014

WEB REFERENCES:

1. https://nptel.ac.in/courses/106/105/106105214/
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-828-operating-system-engineering-fall-2012/lecture-notes-and-readings/