

LINUX PROGRAMMING

Subject Code: A70517

Regulations: R15 - JNTUH

Class: IV Year B.Tech CSE I Semester



Department of Computer Science and Engineering

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Ibrahimpatnam - 501 510, Hyderabad

LINUX PROGRAMMING (A70517)

COURSE PLANNER

I. COURSE OVERVIEW

At the end of the course the student will be in a position to –

1. Use basic fundamental utilities which are required again and again on daily basis to work on a modern operating system.
2. Write useful shell scripts which greatly and effectively enhance the usefulness of computers, from the point of view of programmers and application developers.
3. Understand basics of various OS related concepts, from programmer's point of view, like files, directories, kernel, inodes, APIs, system calls, processes, signals, etc.
4. Develop applications where several processes need to communicate with each other to complete a task.
5. Use different IPC ways in their programs like Message Queues, Semaphores, and Shared Memories.
6. Write programs which employs advanced concepts like multithreading.
7. Write useful programs for networking purposes.

II. PRE-REQUISITES

This is practical course which encompasses knowledge gained in various subjects and allows seeing the concepts in practice. As such, the pre-requisites include many subjects, including, Operating Systems, Computer Organization, Computer Networks, C Programming, Data Structures, etc.

III. COURSE OBJECTIVE

S.N	OBJECTIVES
0	
1	To understand and make effective use of linux utilities and shell scripting language to solve problems
2	To implement in C some standard linux utilities like mv,cp,ls etc...
3	To Develop the skills the necessary for systems programming including file system programming,process and signal management and interprocess communication
4	To develop the basic skills required to write network programs using sockets

IV. COURSE OUTCOMES:

S.No	Description	Blooms level of taxonomy
1	Students will be able to understand the basic commands of linux operating system and can write shell scripts	KNOWLEDGE
2	Students will be able to create file systems and directories and operate them	UNDERSTAND
3	Students will be able to create processes background and fore ground etc..by fork() system calls	SYNTHESIS
4	Students will be create shared memory segments,pipes ,message queues and can exercise interprocess communication	SYNTHEIS

V. HOW PROGRAM OUTCOMES ARE ASSESSED

Program Outcomes (PO)		Level	Proficiency assessed by
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems related to Computer Science and Engineering.	3	Assignments
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems related to Computer Science and Engineering and reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	3	Assignments
PO3	Design/development of solutions: Design solutions for complex engineering problems related to Computer Science and Engineering 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
PO4	Conduct investigations of complex problems: 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.	3	Assignments
PO5	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.	2	Lab experiments
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the Computer Science and Engineering professional engineering practice.	-	--
PO7	Environment and sustainability: Understand the impact of the Computer Science and Engineering professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	-	--
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	-	--
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	-	--
PO10	Communication: 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.	-	--

PO11	Project management and finance: 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.	1	Sample Project
PO12	Life-long learning: 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.	1	Assignments

VI. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED

PROGRAM SPECIFIC OUTCOMES		Level	Proficiency assessed by
PSO1:	To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.	2	Lectures, Assignments
PSO2:	The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.	3	Projects
PSO3:	The ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process. Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research	2	Guest Lectures

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) - : None

SYLLABUS

Unit -I:

Linux Utilities:

File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities, Backup utilities Sed - Scripts, Operations, Addresses, Commands,,awk - Execution, Fields and Records, Scripts, Operations, Actions, Associative Array, Strings 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, 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 concepts, File types File system structure,file metadata - Inodes, kernel support for files, System calls for the file I/O operations- open,create,read,wirte,close,lseek,dup2,file status information-stat family, file and record locking-fcntl function, file permissions- chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links- symlink, link, unlink.

Directories:

Creating,,removing and changing Directories-mkdir,rmdir,chdir,obtaining current working directory-getcwd,directory contents,scanning directories- opendir, readdir, rewind functions.

Unit- III:**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 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, differences between threads and processes.

Signals:

Introduction to signals, Signal generation, Signal handling, Kernel support for signals, signal function, Unreliable signals, Reliable signals, Signal functions: kill, raise, alarm, pause, abort, sleep.

