



Course Title:UNIX OPERATING SYSTEM AND SHELL PROGRAMMING

Course Level: UG

Course Code :CSIT306

Credit Units:03

L	T	P/ S	SW/F W	TOTAL CREDIT UNITS
2	-	2	-	3

Course Objectives :

- The Objective of this course is to expose the students to the fundamentals and the concepts of Unix Operating System.
- This course will prepare the students to work on UNIX ENVIRONMENT as a technical user or system administrator of a powerful, fast growing, multitasking, open operating system which is currently used on all types of computers from micros to mainframes.
- This course introduces students to the fundamentals of the UNIX/Linux operating system and shell programming. It provides an overview of the history of UNIX/Linux and an explanation of operating systems. The course covers in detail basic commands, the vi editor ,the file structure, the shell environment and shell scripts.

Pre-requisites:

Concepts of Operating System

Course Contents/Syllabus:

	Weightage (%)
Module I : INTRODUCTION	15%
Introduction to Operating System , History of Unix ,UNIX Family, Unix System Layered and Detailed Architecture: Concept of Files in UNIX, Absolute Path and Relative Path , UNIX file system structure, Types of shells (Bourne, BASH, KORN,C), Process and Process States , Inode, Introduction of basic system calls	
Module II : UNIX COMMANDS	30%
Telnet connect: Login, password, shell and commands, logout, current working directory, referring to home directories, Commands to move around by path concept, creating new directories, creating files –touch , cat ; copying files; moving files, Deleting files and directories; looking at files: cat, more, pg, less , head , tail; Cal, banner, file, wc, sort, cut, grep ,cmp, comm., diff ;Calculator: expr , bc ;Getting online help; manual pages ; listing commands , meta characters ,Wildcards; hidden files; Standard input and output; redirecting input and output; filter; pipes; file permissions; user and group; Interpreting file permissions; Permission Dependencies; Changing permissions, Setting Permissions. Managing file links; hard links; symbolic links; jobs and process: process ID; foreground	

and background jobs; suspend and interrupt a process; killing jobs; changing password, exit.	
Module III : VI EDITOR	15%
Command mode, insert mode and last line mode; command to delete character, insert line; deleting text, command for moving the cursor; including other files; running shell commands; getting vi help; search and replace commands; changing and deleting text, Change word, Change line, Delete current line, Delete n lines, Delete remainder of Lines; copying and moving; Saving and Exiting.	
Module IV : SHELL PROGRAMMING	30%
Shell as an interpreter; pattern matching; ; redirection; pipes; command substitution; shell variables, environment variables , Keywords, Assignment Statements, read , echo ,Shell scripts and execution methods, Setting positional parameters (set command), Shift , metacharacters , arithmetic operators, logical and relational operators, Test Command: Numerical Test, File Test and String Test ; Control Flow through if, case ; Loops ;while, until , for	
Module V : SYSTEM ADMINISTRATION	10%
Adding and Removing Users, Starting up and Shutting down the System, Disk Management, File System Mounting and Unmounting, Monitoring System Usage, Ensuring System Security	

Student Learning Outcomes:

Upon completion of this course, the students should be able to do the following:

- Login to a Unix / Linux system using secure shell.
- Define and understand absolute versus relative path names.
- Run basic commands from the shell prompt.
- Get help for known commands.
- Define the purpose of a the shell
- Create, change to and remove directories.
- Use file name substitution at the command line.
- Define and use in a command line redirection and piping
- Use standard in, standard out and standard error on a command line
- Use the vi editor
- Create regular expressions and know when and how to use them.
- Change owner, group and permissions at the command line

- Create a shell script and execute it.
- Create system variables and know how and when to use them in a shell script.
- The use of command arguments in a shell script.
- Use looping and decision statements in a shell script
- Allow a shell script to prompt the user and get information from the command line.
- To customize a UNIX login account using environment variables, configuration files and startup scripts.
- To maintain UNIX directories and files, manage UNIX jobs and processes, use of UNIX pipes and file redirection, manipulate data with proper use of Unix filters, role of an operating system and UNIX philosophy.

Pedagogy for Course Delivery:

The course joins together lectures, practical labs, case studies, work assignments. The readings for the course are attracted from text books, journals and commands , scripts published by experts using UNIX/LINUX in real life environment..

