

LINUX INTERNALS

VI Semester: IT								
Course Code	Category	Hours / Week			Credits	Maximum Marks		
AIT005	Core	L	T	P	C	CIA	SEE	Total
		3	1	-	4	30	70	100
Contact Classes: 45		Tutorial Classes: 15		Practical Classes: Nil		Total Classes: 60		
<p>OBJECTIVES:</p> <p>The course should enable the students to:</p> <ol style="list-style-type: none"> I. Familiarize students with the Linux environment, and able to run commands on a standard Linux operating system. II. Provide the skills needed to develop and customize Linux shell programs and to make effective use of a wide range of standard Linux programming and development tools. III. Design Able to write moderate C programs utilizing common system calls. IV. Develop the skills necessary for system programming and inter and intra process communication programming. 								
<p>COURSE OUTCOMES (COs):</p> <p>CO1: Understand the basic commands of linux operating system and can write shell scripts.</p> <p>CO2: Create file systems and directories and operate those using programs.</p> <p>CO3: Understand the processes background and fore ground by process and signals system calls.</p> <p>CO4: Create shared memory segments, pipes, message queues and can exercise inter process communication.</p> <p>CO5: Create sockets and semaphores to interact between process of different system.</p>								
<p>COURSE LEARNING OUTCOMES:</p> <p>Students, who complete the course, will have demonstrated the ability to do the following:</p> <ol style="list-style-type: none"> 1. Learn the importance of Linux architecture along with features. 2. Identify and use Linux utilities to create and manage simple file processing operations. 3. Develop shell scripts to perform more complex tasks in shell programming environment. 4. Illustrate file processing operations such as standard I/O and formatted I/O. 5. Illustrate memory management of file handling through file/region lock. 6. Design and Implement in C some standard linux utilities. 7. Understand process structure, scheduling and management through system calls. 8. Implement C programs to control process using system calls and indentify difference between process and threads. 9. Generalize signal functions to handle interrupts by using system calls. 10. Design and implement inter process communication (IPC) in client server environment by using pipe and named pipes system calls. 11. Design and implement inter process communication (IPC) in client server environment by using message queues system calls. 12. Illustrate client server authenticated communication in IPC through shared memory. 13. Familiarity with Inter Process Communication using Semaphores. 14. Demonstrate various client server applications on network using TCP or UDP protocols. 15. Design custom based network applications using the Sockets Interface in heterogeneous platforms. 								

UNIT-I	LINUX UTILITIES AND BOURNCE AGAIN SHELL (bash)	Classes: 10
<p>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; Sed: Scripts, operation, addresses, commands; awk: Execution, fields and records scripts, operation, patterns, actions, applications;</p> <p>Working with the Bourne again shell(bash): Introduction, shell responsibilities, pipes and input Redirection, output 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.</p>		
UNIT-II	FILES AND DIRECTORIES SYSTEM CALLS	Classes: 08
<p>Files and Directories: File Concept, File types, File System Structure, File metadata- Inodes, kernel support for files, file System calls for file I/O operations- open, create, read, write, close, lseek,dup2, file status information- stat family, file and record locking- fcntl function, permission- chmod, fchmod, file ownership- chown, lchown, fchown, links- soft links & hard links- symlink, link, ulink.</p> <p>Directories: creating, removing and changing directories- mkdir, rmdir, chdir, obtaining current working directory- getcwd, directory contents, scanning directories- opendir, readdir, closedir, rewind dir functions.</p>		
UNIT-III	PROCESS AND SIGNALS	Classes:10
<p>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.</p> <p>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.</p>		
UNIT-IV	INTERPROCESS COMMUNICATION	Classes: 09
<p>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 and named pipes, popen & 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.</p>		
UNIT-V	SHARED MEMORY AND SOCKETS	Classes: 08
<p>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, Socket options - setsockopt and fcntl system calls, Comparison of IPC Mechanisms.</p>		

Text Books:

1. Sumitabha Das, “Your Unix The Ultimate Guide”, Tata McGraw-Hill, New Delhi, India, 2007.
2. W. Richard. Stevens, “Advanced Programming in the UNIX Environment”, 1st Edition, Pearson Education, New Delhi, India, 2005.

Reference Books:

1. T. Chan, “Unix System Programming using C++”, PHI.
2. N. Mathew, R. Stones, Wrox, “Beginning Linux Programming”, 4th Edition, Wiley India Edition.
3. Graham Glass, King Aables, “Unix for Programmers and Users”, 3rd Edition, Pearson Education.
4. A. Hoover, “System Programming with C and Unix”, Pearson Education.
5. K. A. Robbins, “Unix System Programming, Communication, Concurrency and Threads”, Pearson Education.
6. S. G. Kochan and P. Wood, “Unix Shell Programming”, 3rd Edition, Pearson Education.
7. B. A. Forouzan and R. F. Gilberg, “Unix and Shell Programming”, Cengage Learning.
8. Robert Love, “Linux System Programming”, O'Reilly, SPD.