LINUX PROGRAMMING

VI Semester: CSE / IT

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
AITB12	**Core**	2	1	0	3	30	70	100

Contact Classes: 45	**Tutorial Classes: Nil**	**Practical Classes: Nil**	**Total Classes: 60**

OBJECTIVES:

Students will try to learn:

I   The features and architecture of Linux operating system along with basic utilities.

II   The concepts of Low-Level system call for file handling and process management.

III   Mechanisms for establishing communication in client- server applications.

COURSE OUTCOMES:

At the end of the course the students should be able to:

CO 1   Explain the features and architecture of open-source LINUX operating system for overcoming the problems of operating systems like Windows, Mac etc.

CO 2   Make use of general purpose, text processing and network utilities for managing file system.

CO 3   Recall shell responsibilities for solving the problems using Linux utilities

CO 4   Make use of shell scripting for system administration tasks, such as performing disk backups and evaluating system logs.

CO 5   Demonstrate file and directory system calls in C programming for accessing operating system services.

CO 6   Recall file and record locking mechanisms for synchronizing data.

CO 7   Interpret the concepts of process for process creation, scheduling, controlling and termination.

CO 8   Make use of signal handler for handling software interrupts such as abort () pause () exit()

CO 9   Demonstrate concepts of dynamic memory management functions such as malloc, free, realloc and calloc for managing memory.

CO 10   Illustrate IPC mechanisms (such as pipes, named pipes, shared memory, message queues, semaphores, and sockets) for inter process communication.

CO 11   Choose the appropriate protocol such as TCP or UDP for effective communication in client-server applications.

MODULE-I	**INTRODUCTION AND LINUX UTILITIES**

Introduction to Linux operating system: History of Linux, features of Linux, architecture of Unix/Linux, Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities;Applications: 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.

MODULE-II	**FILES AND DIRECTORIES SYSTEM CALLS**

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, create, read, write, close, lseek,dup2, file status informationstat family, file and record locking- fcntl function, permission- chmod, fchmod, file

ownership- chown, lchown, links- soft links & 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.

MODULE-III	PROCESS AND SIGNALS

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 & controlling terminal, differences between threads & 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.

MODULE-IV	INTERPROCESS COMMUNICATION

Inter process 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 123 | P a g e and named pipes. 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.

MODULE-V	SHARED MEMORY AND SOCKETS

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 & internet domain), Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs- single client/server connection, Multiple simultaneous clients.

Text Books:

1. Sumitabha Das, ―Your Unix The Ultimate Guide‖, Tata McGraw-Hill, New Delhi, India, 2012.
2. W. Richard. Stevens, ―Advanced Programming in the UNIX Environment‖, Pearson Education, New Delhi, India, 2013.

Reference Books:

1. T. Chan, ―Unix System Programming using C++‖, PHI. 4 th Edition, 2007.
2. N. Mathew, R. Stones, Wrox, ―Beginning Linux Programming‖, Wiley India Edition, 4 thEdition, 2014.
3. Graham Glass, King Ables, ―Unix for Programmers and Users‖, Pearson Education, 3 rd Edition, 2008.
4. A. Hoover, ―System Programming with C and Unix‖, Pearson Education, 3 rd Edition, 2008.
5. K. A. Robbins, ―Unix System Programming, Communication, Concurrency and Threads‖, Pearson Education, , 4 thEdition, 2014..

Web References:

1. https://www.edx.org/course/introduction-linux-linuxfoundationx-lfs101x-0 2.
2. http://www.tutorialspoint.com/listtutorials/linux/1
3. http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/unix_lecture_notes.php