Unit- IV:**Inter process Communication:**

Introduction to IPC,IPC between processes on a single computer system, IPC between processes on different systems, Pipes-creationIPC between related processes using FIFOs(Named pipes), differences between unnamed and named pipes, popen 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 Berkley 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- setsockopt and fcntl system calls, comparison of IPC mechanisms.

RELEVANT SYLLABUS FOR GATE

Not Applicable

RELEVANT SYLLABUS FOR IES

Not Applicable

LESSON PLAN

Lecture	Week	Topic to be covered	Learning out come	Referenc e
UNIT-I				
1.	1	Linux Utilities: File handling utilities, Security by file permissions	Introducing LINUX	T2
2.		Process utilities, Disk utilities	Basic commnds	
3.		Networking commands, Filters	Basic commands	
4.		Text processing utilities, Backup utilities	Basic ocmmands	
		Tutorial / Overview, Pre-requisites, Explanation (outline) of Course		

Lecture	Week	Topic to be covered	Learning out come	Reference
5.	2	sed - scripts, operations, addresses, commands, applications	Explaining sed scripts	T2
6.		awk - execution, fields and records, scripts	Explaining awk scripts	
7.		awk - operations, patterns, actions, functions	Explaining awk scripts	
8.		awk - using system commands in awk	Explaining awk scripts	
		Tutorial / Bridge Class		
9.	3	Shell Scripting: Introduction to shell, Shell responsibilities, Pipes	Shell working	T2
10.		here documents, Input redirection, Output redirection	Commands of shell	
11.		Running a shell script, Shell as a programming language	Shell scripts	
12.		Shell meta characters, File-name substitution	Shell characters	
		Tutorial / Bridge Class		
13.	4	Shell variables, Command substitution	Usage of shell variables	T2
14.		Shell commands, The environment, Quoting, test command	Shell commands	
15.		Shell control structures, Arithmetic in shell, examples	Shell programming	
16.		Interrupt processing functions, Debugging shell scripts	Debugging of shell programs	
		Tutorial / Mock Test I		
UNIT-II				
17.	5	Files: File concepts, File system structure	Unix file system	T2
18.		Inodes, File attributes, File types	Unix file system	
19.		Standard I/O in C, Formatted I/O in C, Stream errors	Low level file access	
20.		Kernel support for files, System calls, Library functions	Unix system calls	
		Tutorial / Bridge Class	exercise	
21.	6	File descriptors, Low level file access	File descriptors definition	T2
22.		File structure related system calls: open, creat, read, write, close, lseek	System calls	
23.		File related system calls: stat family, umask, dup, dup2, fcntl	System calls explanation	
24.		File & record locking, File and directory management, Symbolic links and hard links	Creating locks for files	
		Tutorial / Bridge Class	exercise	
25.	7	Directory related system calls:	Basic system calls	T2

Lecture	Week	Topic to be covered	Learning out come	Reference
		chmod, chown, links		
26.		Directory related system calls: mkdir, rmdir, chdir, getcwd, Scanning Directories	Basic system calls	
27.		Directory related functions: opendir, readdir, closedir,rewinddir	Basic system calls	
28.		Directory related functions: seekdir, telldirfunctions	Basic system calls	
		Tutorial / Bridge Class		
UNIT-III				
29.		Process: Concept of processes in Linux, layout of a c program image, process environment variables,	Defining a process	
30.	8	Kernel support for process Process identification, Process control - Creation, Waiting for process, Termination, replacing a process image	Creating a process	T2
31.		Zombie process, Orphan process, Process APIs	Creating zombies	
32.		System calls for process management: fork, vfork, exit, wait, waitpid, exec family, system	Forking a child process	
		Tutorial / Bridge Class		
	9	Mid-I Examinations		
UNIT-III cont'd.				
33.	10	Differences between threads and processes	Differentiate process and threads	T2
34.		Signals: Introduction to signals, Signal generation, Signal handling,	Explaining signals	
35.		Kernel support for signals, signal function	System support to signals	
36.		Unreliable signals, Reliable signals	Relaible signals	
		Tutorial / Bridge Class	exercise	
UNIT-IV				
37.	11	Kill,raise,alarm	Kill() system calls	T2
38.		Pause,abort,sleep functions	Aborting a process	
39.		Interprocess Communication-introduction, IPC between processes on a single system	IPC definition	
40.		IPC between processes on different systems- FIFO pipes - introduction	Named pipes creation	
		Tutorial / Bridge Class	exercise	
41.	12	IPC between unrelated processes using FIFO, difference between named and unnamed pipes	Explaining diff between named and unnamed pipes	T2

