INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad -500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTOR

Course Title	**LINUX INTERNALS**				
Course Code	AIT005				
Programme	B.Tech				
Semester	VI				
Course Type	Core				
Regulation	IARE - R16				
Course Structure	**Theory**			**Practical**	
	Lectures	**Tutorials**	**Credits**	**Laboratory**	**Credits**
	3	1	4	-	-
Chief Coordinator	Mr. A Krishna Chaitanya, Assistant Professor, IT				
Course Faculty	Mr. D Rahul, Assistant Professor, IT				

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.

II. COURSE PRE-REQUISITES:

Level	Course Code	Semester	Prerequisites	Credits
UG	ACS001	I	Computer Programming	4
UG	AIT003	IV	Computer Networks	4

III. MARKS DISTRIBUTION

Subject	SEE Examination	CIA Examination	Total Marks
Linux Internals	70 Marks	30 Marks	100

IV. DELIVERY / INSTRUCTIONAL METHODOLOGIES:

✔	Chalk & Talk	✔	Quiz	✔	Assignments	✔	MOOCs
✔	LCD / PPT	✔	Seminars	✘	Mini Project	✔	Videos
✘	Open Ended Experiments						

V. EVALUATION METHODOLOGY:

The course will be evaluated for a total of 100 marks, with 30 marks for Continuous Internal Assessment (CIA) and 70 marks for Semester End Examination (SEE). Out of 30 marks allotted for CIA during the semester, marks are awarded by taking average of two CIA examinations or the marks scored in the make-up examination.

Semester End Examination (SEE): The SEE is conducted for 70 marks of 3 hours duration. The syllabus for the theory courses is divided into five units and each unit carries equal weightage in terms of marks distribution. The question paper pattern is as follows. Two full questions with "either" or "choice" will be drawn from each module. Each question carries 14 marks. There could be a maximum of two sub divisions in a question.

The emphasis on the questions is broadly based on the following criteria:

50 %	To test the objectiveness of the concept.
50 %	To test the analytical skill of the concept OR to test the application skill of the concept.

Continuous Internal Assessment (CIA):
CIA is conducted for a total of 30 marks (Table 1), with 20 marks for Continuous Internal Examination (CIE), 05 marks for Quiz and 05 marks for Alternative Assessment Tool (AAT).

Table 1: Assessment pattern for CIA

Component	**Theory**		**Total Marks**
Type of Assessment	CIE Exam	Quiz/AAT	
CIA Marks	25	05	30

Continuous Internal Examination (CIE):
Two CIE exams shall be conducted at the end of the 8th and 16th week of the semester respectively. The CIE exam is conducted for 25 marks of 2 hours duration consisting of two parts. Part–A shall have five compulsory questions of one mark each. In part–B, four out of five questions have to be answered where, each question carries 5 marks. Marks are awarded by taking average of marks scored in two CIE exams.

Quiz / Alternative Assessment Tool (AAT):
Two Quiz exams shall be online examination consisting of 25 multiple choice questions and are be answered by choosing the correct answer from a given set of choices (commonly four). Marks shall be awarded considering the average of two quizzes for every course. The AAT may include seminars, assignments, term paper, open ended experiments, five minutes video and MOOCs.

VI. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		Strength	Proficiency assessed by
PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Presentation on real-world problems
PO 2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences	2	Seminars
PO 3	**Design / development of solutions**: 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.	2	Assignments
PO 5	**Modern tool usage**: 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.	3	Seminars

3 = High; 2 = Medium; 1 = Low

VII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSOs)		Strength	Proficiency assessed by
PSO 1	**Professional Skills:** The ability to understand, analyze and develop 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.	3	Assignments
PSO 2	**Software Engineering Practices:** 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.	2	Seminars
PSO 3	**Successful Career and Entrepreneurship:** 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.	-	-

3 = High; 2 = Medium; 1 = Low

VIII. COURSE OBJECTIVES (COs):

The course should enable the students to:	
I	Familiarize students with the Linux environment, and able to run commands on a standard Linux operating system.
II	Provide the skills needed to develop and customize Linux shell programs and to make effective use of a wide range of standard Linux programming and development tools.
III	Able to write moderate C programs utilizing common system calls.
IV	Develop the skills necessary for system programming and Inter and Intra process communication programming.

