

LINUX PROGRAMMING

VI Semester: CSE / IT																																						
Course Code	Category	Hours / Week			Credits	Maximum Marks																																
AITB12	Core	L	T	P	C	CIA	SEE	Total																														
		2	1	-	3	30	70	100																														
Contact Classes: 30		Tutorial Classes: 15		Practical Classes: NIL			Total Classes: 45																															
<p>I. COURSE OVERVIEW: This course provides a deep understanding of the operating system architecture and low-level interfaces (principally, system calls and library functions) that are required to build system-level, multithreaded, and network applications on Linux and UNIX systems. The course consists of a mixture of detailed presentations coupled with a large number of carefully designed practical exercises that allow participants to apply the knowledge learned in the presentations. By the completion of the course, participants will have the mastery needed to write complex system, network, and multithreaded applications on a Linux or UNIX system.</p> <p>II. OBJECTIVES: The course should enable the students to:</p> <ul style="list-style-type: none"> I The features and architecture of Linux operating system along with linux utilities and shell scripting language to solve problems. II The concepts of Low-Level system call for file handling and process management. III To develop skills the necessary for systems programming including file system programming, process and signal management. IV Mechanisms for establishing communication in client- server applications. <p>III. COURSE OUTCOMES: After successful completion of the course, students should be able to:</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 15%;">CO 1</td> <td style="width: 65%;">Demonstrate operations using file handling, text processing and Linux utilities.</td> <td style="width: 20%;">Understand</td> </tr> <tr> <td>CO 2</td> <td>Outline the different shell scripts to execute systems programs and application programs.</td> <td>Remember</td> </tr> <tr> <td>CO 3</td> <td>Make use of different system calls for file I/O operations and managing the file systems.</td> <td>Apply</td> </tr> <tr> <td>CO 4</td> <td>Demonstrate the concepts of process and signal system calls for process creation, scheduling, controlling and termination.</td> <td>Understand</td> </tr> <tr> <td>CO 5</td> <td>Outline IPC mechanisms such as pipes, shared memory, message queues, semaphores for performing inter process communication</td> <td>Remember</td> </tr> <tr> <td>CO 6</td> <td>Utilize socket concepts for connection-oriented and connectionless communication between client and server systems.</td> <td>Apply</td> </tr> </table> <p>IV. SYLLABUS:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">MODULE-I</td> <td style="width: 65%;">INTRODUCTION AND LINUX UTILITIES</td> <td style="width: 20%;">Classes: 10</td> </tr> <tr> <td colspan="3"> 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. </td> </tr> <tr> <td>MODULE-II</td> <td>FILES AND DIRECTORIES SYSTEM CALLS</td> <td>Classes: 08</td> </tr> <tr> <td colspan="3"> 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 information- stat family, file and record locking- fcntl function, permission- chmod, fchmod, file ownership- chown, lchown, links- soft links </td> </tr> </table>									CO 1	Demonstrate operations using file handling, text processing and Linux utilities.	Understand	CO 2	Outline the different shell scripts to execute systems programs and application programs.	Remember	CO 3	Make use of different system calls for file I/O operations and managing the file systems.	Apply	CO 4	Demonstrate the concepts of process and signal system calls for process creation, scheduling, controlling and termination.	Understand	CO 5	Outline IPC mechanisms such as pipes, shared memory, message queues, semaphores for performing inter process communication	Remember	CO 6	Utilize socket concepts for connection-oriented and connectionless communication between client and server systems.	Apply	MODULE-I	INTRODUCTION AND LINUX UTILITIES	Classes: 10	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	Classes: 08	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 information- stat family, file and record locking- fcntl function, permission- chmod, fchmod, file ownership- chown, lchown, links- soft links		
CO 1	Demonstrate operations using file handling, text processing and Linux utilities.	Understand																																				
CO 2	Outline the different shell scripts to execute systems programs and application programs.	Remember																																				
CO 3	Make use of different system calls for file I/O operations and managing the file systems.	Apply																																				
CO 4	Demonstrate the concepts of process and signal system calls for process creation, scheduling, controlling and termination.	Understand																																				
CO 5	Outline IPC mechanisms such as pipes, shared memory, message queues, semaphores for performing inter process communication	Remember																																				
CO 6	Utilize socket concepts for connection-oriented and connectionless communication between client and server systems.	Apply																																				
MODULE-I	INTRODUCTION AND LINUX UTILITIES	Classes: 10																																				
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	Classes: 08																																				
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 information- stat family, file and record locking- fcntl function, permission- chmod, fchmod, file ownership- chown, lchown, links- soft links																																						

<p>& 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.</p>		
MODULE-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>		
MODULE-IV	INTERPROCESS COMMUNICATION	Classes: 9
<p>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. 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>		
MODULE-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.</p>		
Text Books:		
<ol style="list-style-type: none"> 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”, Pearson Education, New Delhi, India, 1st Edition, 2005. 		
Reference Books:		
<ol style="list-style-type: none"> 1. T. Chan, “Unix System Programming using C++”, PHI. 4th Edition, 2007. 2. N. Mathew, R. Stones, Wrox, “Beginning Linux Programming”, Wiley India Edition, 4th Edition, 2014. 3. Graham Glass, King Ables, “Unix for Programmers and Users”, Pearson Education, 3rd Edition, 2008. 4. A. Hoover, “System Programming with C and Unix”, Pearson Education, 3rd Edition, 2008. 5. K. A. Robbins, “Unix System Programming, Communication, Concurrency and Threads”, Pearson Education, , 4th Edition, 2014. 		
Web References:		
<ol style="list-style-type: none"> 1. https://www.edx.org/course/introduction-linux-linuxfoundationx-lfs101x-0 2. http://www.tutorialspoint.com/listtutorials/linux/1 3. http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/unix_lecture_notes.php 		
E-Text Books:		
<ol style="list-style-type: none"> 1. http://www.freebookcentre.net/UnixCategory/Free-Linux-Programming-Books-Download.html 2. http://www.fuky.org/abicko/beginning-linux-programming.pdf 3. http://www.penguintutor.com/linux/introduction-creating-website 		
MOOC Course		
<ol style="list-style-type: none"> 1. https://training.linuxfoundation.org/free-linux-training 2. http://cloud62.wixsite.com/v-mooc/linux-programming 		