



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

COURSE CONTENT

ADVANCED UNIX PROGRAMMING								
I Semester: CSE								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P		C	CIA	SEE
2515808	Foundation	3	0	0	3	40	60	100
		Contact Classes: 45			Tutorial Classes: Nil		Practical Classes: Nil	
		Total Classes: 45						
Prerequisites: A course on “Advanced UNIX Programming”								

Course Overview:

Advanced UNIX Programming focuses on system-level programming concepts in UNIX and Linux environments.

The course covers process management, inter-process communication (IPC), signals, threads, and synchronization mechanisms.

It emphasizes system calls, file systems, memory management, and socket programming for network applications.

Course Objectives:

1. To provide a strong foundation in UNIX/Linux operating system concepts, shell scripting, and command-line utilities for efficient system-level programming and automation.
2. To develop programming skills in C for implementing file handling, process management, and directory operations using UNIX system calls and APIs.
3. To enable students to understand and apply process creation, signal handling, multithreading.
4. To design and implement Inter Process Communication (IPC) techniques such as pipes, FIFOs, message queues, semaphores, and shared memory for client-server and parallel applications.
5. To develop network programming skills using Berkeley sockets for creating scalable client-server applications

Course Outcomes: After Completion of the Course, Students should be able to

1. Apply Linux file handling, process, networking, and text utilities to automate system administration tasks and backup management.
2. Implement shell scripts using control structures, variables, functions, and I/O redirection to develop automated system monitoring tools.
3. Analyze file system structures, file metadata, and directory operations to optimize storage management
4. Design interprocess communication mechanisms, including pipes, FIFOs, message queues, and semaphores, for concurrent and client-server applications.
5. Evaluate shared memory and socket-based IPC solutions for building scalable networked applications and multi-client server systems.

UNIT - I:

Linux Utilities - File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

Shell programming with Bourne again shell (bash) - Introduction, shell responsibilities, pipes and Redirection, here documents, 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

UNIT - II:

Files and Directories - File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, creat, read, write, close, lseek, dup2, file status information-stat family, file and record locking- fcntl function, file permissions - chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links – symlink, link, unlink. **Directories**

- Creating, removing and changing Directories- mkdir, rmdir, chdir, obtaining current working directory- getcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir functions

UNIT - III:

Process – Process concept, Layout of a C program image in main memory, Process environment- environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process control - process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process, system call interface for process management- fork, vfork, exit, wait, waitpid, exec family, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes.

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

UNIT - IV:

Interprocess Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions. **Message Queues** - Kernel support for messages, APIs for message queues, client/server example. **Semaphores** - Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

UNIT - V:

Shared Memory - Kernel support for shared memory, APIs for shared memory, shared memory example. **Sockets** - Introduction to Berkeley Sockets, IPC over a network, Client-Server model, Socket address structures (Unix domain and Internet domain), Socket

system calls for connection-oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection, Multiple simultaneous clients, Socket options- setsockopt and fcntl system calls, Comparison of IPC mechanisms

TEXT BOOKS:

1. Unix System Programming using C++, T. Chan, PHI.
2. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S. A. Rago, Pearson Education.
3. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
4. Unix Network Programming, W. R. Stevens, PHI.

REFERENCE BOOKS:

- 1 C Programming Language, Kernighan and Ritchie, PHI.
- 2 Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
- 3 Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.
- 4 System Programming with C and Unix, A. Hoover, Pearson.
- 5 Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.
- 6 Unix shell Programming, S. G. Kochan and P. Wood, 3rd edition, Pearson Education.
- 7 Shell Scripting, S. Parker, Wiley India Pvt. Ltd.
- 8 Unix and Shell programming, B. A. Forouzan and R. F. Gilberg, Cengage Learning.
- 9 Linux System Programming, Robert Love, O'Reilly, SPD.

ELECTRONIC RESOURCES:

1. <https://www.coursera.org/specializations/advanced-unix-system-programming-performance>
2. <https://www.coursera.org/learn/advanced-unix-concepts-and-ipc>.
3. <https://getvm.io/tutorials/advanced-programming-in-the-unix-environment>
4. <https://getvm.io/tutorials/unix-application-and-system-programming-lecture-notes>

MATERIALS ONLINE:

1. Course template
2. Tutorial question bank
3. Tech talk and Concept Video topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. E-Learning Readiness Videos (ELRV)