Lecture	Week	Topic to be covered	Learning out come	Reference
42.		Popen and pclose library functions, kernel support for messages	Pipe related system calls	
43.		API for message queues	APIs explanation related to mesg queues	
44.		Client / server example	Test case	
		Tutorial / Bridge Class	exercise	
UNIT-V				
45.	13	Kernel support for semaphores-introduction	Introduction to semaphores	T2
46.		Kernel support for semaphores cont'd		
47.		API for semaphores	Semop() system calls	
48.		Summary	Exercising	
		Tutorial / Bridge Class	Exercising	
49.	14	File locking semaphores	Creating locks using semaphores	T2
50.		Kernel support for shared memory	Shared memory creation	
51.		API for shared memory	Shmat(),shmdt()	
52.		Example	exercise	
		Tutorial / Bridge Class	exercise	
53.	15	Sockets: Introduction to Berkeley Sockets	Introduction to sockets	T2
54.		IPC over a network	Linkage of IPC	
55.		Client server model		
56.		Socket address structures	System calls related to sockets	
57.	16	Socket system calls for connection oriented	System calls related to sockets	T2
58.		Socket system calls for connectionless protocol	System calls related to sockets	
59.		Examples of client – server programs	Example program	
60.		Socket options, comparison of IPC mechanisms	IPC exercising for sockets	
61.	17	Multiple simultaneous clients	Example of simultaneous clients	T2
62.		Socket options	Sockets explanaton	
63.		Comparison of IPC mechanisms	Comparision of IPC mechanisms	
64.		Revision		
		Mid-II Examinations		

SUGGESTED BOOKS

Text Books

1. Unix System Programming using C++, T.Chan, PHI (Unit III to Unit VIII)
2. Unix Concepts and Applications, 4th Ed, Sumitabha Das, TMH
3. Unix Network Programming, W.R.Stevens,PHI.

Reference Books

1. Begining Linux Programming, 4th Edition, N. Matthew, R.Stones, Wrox, Wiley India Edition.
2. Unix for Programmers 3rd Ed, Graham Glass & King Ables, Pearson Education.
3. System Programming with C and Unix, A.Hoover, Pearson.
4. Unix System Programming, communication, concurrency and Threads, K.A. Robbins and S.Robbins, Pearson Education.
5. Unix Shell Programming, S.G. Kochan and P.Wood, 3rd edition, Pearson Education.
6. Shell Scripting, S.Parker, Wiley India Pvt. Ltd.
7. Advanced Programming in the Unix Environment, 2nd Ed, W.R.Stevens, Pearson Education
8. Unix and Shell Programming, B.A.Forouzan&R.F.Gilberg,Cengage Learning
9. Linux System Programming, Robert Love, O'Reily, SPD.
10. C Programming Language, Kernighan and Ritchie, PHI

X. CO-POS MAPPING

CO'S	Program Outcomes (PO's)											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1. Identify and use Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security.	2	-	3	2	2	-	1	-	3	2	1	-
CO2. Work confidently in Linux environment.	3	-	1	2	3	-	2	-	1	3	3	-
CO3. Work with shell script to automate different tasks as Linux	3	-	1	2	3	-	1	-	3	2	3	-
CO4. Illustrate file processing operations such as standard I/O and formatted I/O.	2	-	3	2	3	-	3	-	3	2	3	-
CO5. Design various client server applications using TCP or UDP protocols.	3	-	3	2	1	-	1	-	3	3	3	-

QUESTION BANK

Unit - I

S. No	Question	Blooms level
1	Explain the significance of the following commands) a)ls b)ls -ld	KNOWLEDGE
2	How to remove duplicate lines from a file using sort	KNOWLEDGE
3	Discuss the various backup utilities available in Linux	KNOWLEDGE
4	What is a shell? What are the responsibilities of a shell?	KNOWLEDGE
5	How to count the number of ordinary files in home directory tree?	UNDERSTAND
6	Write a short note on here documents	UNDERSTAND
7	Write a shell script using “sed” command	UNDERSTAND