List of Experiments:

1. Write a Shell Script that takes a search string and filename from the terminal & displays the results.
2. Write a Shell Script that takes pattern and filename as command line arguments and displays the results appropriately i.e. pattern found/pattern not found.
3. Write a Shell Script that accepts only three arguments from the command line. The first argument is the pattern string, the second argument is the filename in which the pattern is to be searched and the third argument is the filename in which the result is to be stored.
4. Write a Shell Script that accepts a filename as a command line argument and finds out if its a regular file or a directory. If its a regular file, then performs various tests to see if it is readable, writeable, executable etc.
5. Write a Shell Script which creates the following menu and prompts for choice from user and runs the chosen command.
 - Today's date
 - Process of user
 - List of files
 - Quit to UNIX
6. Write a Shell Script that computes the factorial of a given number.
7. Write a Shell Script that works like a calendar reminding the user of certain things depending on the day of the week.
8. Write a Shell Script that changes the extension of a group of files from txt to doc
9. Write a Shell Script that accepts both filename and a set of patterns as positional parameters to a script.
10. Write a Shell Script which will redirect the output of the date command without the time into a file.
11. Write a Shell Script (using while loop) to execute endlessly (until terminated by user) a loop which displays contents of current directory, disk space status, sleep for 30 seconds and display the users currently logged in on the screen.
12. Write a Shell Script that receives two filenames as arguments. It should check whether content of the two files is same or not. If they are same, second file should be deleted.
13. If a number is input through the keyboard, WASS to calculate sum of its digits.
14. Write a Shell Script that performs a count-down either from 10 (default) or from the value that is entered by the user.

15. Write a Shell Script which takes a command line argument of Kms and by default converts that number into meters. Also provide options to convert km to dm and km to cm.
16. Write a Shell Script using for loop, which displays the message "Welcome to the UNIX System"
17. Write a Shell Script to change the filename of all files in a directory from lower-case to upper-case.
18. Write a Shell Script that examines each file in the current directory. Files whose names end in **old** are moved to a directory named **old files** and files whose names end in **.c** are moved to directory named **programs**.
19. Write a shell script to give the result of the student. Take marks of the five subjects , student name, roll no , percentage and show a message whether a student gets division as per the following rules:
 70% and above ---- distinction
 60%-70% -----first division
 40%-59% -----second division
 Less than 40% ----- fail
20. Write a shell script which reports names and sizes of all files in a directory (directory would be supplied as an argument to the shell script) whose size is exceeding 1000 bytes. The filenames should be printed in descending order of their sizes. The total number of such files should also be reported.
21. WASS for renaming each file in the directory such that it will have the current shell PID as an extension. The shell script should ensure that the directories do not get renamed.
22. WAP to calculate and print the first *m* Fibonacci numbers.
23. WASS that will receive any number of filenames as arguments. The shell script should check whether such files already exist. If they do, then it should be reported. The files that do not exist should be created in a sub-directory called **mydir**. The shell script should first check whether the sub-directory **mydir** exists in the current directory. If it doesn't exist, then it should be created. If **mydir** already exists, then it should be reported along with the number of files that are currently present in **mydir**.
24. A shell script receives even number of filenames. Suppose four filenames are supplied, then the first file should get copied into second file, the third file should get copied into fourth and so on. If odd number of filenames is supplied then no copying should take place and an error message should be displayed.
25. WASS to identify all zero-byte files in the current directory and delete them. Before proceeding with deletion, the shell script should get a confirmation from the user.
26. WASS to compute the **GCD** and **LCM** of two numbers.
27. Two numbers are entered through the keyboard. WAP to find the value of one number raised to the power of another.
28. WASS that prompts the user for the password. The user has maximum of 3 attempts. If the user enters the correct password, the message "Correct Password" is displayed else the message "Wrong Password".
29. WASS that repeatedly asks the user repeatedly for the "Name of the Institution" until the user gives the correct answer.
30. WAP to generate all combinations of 1, 2 and 3 using **for loop**.

Assessment/ Examination Scheme:

Theory L/T (%)	Lab/Practical/Studio (%)	Total
66.6	33.4	100

Theory Assessment (L&T):

Continuous Assessment/Internal Assessment						End Examination	Term
Components down)	(Drop	Mid-Term Exam	Home Assignment	Class Performance/Response	Attendance		
Weightage (%)		10	5	10	5	70	

Lab/ Practical/ Studio Assessment:

		Continuous Assessment/Internal Assessment				End Examination	Term
Components down	(Drop	Lab file	Lab Performance	Viva	Attd	PR	viva
Weightage (%)		10	20	5	5	40	20

Text Books :

1. UNIX AND SHELL PROGRAMMING , Yashwant P.Kanetkar , BPB Publication , 2002

Reference Books :

2. “Unix: Concepts and Application”, Sumitabha Das, TMH, Second Edition, 1998

3.“Linux Programming by Examples: The Fundamentals”, Arnold Robbins, Pearson Education, First Edition, 2004

4.“Design of the Unix operating System” , Maurice J. Bach ,PHI, First Edition, 1986

5.Unix Shell Programming, by Stephen G. Kochan and Patrick Wood , Pearson Education ,3rd edition , 2007

6. Introduction to UNIX , David I. Schwartz , Pearson Education , Second Edition , 2009

7. UNIX SHELLS by Example, Ellie Quigley , Prentice Hall, Fourth Edition, 2008