



# INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

DUNDIGAL, HYDERABAD -500 043

## INFORMATION TECHNOLOGY

### COURSE DESCRIPTION FORM

Course Title	LINUX PROGRAMMING			
Course Code	A50517			
Regulation	R15			
Course Structure	Lectures	Tutorials	Practicals	Credits
	4	-	-	4
Course Coordinator	Mr. A Krishna Chaitanya, Associate Professor, IT			
Team of Instructors	Mr. A Krishna Chaitanya, Associate Professor, IT			

#### I. COURSE OVERVIEW:

The main objective of this course is to present the fundamental ideas behind the Linux operating system. Explores fundamental relationship between principles of Linux file system and Linux system calls, Implement extensive scripting techniques with the BASH Shell and advanced Programming Languages. Designing and developing client server applications in Linux using major methods of Inter Process (IPC) and concurrent programming, it also emphasizes files, process and signals. Provides extensive implementation skills in multithread based socket programming.

#### II. PREREQUISITES:

Level	Credits	Periods / Week	Prerequisites
UG	4	5	Computer Programming, Operating Systems

#### III. MARKS DISTRIBUTION:

Sessional Marks	University End Exam marks	Total marks
<p>There shall be 2 midterm examinations. Each midterm examination consists of subjective test. The subjective test is for 20 marks, with duration of 2 hours. Subjective test of each semester shall contain 5 one mark compulsory questions in part-A and part-B contains 5 questions, the student has to answer 3 questions, each carrying 5 marks.</p> <p>First midterm examination shall be conducted for the first two and half units of syllabus and second midterm examination shall be conducted for the remaining portion.</p> <p>Five marks are earmarked for assignments. There shall be two assignments in every theory course. Marks shall be awarded considering the average of two assignments in each course. Five marks are earmarked for assignments. There shall be two assignments</p>	75	100

#### IV. EVALUATION SCHEME:

S. No	Component	Duration	Marks
1	I Mid Examination	80 minutes	20
2	I Assignment	-	05
3	II Mid Examination	80 minutes	20
4	II Assignment	-	05
5	External Examination	3 hours	75

#### V. COURSE OBJECTIVES:

- I. **Analyze and** understand genesis and diversity of Both Linux and Unix System utilities.
- II. **Demonstrate** the shell scripts in Bourne shell Environment.
- III. **Understand** the basics of Kernel support for process and user level process.
- IV. **Demonstrate** File Systems, Signal generation and handling.
- V. **Understand** Inter Process Communication (IPC) and distinguish different concurrent mechanisms.
- VI. **Demonstrate** hands-on experience to implement multitasking with TCP/UDP socket system calls

#### VI. COURSE OUTCOMES:

1. **Identify** and use Linux utilities to create and manage simple file processing operations,
2. **Develop** shell scripts to perform more complex tasks.
3. **Illustrate** file processing operations such as standard I/O and formatted I/O.
4. **Generalize** Signal generation and handling signals.
5. **Develop** client server Inter Process Communication (IPC) Mechanisms.
6. **Illustrate** multithreading concepts to reduce the wastage of CPU time.
7. **Demonstrate** various concurrent technologies using System V Shared Memory and Semaphores
8. **Design and implement** IPCs using unnamed and named Pipes.
9. **Demonstrate** various client server applications using TCP or UDP protocols.
10. **Illustrate client** server applications with appropriate security.
11. **Design and implement** system-level applications for open-source operating systems.
12. **Design** custom based network applications using the Sockets Interface in heterogeneous platforms

## VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Level	Proficiency assessed
PO1	<b>Engineering knowledge:</b> Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	H	Assignments, Tutorials
PO2	<b>Problem analysis:</b> Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	H	Assignments
PO3	<b>Design/development of solutions:</b> Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	S	Mini Projects
PO4	<b>Conduct investigations of complex problems:</b> Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	S	Projects
PO5	<b>Modern tool usage:</b> Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	S	Mini Projects
PO6	<b>The engineer and society:</b> Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	S	--
PO7	<b>Environment and sustainability:</b> Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	N	--
PO8	<b>Ethics:</b> Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	N	--
PO9	<b>Individual and team work:</b> Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	N	--
PO10	<b>Communication:</b> Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	H	Mini Projects
PO11	<b>Project management and finance:</b> Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	S	Projects
PO12	<b>Life-long learning:</b> Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	S	Projects

N - None

S - Supportive

H - Highly Related

## VIII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Level	Proficiency assessed by
PSO1	<b>Professional Skills:</b> The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	H	Lectures, Assignments
PSO2	<b>Software Engineering Practices:</b> The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality service for business success.	H	Projects
PSO3	<b>Successful Career and Entrepreneurship:</b> The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None

S - Supportive

H - Highly Related

## IX. SYLLABUS:

### UNIT I:

**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, applications, awk – execution, fields and records, scripts, operation, patterns, actions, Associative Arrays, String and Mathematical functions, system commands in awk, 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.

