Programme: M.C.A

Course Code: CSC-106 **Title of the Course:** LINUX Lab

Number of Credits: 4 (2L-0T-2P) **Contact hours:** 72 hours (24L-0T-48P)

Effective from AY: 2021-22

Prerequisites for the course:	Program Prerequisites	
Objectives:	The objective is to introduce students to the Linux operating system environment and provide a knowledge of basic Linux commands and shell scripting and system call API.	
Content:	**LINUX Environment** Linux Installation and disk partitioning. Shell, Linux commands, Internal and External Commands, using the documentation/manual, users in Linux: user id, effective user id, use of commands su, sudo, id Basic commands: echo, who, whoami, date, cal, ls, passwd, history, shutdown. Input and output redirection operators (<,<<, >, >>)	3L + 3P
	The Linux File System, File and Directory management Structure of LINUX file system. Parent-child relationship. Concept of Home directory, current working directory and referring to home directory. Special Files: . and .. Absolute and relative pathnames. Use of PATH variable, Use of command: mkdir, rmdir, pwd, ls and cd. Use of file management commands: nano, touch, cat, cp, mv and rm. FIND command: Searching for a file using find, Finding List of files and directories. Concept of hard disk partitions, file system, Superblock and Inodes, General structure of Linux inode. use of stat command. Analysing the output of ls -l command. File type and permission. Use of chmod command. File ownership: Changing ownership using chown and chgrp commands. Modification and access times. Default file and directory permissions. Use of umask command. Concept of symbolic links. Hard and soft links. Use of ln command to create hard and soft links. Use of commands du, df, tar, zip, gzip, type, which	4L + 8P

	Filters: File commands- sort, wc, uniq, comm, cmp, diff, pg, tail, head, less, and more , Cut and Paste command Shells' sequence of interpretation of a command; Connecting commands with pipes Regular expressions: grep & sed command AWK script: Selection criteria and action- The BEGIN and END sections, Splitting a line into fields and using printf. Getline function and reading input from files. Writing output to file and pipes. Awk system variables. Using regular expressions. Relational and Boolean operations. Command line parameters and environment variables. Programming constructs: if, for, while.	6L + 8P
	Process Management Concept of UNIX process. Role of init in process creation. Process ID and exit status of a process. Displaying process attributes using ps command, Killing processes, foreground and background processes. Use of commands job, fg, bg	1L + 2P
	Package management: Installing & removing packages	1L+1P
	Shell Script Shell scripts and execution methods. The dot command, Interactive and Non Interactive execution. Use of export command, Aliases and command history. Shell variables, Special variables, Built-in shell parameters. Command line arguments. Escaping and quoting. Difference between single and double quotes. Command substitution, brace and tilde expansion, I/O using read and echo. Escape sequences, 'test' command, arithmetic expressions, operators, Control flow: For, If, While, Case. Shell functions, error handling, debugging.	4L + 8P
	System programming Introduction to system programming, System calls and library functions. **Files and Directory system calls** List of sample programs *1.* Write a program to implement the functionality of Linux command *touch* *2.* Write a program to implement the functionality of Linux command *cat*	5L + 18P

	3. Write a program to implement the functionality of Linux command *ls* 4. Write a program to redirect the output of all the printf statements to a user file using dup system call. 5. Write a program to read the standard input from a user file using dup system call. 6. Write a program to implement the functionality of Linux command *chmod* 7. Write two programs : one called parent.c, the other called child.c. The parent program reads two integers from the keyboard and arithmetic operator (+, -, *, /). The read information is transmitted to a child process. After the child process finishes the operation, it transmits the result to the parent process. The parent process prints the result on the screen. 8. Write a c program namely "parent.c", which reads the processes along with their burst time (bt) and saves it in a file. Using fork, create a child process namely fcfs.c, which takes the filename containing process information as a parameter from the parent. The child process task is to calculate the average waiting time using the FCFS scheduling algorithm.	
Pedagogy:	Practical/ tutorials/assignments/self-study	
References/ Readings	1. Unix Concepts and Applications – Sumitaba Das, Tata MacGraw Hill. 2. Unix and Shell Programming – Graham Glass and King Ables Pearson Education 3. C and Unix Programming – Kerningham and Pike, Prentice Hall 4. UNIX man pages	
Learning Outcomes	Upon completion of this course, the student will be able to: 1. Run various LINUX commands 2. Write shell script on LINUX OS. 3. Use various advanced LINUX tools such as grep, SED and AWK	