

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**  
**M.Tech – CN/CNIS - I Year - II Sem**

**NETWORK PROGRAMMING LAB**

**Course Objectives:**

- To gain hands-on experiences in installing and administering computer systems and networks, in particular, the UNIX version.
- To implement networking and Internet protocols via programming and TCP/IP protocol architecture; user datagram protocol.
- TO implement shell script that accepts a list of files.

**LIST OF SAMPLE PROBLEMS/EXPERIMENTS:**

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script that accepts any number of arguments and prints them in the reverse order.
7. Write a shell script that determines the period for which a specified user is working on the system.
8. Write a shell script to list all of the directory files in a directory.
9. Write an interactive file-handling shell program- Let it offer the user the choice of copying, removing or linking files. Once the user has made a choice, have the program ask him for the necessary information such as the file name, new name and so on.
10. Write a shell script to find factorial of a given integer.
11. Write a shell script to find the G.C.D. of two integers.
12. Write a shell script to generate a multiplication table.
13. Write a shell script that copies multiple files to a directory.
14. Write a shell script that counts the number of lines and words present in a given file. \*15. Write a shell script that displays the list of all files in the given directory.
15. Write a shell script (small calculator) that adds, subtracts, multiplies and divides the given two integers. There are two division options: one returns the quotient and the other returns remainder. The script requires 3 arguments: The operation to be used and two integer numbers. The options are add (-a), subtract (-s), multiply (-m), quotient (-c) and remainder (-r).
16. Write a shell script to reverse the rows and columns of a matrix.
17. Write a sed command that deletes the first character in each line in a file.
18. Write sed command that deletes the character before the last character in each line a file.
19. Write a sed command that swaps the first and second words in each line of a file.
20. Write an awk script that reads a file of which each line has 5 fields – ID, NAME, MARKS1, MARKS2, MARKS3 and finds out the average for each student. Print out the average marks with appropriate messages.

21. Write an awk script to find the factorial of a user supplied number.
22. ls -l command produces long listing of files.
23. Write an awk script 1) to print the selected fields (Ex: size and name of the files) from the file listing. 2) to print the size of all files and number of files.
24. Write an awk script to count the number of lines in a file that do not contain vowels.
  25. Write an awk script to find the number of characters, words and lines in a file.
  26. Write a c program that makes a copy of a file using
    - a. Standard I/O
    - b. System calls.
  27. Write a C program that counts the number of blanks in a text file
    - a. Using standard I/O
    - b. Using system calls
  28. Implement in C the following UNIX commands using system calls
    - a. cat
    - b. ls
    - c. mv
  29. Write a program that takes one or more file/directory names as command line input and reports the following information on the file.
    - a. File type.
    - b. Number of links.
    - c. Time of last access.
    - d. Read, Write and Execute permissions.
  30. Write a c program to emulate the UNIX ls -l command.
  31. Write a c program that creates a directory, puts a file into it, and then removes it.
  32. Write a c program that searches for a file in a directory and reports whether the file is present in the directory or not.
33. Write a c program to list for every file in a directory, its inode number and file name.
34. Write a c program that creates a file containing hole which is occupying some space but having nothing.
35. Write a c program that demonstrates redirection of standard output to a file.  
Ex: ls > f1.
36. Write a c program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen.
37. Write a c program to create a Zombie process.
38. Write a c program that illustrates how an orphan is created.
39. Write a c program that creates a child process to execute a command. The command to be executed is passed on the command line.
40. Write a c program that accepts two small numbers as arguments and then sums the two numbers in a child process. The sum should be returned by child to the parent as its exit status and the parent should print the sum.
41. Write a c program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls -l | sort
42. Write c programs that illustrate communication between two unrelated processes using named pipe.
43. Write a c program in which a parent writes a message to a pipe and the child reads the message.
44. Write a c program that illustrates suspending and resuming processes using signals.
45. Write a c program that displays the real time of a day every 60 seconds, 10 times.
46. Write a c program that runs a command that is input by the user and prints the exit status if the command completes in 5 seconds. If it doesn't, then the parent uses kill to send a SIGTERM signal to kill the child process.
47. Write a C program that illustrates file-locking using semaphores.
48. Write a C program that implements a producer-consumer system with two processes. (Using semaphores).
49. Write client and server programs (using C) for
  - a. Interaction between server and client processes using Unix Domain Sockets.

- b. Interaction between server and client processes using Internet Domain Sockets.
50. Write a C program (sender.c )
    - a. To create a message queue with read and write permissions.
    - b. To write 3 messages to it with different priority numbers.
  51. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in 63.a) and displays them.
  52. Write C program that illustrates two processes communicating via shared memory.
  53. Design TCP iterative Client and server application to reverse the given input sentence
  54. Design TCP iterative Client and server application to reverse the given input sentence
  55. Design TCP client and server application to transfer file
  56. Design a TCP concurrent server to convert a given text into upper case using multiplexing system call "select"
  57. Design a TCP concurrent server to echo given set of sentences using poll functions
  58. Design UDP Client and server application to reverse the given input sentence
  59. Design UDP Client server to transfer a file
  60. Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.
  61. Design a RPC application to add and subtract a given pair of integers

**TEXT BOOKS:**

1. Advance Unix Programming Richard Stevens, Second Edition Pearson Education
2. Advance UNIX Programming, N.B. Venkateswarlu, BS Publication.
3. UNIX and Shell programming, B.A.Forouzan and R.F.Gilberg, Thomson.
4. UNIX and Shell Programming, M.G. Venkatesh Murthy, Pearson Education.
5. UNIX Shells by Example, 4th Edition, Ellie Quigley, Pearson Education.