### UNIT II:

**Files and Directories:** File Concept, File System Structure, File metadata- Inodes, kernel support for files, System calls, Kernel support for files, System calls for File I/O Operations- open, creat, read, write, close, lseek, dup2, file, filestatus information- stat family, File and record locking-fcntl function, File permissions- chmod, fchmod, file ownership- chown, lchown, fchown, links- softlinks and hardlinks-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

### UNIT III:

**Process:** Process concept, Layout of a C program image in main memory, Process 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 and controlling Terminal, diff b/n threads and process

**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.

## UNIT IV:

**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), difference between unnamed and named pipes, pipe n and 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.

## UNIT V:

**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 and internet domain), Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection, Multiple simultaneous Clients, Socket options-stockpot and fcntl system calls, Comparison of IPC mechanisms.

### Text Books:

1. T.Chan, “UNIX System Programming using C++”, PHI.
2. Sumitabha Das, “UNIX Concepts and Applications”, 4e, TMH, 2006.
3. N.Matthew, R.Stones, Wrox, “Beginning Linux Programming”, 4e, Wiley India Edition.

### References:

1. Robert Love, “Linux System Programming”, O’Reilly.
2. W.R.Stevens, “UNIX Network Programming”, PHI.
3. Graham Glass, King Ables, “UNIX for programmers and users”, 3e, Pearson Education, 2003.
4. W.R.Stevens, “Advanced Programming in the Unix environment”, 2e, PHI, Pearson Education.
5. A.Hoover, “System Programming with C and UNIX”, Pearson.
6. Kumar Saurabh, “Unix Programming”, 1e, Wiley India pvt Ltd.
7. B.A.Forouzan and R.F.Gilberg, “UNIX and Shell programming”, Cengage Learning.

## X. COURSE PLAN:

At the end of the course, the students are able to achieve the following course learning outcomes:

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
1-3	A brief history of Linux, architecture of Linux, features of Linux	<b>Understand</b> history of Linux and its features	T2: 1.1, 2.1-2.2
4-6	<b>Linux utilities</b> -File handling utilities, Security by file permissions	<b>Use</b> system level scripts to create and manage simple file processing operations.	T2: 4.7-4.8, 5.3-5.4
7-8	<b>Process utilities</b> , disk utilities, Networking commands, Filters	<b>Illustrate</b> manipulating of information.	T2: 3.10,15.6, 17.5-17.6
9-10	Text processing utilities, Backup utilities	<b>Understand</b> restoring and retrieving text.	T2: 12.3-12.9 15.9-15.10
11-12	<b>sed</b> – scripts, operation, addresses, commands, applications	<b>Understand</b> two data buffers: the active pattern space.	T2: 13.4
13-14	<b>awk</b> – execution, fields and records, scripts, operation, patterns, actions,	<b>Demonstrate</b> pattern scanning and processing in problem	T2: 18.1-