Long Answer Questions:

S. No	Question	Blooms level
1	Explain basic file attributes. Discuss the appropriate commands associated with them.	ANALYZE
2	Write a shell program for differentiating the commands Comm.,cmp	UNDERSTAND
3	Explain awk with necessary examples	KNOWLEDGE
4	Write a shell script which receives two filenames as arguments. It should check whether the two file's contents are same or not. If they are same then second file should be deleted	ANALYZE
5	Write a shell script which will receive, as run-time arguments, either the filename or the filename with absolute path. This script should obtain information about this file as given by ls -l and display in proper format	ANALYZE
6	Compare and contrast while and until loops	UNDERSTAND
7	What is a shell? What are the responsibilities of a shell? What are the different types of shells available in Linux?	KNOWLEDGE
8	Write short notes on following commands: trap, export, exit, expr, break,continue	L1:KNOWLEDGE

Unit- II:

Short Answer questions:

S. No	Question	Blooms level
1	Give the syntax of fcntl() system call	KNOWLEDGE
2	Write short notes on hard links and symbolic links.(KNOWLEDGE
3	Explain file descriptors and i-node?(UNDERSTAND
4	Explain the difference between fgetc and getc system calls	ANALYZE
5	Explain in detail about lseek function	UNDERTSAND
6	Give syntax for chmod and fchmod functions and state their importance	ANALYZE

Long Answer Questions:

S. No	Question	Blooms level
1	Give the syntax of creat() system call. Explain how this function is equivalent to open function	KNOWLEDGE
2	What are stat(), fstat() and lstat() functions? Explain	UNDERSTAND
3	Write the syntax of following system calls: open(), creat(), read(), write(), chmod(),chown(), unlink(), link()	UNDERSTAND
4	Create a shared lock using system calls	ANALYZE

Unit- III

Short Answer Questions:

S. No	Question	Blooms level
1	Give the syntax and return values of fork() system call	UNDERSTAND
2	Write short notes on Abort system call	KNOWLEDGE
3	Obtain the status of a process using KILL system call	APPLY
4	Write the differences between wait and waitpid()system calls.	ANALYZE

Long Answer Questions:

S. No	Question	Blooms level
1	Write a program to illustrate the vfork ()	KNOWLEDGE
2	Write a C program to illustrate the use SIG KILL,SIG INT	APPLY
3	Explain how a prcess can be created and terminated in LINUX operating system	APPLY
4	What is a process? Explain waiting for a process in detail	UNDERSTAND
5	Explain each for argument: kill(), raise(),alarm(), exit()	APPLY

Unit -IV**Short Answer Questions:**

S. No	Question	Blooms level
1	Write a C program to create and unnamed pipe using pipe() system call	CREATE
2	Give the differences between named and unnamed pipes.	ANALYZE
3	What are popen() and pclose() library functions.	UNDERSTAND
4	Write a C program to illustrate the system call Semop()	UNDERSTAND

Long Answer Questions:

S. No	Question	Blooms level
1	What is Inter-Process Communication? Explain the IPC methods supported by the linux.	UNDERSTAND
2	Explain Linux kernel support for messages.	ANALYZE
3	Explain in detail the four API's four message manipulation	APPLY
4	What is a Semaphore? Explain the kernel support for Semaphores	UNDERSTAND
5	Explain in detail the Linux API for Semaphores.	UNDERSTAND

Unit- V**Short Answer Questions:**

S. No	Question	Blooms level
1	Explain the differences between threads and processes.	UNDERSTAND
2	Write short notes on threads and light weight process	ANALYZE
3	Explain in brief about the function used to create threads.	APPLY
4	What are thread attributes?	UNDERSTAND
5	Write short notes on mutex.	UNDERSTAND

Long Answer Questions:

S. No	Question	Blooms level
1	Explain the thread synchronization using Semaphores and Mutexes	UNDERSTAND
2	What is Shared Memory and explain the kernel support for Shared Memory	ANALYZE
3	Explain in detail the Linux API for Shared Memory	ANALYZE
4	What is a socket? Explain in detail about the socket addresses	UNDERSTAND
5	Explain in detail the system calls used for connection oriented and connectionless protocol.	UNDERSTAND

B. Objective:

UNIT- I

1. What command is used to count the total number of lines, words, and characters contained in a file? []
a) countw b) wcount c) wc d) count p
2. What command is used to sort the lines of data in a file in alphabetical order? []
a) sort - r b) st c) sh d) sort
3. Which command is used to clear the screen? []
a) clear b) %clear c) %clrscr d) either a or c
4. What command is used to remove the directory? []
a) rdir b) remove c) rd d) rmdir
5. Which character is used to match exactly one character? []
a) | b) * c) ^ d) ?