IX. COURSE OUTCOMES (COs):

COs	Course Outcome	CLOs	Course Learning Outcome
CO 1	Understand the basic commands of linux operating system and can write shell scripts.	CLO 1	Learn the importance of Linux architecture along with features.
		CLO 2	Identify and use Linux utilities to create and manage simple file and text processing operations.
		CLO 3	Develop shell scripts to perform more complex tasks in shell programming environment.
CO 2	Create file systems and directories and operate those using programs.	CLO 4	Illustrate file processing operations such as standard I/O and formatted I/O.
		CL05	Illustrate memory management of file handling through file/region lock.
		CLO 6	Design and Implement in C some standard linux utilities.
CO 3	Understand the processes background and fore ground by process and signals system calls.	CLO 7	Understand process structure, scheduling and management through system calls.
		CLO 8	Implement C programs to control process using system calls and identify difference between process and threads.
		CLO 9	Generalize signal functions to handle interrupts by using system calls.
CO 4	Create shared memory segments, pipes, message queues and can exercise inter process communication.	CL10	Design and implement inter process communication (IPC) in client server environment by using pipes and named pipes system calls.
		CLO 11	Design and implement inter process communication (IPC) in client server environment by using message queues systems calls.
		CLO 12	Illustrate client server authenticated communication in IPC through shared memory.
CO 5	Create sockets and semaphores to interact between process of different system.	CLO 13	Familiarity with Inter Process Communication using Semaphores
		CLO 14	Demonstrate various client server applications on network using TCP or UDP protocols.
		CLO 15	Design custom based network applications using the Sockets Interface inheterogeneous platforms.

X. COURSE LEARNING OUTCOMES (CLOs):

CLO Code	CLO's	At the end of the course, the student will have the ability to:	PO's Mapped	Strength of Mapping
AIT005.01	CLO 1	Learn the importance of Linux architecture along with features.	PO 1;PO 3	3
AIT005.02	CLO 2	Identify and use Linux utilities to create and manage simple file processing operations.	PO 2;PO 5	2
AIT005.03	CLO 3	Apply the security features on file access permissions by restricting the ownership using advance Linux commands.	PO 3	2
AIT005.04	CLO 4	Implement the SED,GREP and AWK commands for pattern matching and mathematical functions.	PO 1: PO 3	2
AIT005.05	CLO 5	Understand the shell responsibilities of different types of shells.	PO 2; PO 5	2
AIT005.06	CLO 6	Develop shell scripts to perform more complex tasks in shell programming environment.	PO 3	2

AIT005.07	CLO 7	Illustrate file processing operations such as standard I/O and formatted I/O.	PO 1	3
AIT005.08	CLO 8	Understand process structure, scheduling and management through system calls.	PO 2	2
AIT005.09	CLO 9	Generalize signal functions to handle interrupts by using system calls.	PO 1;PO 5	3
AIT005.10	CLO 10	Illustrate memory management of file handling through file/region lock.	PO 1;PO 3	3
AIT005.11	CLO 11	Design and implement inter process communication(IPC) in client server environment by using pipe, message queues, named Pipes.	PO 5	3
AIT005.12	CLO 12	Illustrate client server authenticated communication in IPC through semaphores and shared memory.	PO 2	2
AIT005.13	CLO 13	Demonstrate various client server applications on network using TCP or UDP protocols.	PO 3	3
AIT005.14	CLO 14	Design custom based network applications using the Sockets Interface in heterogeneous platforms.	PO 1;PO 5	3
AIT005.15	CLO 15	Understand the comparison of various Inter Process Communication Mechanisms.	PO 1	3