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
	Functions, using system commands in awk.	solving.	18.12
15	Working with the Bourne again <b>shell</b> (bash): introduction, shell responsibilities, pipes and input Redirection, output redirection	<b>Understand</b> basic shell scripting.	T2: 8.5
16	Documents, running a shell script	<b>Understand</b> shell script execution.	T2: 14.14
17	The shell as a programming language, shell meta characters, file name substitution	<b>Classify</b> use special characters.	T2: 8.9
18	Shell command substitution, shell commands, environment, quoting, test command, control structures	<b>Illustrate</b> forwarding the command output into another context	T2: 8.4, 8.10
19	Arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.	<b>Develop</b> solutions to complex tasks.	T2: 14.5-14.17
20	<b>Files:</b> File Concept, file system structure, Inodes, file attributes, file types, library functions	<b>Demonstrate the</b> use of the formatting specifiers of io.	R4: 4.1-4.14
21	<b>The standard I/O</b> (fopen, fclose, fflush, fseek, fgetc, getc, getchar, fputc, putc, putchar, fgets, gets etc.)	<b>Demonstrate</b> standard stream and buffer based input and output system calls.	R4: 5.1-5.9
22	<b>Formatted I/O</b> , stream errors, kernel support for files	<b>Demonstrate</b> layout of what's being printed.	R4: 5.10-5.11
23	<b>File descriptors</b> ,low level file access open,create,read,write,close,lseek,stat family,umask,dup,dup2	<b>Demonstrate</b> modification and editing.	R4: 3.1-3.12, 4.2
24-26	fcntl, file and record locking	<b>Demonstrate</b> security concepts in files.	R4: 3.13
27-31	<b>File and directory management-</b> Directory file APIs, symbolic links and hard links.	<b>Discuss</b> scanning and linking methods.	R4: 4.20-4.22, 4.15-4.17
32-35	Process kernel support for process, process attributes, process hierarchy, process states, process composition, process creation and termination	<b>Understand</b> internal procedures and states of IPC	R4: 8.1-8.9
36-38	Zombie process, orphan process.	<b>Illustrate</b> daemons and varieties.	R4: 8.6
39-40	<b>Signals</b> – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function.	<b>Classify</b> processes to respond to asynchronous events.	R4: 10.1-10.3
41-42	Unreliable signals, reliable signals, kill, raise, alarm, Pause, abort, and sleep functions.	<b>Understand</b> and to handle exceptional situations.	R4: 10.4-10.19
43-45	<b>Interprocess Communication:</b> Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems	<b>Demonstrate</b> inter related process communication	R4: 14.1-14.4

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
46-52	pipes. FIFOs and examples.	<b>Demonstrate</b> named pipes.	R4: 14.5
53	Introduction to three types of IPC (Linux)-message queues, semaphores and shared memory.	<b>Discuss</b> types of restricting and accessing different resources.	R4: 14.6
54-55	<b>Message Queues</b> - Kernel support for messages, Linux APIs for messages, Client/server example.	<b>Demonstrate</b> dividing up work among to balance work over multiple processes.	R4: 14.7
56-57	<b>Semaphores</b> - Kernel support for semaphores	<b>Demonstrate</b> user variables and semaphore operations, provided at the kernel level.	R4: 14.8
58-59	File locking with semaphores. Linux APIs for semaphores	<b>Solve</b> security hurdles using programming interface of Linux	R4: 14.8
60	<b>Shared Memory</b> - Kernel support for shared memory	<b>Demonstrate</b> common memory portion which other processes	R4: 14.9
61	Linux APIs for shared memory, semaphore and shared memory	<b>Illustrate</b> common memory sharing interfacing example.	R4: 14.9
62	<b>Sockets</b> : introduction to sockets	<b>Understand</b> end to end network communication	R2: 15.1
63-64	Socket system calls for connection oriented protocol	<b>Understand</b> tcp based system calls	R2: 15.5
65-67	Connection less protocol, Multiple simultaneous clients	<b>Understand</b> udp protocol system calls	R2: 15.5
68-69	Example-client/server programs, Comparison of IPC Mechanisms	<b>Demonstrate connection</b> oriented, connectionless communications in two and three	R2: 15.5

**XI. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:**

Course Objectives	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I	H		H			S						S	H	S	
II	S	H	S	H						S			S	S	
III	H	H	S	S						H			H	H	S
IV		S				S							S	S	H
V	H	S			S					S			S		S
VI		S	H		S	H							H	H	H

**S - Supportive**

**H - Highly Related**

**XII. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:**

Course Outcomes	Program Outcomes												Program Specific Outcomes		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	H	S		H						S			S	H	
2	H			S	S	S							S	S	S
3		H			H								H	S	
4	S	S				H				H			H	H	S
5	S	S											S	H	
6	S			H		S				H		H	H	S	
7	H			H									H	H	
8	S	H				H				H			S	S	
9			H	S	S	S						H	S	H	
10	H		S	H						S			H	H	S
11	H			S	S	H							H	S	
12	H		H			S				S		S	S	H	H

**S - Supportive**

**H - Highly Related**

**Prepared by:** Mr. A Krishna Chaitanya, Associate Professor, IT

**HOD, IT**