

PO	1	2	3	4	5	6	7	8	9	10	11	12
Level		M	M		M							

**Syllabus for B. Tech. IV Year I semester**  
**Computer Science and Engineering**  
**LINUX PROGRAMMING**

**Code: 7F708**

**L**      **T**      **P**      **C**  
**3**      **-**      **-**      **3**

**Prerequisite : Operating Systems**

**Course Objectives:** To Induce working principles of Linux operating system, usage of File handling utilities, Security by file permissions, process utilities, Disk utilities, Networking utilities. To impart the shell responsibilities and meta-characters of it, control structures, shell interrupt processing, functions, debugging shell scripts. To impart basics of file concepts kernel support for file, file structure and low-level I/O functions, system calls (file API's). Induce knowledge regarding Directory management and its API. To demonstrate basics of process creation, execution and synchronization mechanisms. Give knowledge regarding a signal, need for having them, usage of various signals. To narrate the need for Inter Process Communication. Explore the possible mechanisms to implement System V APIs. To demonstrate the usage of Message queues. To incorporate implementation for semaphore API and shared memory API. To explain the need for using a basic Client-Server model.

**Course Outcomes:**

**At the end of this course, the student will be able to**

1. Describe the basic Linux commands
2. Write Shell Scripts
3. Enlist various System Calls in Linux
4. Classify various system calls to handle the processes and signal the process
5. Elaborate the working of IPC
6. Demonstrate the significance of Semaphores for Kernel support and simulate program using the same.

**UNIT-I :** Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities, sed – scripts, operation, addresses, commands, applications, awk – execution, fields and records, scripts, operation, patterns, using system commands in awk. (Applications: Determining what types of files are present in a system, debugging issues with file accessibility, finding a process troubling for a task and discarding from its existing, Write and extract necessary information from huge test files.)

**UNIT – II** Working with the Bourne again shell(bash): Introduction, shell responsibilities, pipes and input Redirection, output redirection, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

(Applications: Writing shell scripts for automating most of the regular jobs, taking backup on regular basis and restoring the same)

**UNIT-III:** Files: File Concept, File System Structure, I nodes, File Attributes, File types, Library functions, the standard I/O and formatted I/O in C, stream errors, kernel support for files, System calls, file descriptors, low level file access – File structure related system calls(File APIs), file and record locking, file and directory management – Directory file APIs, Symbolic links & hard links. (Applications: write some system programs to interact with file system, developing small systemsoftwares to work with files and devices, Developing program's on directory management system)

**UNIT-IV:** Process – Process concept, Kernel support for process, process attributes, process control - process creation, waiting for a process, process termination, zombie process, orphan process, Process APIs. Signals– Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.

(Applications: Applications to find number of typical processes are under different context and controlling them in synchronous manner. Develop user defined modules for handling a signal and controlling several issues with signals.)

**UNIT-V:** Interprocess Communication: Introduction to IPC, Pipes, FIFOs, Introduction to three types of IPC-message queues, semaphores and shared memory. Message Queues Kernel support for messages, Unix system V APIs for messages, client/server example.

(Applications: Developing applications complying with IPC mechanisms, Developing an application that exchanges a set of messages among different processes. Write a client server application to go with any concurrent approach)

**UNIT- VI :** Semaphores-Kernel support for semaphores, Unix system V APIs for semaphores. Shared Memory- Kernel support for shared memory, Unix system V APIs for shared memory, semaphore and shared memory example.

(Applications: Develop critical section handling mechanisms to deal with any real problems. Building applications to share a piece of memory resource among processes concurrently)

### **TEXT BOOKS:**

1. Unix System Programming using C++, T.Chan, PHI.
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH,2006.
3. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones,Wrox, Wiley India Edition,rp-2008

### **REFERENCES:**

- 1.Linux System Programming, Robert Love, O'Reilly, SPD.
- 2.Advanced Programming in the Unix environment, 2nd Edition, W.R.Stevens, Pearson Education.
- 3.Unix Network Programming ,W.R.Stevens,PHI.
- 4.Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education