3 = High; 2 = Medium; 1 = Low

XI. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES

Course Outcomes (COs)	Program Outcomes and Program Specific Outcomes					
	PO 1	PO 2	PO 3	PO 5	PSO 1	PSO 2
CO 1	3	2	2	3	3	
CO 2	3	2	2	3	3	2
CO 3	3		2	3	3	
CO 4	3	2	3	3	3	
CO 5	3		3	3	3	2

3 = High; 2 = Medium; 1 = Low

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

Course Learning Outcomes (CLOs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 1	3		3												
CLO 2		2			3								3		
CLO 3			2												

Course Learning Outcomes (CLOs)	Program Outcomes (POs)												Program Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CLO 4	3		2												
CLO 5		2			3								3		
CLO 6			2											2	
CLO 7	3												3		
CLO 8			2										3		
CLO 9	3				3										
CLO 10	3		3												
CLO 11					3								3		
CLO 12		2													
CLO 13			3										3		
CLO 14	3				3										
CLO 15	3													2	

3 = High; 2 = Medium; 1 = Low

XIII. ASSESSMENT METHODOLOGIES–DIRECT

CIE Exams	PO 1, PO 2, PO 3, PO 5, PSO1, PSO2	SEE Exams	PO 1, PO 2, PO 3, PO 5, PSO1, PSO2	Assignments	PO 1, PO 2, PO 3, PO 5	Seminars	PO 2
Laboratory Practices	PO 1	Student Viva	-	Mini Project	-	Certification	-
Term Paper	-						

XIV. ASSESSMENT METHODOLOGIES-INDIRECT

✔	Early Semester Feedback	✔	End Semester OBE Feedback
✘	Assessment of Mini Projects by Experts		

XV. SYLLABUS

UNIT-I	LINUX UTILITIES AND BOURNCE AGAIN SHELL (bash)
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; Sed: Scripts, operation, addresses, commands; Awk: Execution, fields and records scripts, operation, patterns, actions, applications; Working with the Bourne again shell(bash): Introduction, shell responsibilities, pipes and input Redirection, output 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 SYSTEM CALLS

Files and Directories: File Concept, File types, File System Structure, File metadata- Inodes, kernel support for files, file 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, fchown, links- soft links & hard links- symlink, link, ulink.
Directories: creating, removing and changing directories- mkdir, rmdir, chdir, obtaining current working directory- getcwd, directory contents, scanning directories- opendir, readdir, closedir, rewind dir functions.

UNIT-III	PROCESS AND SIGNALS

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.

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

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), differences between unnamed and named pipes, popen & 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 AND SOCKETS

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, Socket options - setsockopt and fcntl system calls, Comparison of IPC Mechanisms.

Text Books:

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", 1st Edition, Pearson Education, New Delhi, India, 2005.

Reference Books:

1. T. Chan, "Unix System Programming using C++", PHI.
2. N. Mathew, R. Stones, Wrox, "Beginning Linux Programming", 4th Edition, Wiley India Edition.
3. Graham Glass, King Ables, "Unix for Programmers and Users", 3rd Edition, Pearson Education.
4. A. Hoover, "System Programming with C and Unix", Pearson Education.
5. K. A. Robbins, "Unix System Programming, Communication, Concurrency and Threads", Pearson Education.
6. S. G. Kochan and P. Wood, "Unix Shell Programming", 3rd Edition, Pearson Education.
7. B. A. Forouzan and R. F. Gilberg, "Unix and Shell Programming", Cengage Learning.
8. Robert Love, "Linux System Programming", O'Reilly, SPD.

XVI. COURSE PLAN:

The course plan is meant as a guideline. Probably there may be changes.

