

COURSE TITLE	LINUX AND SCRIPTING LANGUAGES
COURSE CODE	01EC0310
COURSE CREDITS	1

Objective:

- 1 This course aims to provide students with a comprehensive understanding of Linux operating systems and scripting languages, specifically Python and TCL, in the context of Electronic Design Automation (EDA). Students will learn how to leverage the power of Linux, automate design processes using Python and TCL scripts, and gain hands-on experience with the integration of these scripting languages in Electronic Design Automation (EDA) tools.

Course Outcomes: After completion of this course, student will be able to:

- 1 Understand and apply Linux command-line operations for efficient use of EDA tools and design environments.
- 2 Develop Python scripts for automation, data analysis, and interaction with EDA tools to improve design productivity.
- 3 Create and utilize TCL/TK scripts, including GUI development, for task automation and extending EDA tool functionality.
- 4 Integrate Python and TCL scripting techniques to streamline EDA design flows, enhance collaboration, and solve real-world challenges.

Pre-requisite of course: Basic knowledge of computers and fundamentals of operating systems, Fundamental understanding of files and file handling.

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
0	0	2	0	0	0	50	50
Contents : Unit	Topics						Contact Hours
Total Hours							

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	Experiment No. 1 Explore OS Components: Identify kernel, shell, and file system using system commands and visualize process hierarchy.	2
2	Experiment No. 2 Process & Memory Observation: Monitor CPU scheduling and memory usage using system tools and interpret results.	2

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
3	Experiment No. 3 File System Navigation: Practice directory navigation and file operations using basic Linux commands.	2
4	Experiment No. 4 File Management Task: Create, copy, move, and delete files/directories to simulate EDA project organization.	2
5	Experiment No. 5 Process Management: Execute and manage processes using commands like ps, top, and kill.	2
6	Experiment No. 6 User Permissions: Assign and modify file permissions and user access controls.	2
7	Experiment No. 7 Text Editing Practice: Create and edit script files using vim, nano, or gedit.	2
8	Experiment No. 8 Environment Variables Setup: Configure PATH and environment variables for a mock EDA tool.	2
9	Experiment No. 9 Basic Shell Scripting: Write a shell script to automate file organization and execution tasks.	2
10	Experiment No. 10 Remote Access: Connect to a remote system using SSH and execute commands.	2
11	Experiment No. 11 Batch Job Execution: Simulate job scheduling using shell scripts in a shared environment.	2
12	Experiment No. 12 Version Control (Git): Initialize a repository, commit changes, and manage versions of scripts.	2
13	Experiment No. 13 TCL Basics: Write simple TCL scripts using variables, loops, and control structures.	2
14	Experiment No. 14 Procedure Development: Create reusable TCL procedures for automation tasks.	2
15	Experiment No. 15 Regex in TCL: Implement pattern matching for parsing design or log files.	2
16	Experiment No. 16 EDA Flow Automation: Develop a TCL script to automate a mock simulation/synthesis flow.	2
17	Experiment No. 17 Advanced TCL Automation: Build a script handling multiple steps (input, execution, reporting).	2

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
18	Experiment No. 18 Python File Handling: Read and parse simulation output files and extract relevant data.	2
19	Experiment No. 19 Data Analysis & Visualization: Use NumPy and Matplotlib to analyze and plot results.	2
20	Experiment No. 20 Hybrid Automation Project: Integrate Python with TCL scripts to automate an end-to-end EDA workflow and generate reports.	2
Total Hours		40

Textbook :

- 1 The Linux Command Line, W. Shotts, No Starch Press, 2012
- 2 UNIX and Linux System Administration Handbook, E. Nemeth, G. Snyder, T. R. Hein, B. Whaley, and D. Mackin, NJ: Pearson, 2017

References:

- 1 Linux: The Complete Reference, Linux: The Complete Reference, R. Petersen, NY: McGraw-Hill, 2006
- 2 Learning the vi and Vim Editors, Learning the vi and Vim Editors, A. Robbins and E. Hannah, CA: O'Reilly Media, 2021

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
0.00	0.00	60.00	20.00	10.00	10.00

Instructional Method:

- 1 The internal evaluation will be done based on the continuous evaluation of students in the laboratory and class-room.
- 2 A practical examination will be conducted at the end of the semester for evaluation of practical performance.
- 3 Students may use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory, etc.

Instructional Method:

- 4 The course delivery method will depend upon the requirements of content and need of the students. The teacher in addition to conventional teaching methods (Chalk and Talk) may use any of the tools/techniques such as demonstration, role play, Quiz, brainstorming, Flipped class, Project based learning, Collaborative learning, MOOCs etc. for effective teaching.
- 5 It is advised to deliver the experiments based on the course content.
- 6 A project-based assessment may be considered for the evaluation of the course.

Supplementary Resources:

- 1 <https://www.kernel.org/doc/html/latest/>
- 2 <https://ubuntu.com/tutorials>
- 3 <https://training.linuxfoundation.org/>
- 4 <https://explainshell.com/>