

**OBJECTIVES:**

The student should be made to:

- Learn shell programming and the use of filters in the UNIX environment.
- Be exposed to programming in C using system calls.
- Learn to use the file system related system calls.
- Be exposed to process creation and inter process communication.
- Be familiar with implementation of CPU Scheduling Algorithms, page replacement algorithms and Deadlock avoidance

**LIST OF EXPERIMENTS:**

1. Basics of UNIX commands.
2. Shell Programming.
3. Implement the following CPU scheduling algorithms
  - a) Round Robin b) SJF c) FCFS d) Priority
4. Implement all file allocation strategies
  - a) Sequential b) Indexed c) Linked
5. Implement Semaphores
6. Implement all File Organization Techniques
  - a) Single level directory b) Two level c) Hierarchical d) DAG
7. Implement Bankers Algorithm for Dead Lock Avoidance
8. Implement an Algorithm for Dead Lock Detection
9. Implement e all page replacement algorithms
  - a) FIFO b) LRU c) LFU
10. Implement Shared memory and IPC
11. Implement Paging Technique of memory management.
12. Implement Threading & Synchronization Applications

**TOTAL: 45 PERIODS****OUTCOMES:**

At the end of the course, the student should be able to

- Implement deadlock avoidance, and Detection Algorithms
- Compare the performance of various CPU Scheduling Algorithm
- Critically analyze the performance of the various page replacement algorithms
- Create processes and implement IPC

**REFERENCE:**

[spoken-tutorial.org](http://spoken-tutorial.org)

**LIST OF EQUIPMENT FOR A BATCH OF 30 STUDENTS:**

Standalone desktops with C / C++ / Java / Equivalent compiler 30 Nos.

(or)

Server with C / C++ / Java / Equivalent compiler supporting 30 terminals