Lecture No	Topic/s to be covered	Course Learning Outcomes (CLOs)	Reference
1-2	Understand history of Linux and its features	CLO 1	T2: 1.1, 2.1-2.2
3-4	Use system level scripts to create and manage simple file processing operations.	CLO 1	T2: 4.7-4.8, 5.3-5.4
5-6	Illustrate manipulating of information.	CLO 2	T2: 3.10,15.6, 17.5-17.6
7-8	Understand restoring and retrieving text.	CLO 2	T2: 12.3-12.9 15.9-15.10
9	Understand two data buffers: the active pattern space.	CLO 2	T2: 13.4
10-11	Demonstrate pattern scanning and processing in problem solving.	CLO 2	T2: 18.1, T2:18.12
12-13	Understand basic shell scripting.	CLO 3	T2: 8.5
14	Understand shell script execution.	CLO 3	T2: 14.14
15	Classify use of special characters.	CLO 3	T2: 8.9
16-17	Illustrate forwarding the command output into another context	CLO 3	T2: 8.4, 8.10
18-19	Develop solutions to complex tasks.	CLO 4	T2: 14.5-14.17
20	Demonstrate the use of the formatting Specifies of IO.	CLO 4	R4: 4.1-4.14
21-22	Demonstrate standard stream and buffer based input and output system calls.	CLO 4	R4: 5.1-5.9
23	Demonstrate layout of what's being printed.	CLO 4	R4: 5.10-5.11
24-25	Demonstrate modification and editing.	CLO 4	R4: 3.1-3.12, 4.2
26-27	Demonstrate security concepts in files.	CLO 4	R4: 3.13
28	Discuss scanning and linking methods.	CLO 5	R4: 4.20-4.22, 4.15-4.17
29-31	Understand internal procedures and states of IPC	CLO 5	R4: 8.6
32-33	Illustrate daemons and varieties.	CLO 6	R4: 8.6
34	Classify processes to respond to asynchronous events.	CLO 7	R4: 10.1-10.3
35-36	Understand and to handle exceptional situations.	CLO 8	R4: 10.4-10.19
37-38	Demonstrate inter related process communication	CLO 9	R4: 14.1-14.4
39	Demonstrate named pipes.	CLO 10	R4: 14.5
40	Discuss types of restricting and accessing different resources.	CLO 11	R4: 14.6
41-43	Demonstrate dividing up work among to balance work over multiple processes.	CLO 11	R4: 14.7
44	Demonstrate user variables and semaphore operations, provided at the kernel level.	CLO 12	R4: 14.8

Lecture No	Topic/s to be covered	Course Learning Outcomes (CLOs)	Reference
45-46	Solve security hurdles using programming interface of Linux	CLO 12	R4: 14.8
47	Demonstrate common memory portion which other processes	CLO 12	R4: 14.9
48-49	Illustrate common memory sharing interfacing example.	CLO 12	R4: 14.9
50-51	Demonstrate parallelism in Linux based system calls.	CLO 12	T1: 13.1-13.2
52	Demonstrate concurrency in Linux APIs.	CLO 13	T1: 13.4
53-54	Demonstrate multiple processes to a common resource in Linux based parallel	CLO 13	T1: 13.5
55-57	Demonstrate multiple threads access the same resource for read and write.	CLO 13	T1: 13.5
58	Understand end to end network communication	CLO 14	R2: 15.1
59-60	Understand TCP based system calls	CLO 14	R2: 15.5
61-62	Understand UDP protocol system calls	CLO 14	R2: 15.5
63-64	Demonstrate connection oriented, connectionless communications in two and three	CLO 15	R2: 15.5

XVII.GAPS IN THE SYLLABUS - TO MEET INDUSTRY / PROFESSION REQUIREMENTS:

S No	Description	Proposed actions	Relevance with POS	Relevance with PSOS
1	Familiarizing different types of Linux Internals.	Seminars /NPTEL	PO 1,PO 2,PO 5	PSO 2
2	Familiarizing different types of shells in Linux Internals.	Assignments/ NPTEL	PO 3	PSO 1
3	Implementation of commands using system calls and learning Linux administration commands.	Seminars / Guest Lectures / NPTEL	PO 5	PSO 2
4	Familiarizing of inter process communication between unrelated process.	Seminars/ Guest Lecturers	PO 1	PSO 2

Prepared by:
Mr. A Krishna Chaitanya, Assistant Professor

HOD, IT