UNIT- II

1. If command checks []
a) Exit status of a command b) value of a command
c) Checks condition true or false d) a
2. The first thing that is searched when a command references a file is its []
a) i-node b) inode number c) permission setting d) none
3. Which of the following contains other files in the filesystem. []
a) ordinary file b) device file c) directory d)none
4. Which of the following command is used to remove the directory from a file system?
a) mkdir b) rmdir c) cd d)none
5. The command chmod is used for
a) creating a file b) deleting afile c) changing permissions d)none

UNIT- III

1. Differences between system calls and library function []
a) System calls are part of OS b) Library calls are adds-on
c) library functions invoke system calls d) all
2. Fork() is used for
a) creating a new process b) deleting the parent process
c) to know the process id d) none of the above
3. A process which doesnot have a parent process is called []
a) zombie b)orphan c) child process d)none

4. Which of the following signal resumes the working of the process for a specified time interval []

a) abort b) sleep c) wait d) waitpid

5. Each thread has a unique []

a) Thread ID b) set of registers c) stack pointer d) all

UNIT- IV

1 In binary semaphores value of semaphore = 0 []

a) resource is locked b) resource is available c) resource is unavailable d) none

2. Which of the following is not a System V IPC mechanism? []

a) semaphores b) shared memory c) message queues d) fifos

3. The IPC mechanism that is used to provide locking facility is []

a) semaphores b) shared memory c) pipes d) fifos

4. The system call that is used to alter the permissions and other characteristics of a shared memory segment. []

a) shmget() b) shmat() c) shmctl() d) shmdt()

5. msgsnd() []

a) To access the member of message queue b) To create message queue

c) To send message to queue d) To receive message from queue

UNIT- V

1. Mutual Exclusion is done by using []

a) semop() b) shmop() c) msgop() d) semctl()

2. _____ sockets exhibits a behavior with the read and write functions that differ from normal file I/O. []

a) datagram b) raw sockets c) **stream** d) sequenced packets

3. The function that assigns a local protocol address to a socket is []

a) accept() b) **bind()** c) connect() d) listen()

4. TELNET used _____ protocol for data connection. [TCP]

5. _____ is added to data packet for error detection. [checksum & paritybit]

WEBSITE ADDRESSES

1. www.advancedlinuxprogramming.com

2. www.tldp.org

3. www.gnu.org

4. www.kernel.org

5. developer.intel.com

6. www.linuxsecurity.com

EXPERTS DETAILS

1. Dr. M. Srinivasa Rao, Professor, JNTU, Hyderabad

2. Dr. M.Sadanandam, KU Wrangal.

JOURNALS (NATIONAL & INTERNATIONAL)

1. The Growing Role of UEFI Secure Boot in Linux Distributions [May 21, 2014 By Mark Doran]

2. A Cool Project for Microsoft: Adopt Linux [May 02, 2014 By Doc Searls]

3. LINUX Based Cloud Operating System [Jan 01, 2012 by Lokesh Patel, Gajendra Singh]

LIST OF TOPICS FOR STUDENT SEMINARS

1. File and directory maintenance

2. Process creation using fork, vfork()

3. Introduction to IPC

4. Semaphores

5. Shared memory example

6. Thread synchronization
7. Pipes and FIFOs

CASE STUDIES / SMALL PROJECTS

Case studies

1. Create process and threads and establish communication among them.
2. **Create message queues and check how they are used in synchronization**

Small Projects :(<http://www.unc.edu/~feiner/prog/>)

1. Graphing calculator.
2. Cellular automata simulator.
3. Virtual machine and assembler.
4. Console-based maze game Hexeditor.
5. Minesweeper clone without clock or score, but with three levels of bombs.