

## Linux Programming Lab

### Objectives:

To give a practical orientation of programming in Linux environment using system calls and advanced concepts in unix programming

### PROGRAMS LIST:

1. Write C programs that uses open, read, write system calls.
2. Write C programs that differentiates FILE \*( file stream pointers in C standard library) and file descriptors by using functions such as fdopen, fileno.
3. Write a C program which displays a given files meta data by using stat system call and st\_mode structure.
4. Write a C program which lists all the files of current working directory whose size is more than given number of data blocks.
5. Write a C program which lists all the files of current working directory which contains hard link files.
6. Write a C program to emulates file system checking utility (fsck command) using system calls.
7. Example C program which supports that child process inherits environment variables, command line arguments, opened files.
8. Simple C programs to have process trees and process chains.
9. Simple C program that demonstrates the failure of fork system call because of crossing system limits.
10. Simple C programs to demonstrate the use of pipe system call for inter process communication and also emulating piping in shell.
11. Simple C programs to demonstrate the use of popen standard library function call for inter process communication and also emulating piping in shell.
12. Simple C program to use named pipes for inter process communication.
13. Simple C programs to illustrate the use of exec family of functions.
14. Write a C program which emulates simple shell.
15. Write C program to create a thread using pthreads library and let it run its function.
16. Write a C program to illustrate concurrent execution of threads using pthreads library.
17. Write a C program to simulate pthread\_create function failure by repeatedly calling the same.
18. Write a C program which creates a thread using pthread and passes arguments to the thread function.
19. Write C programs which uses sigset, sigfillset, sigprocmask, related system calls and structures.
20. Write a C program to simulate memory segment violation run time error and implement a signal handler (both reliable and unreliable) which handles situation.
21. Write a C program to illustrate the use of sbrk system call.
22. Write a C program to illustrate inter process communication via message queues.
23. Write a C program to illustrate inter process communication via shared memory.
24. Write a C program to simulate producer and consumer problem using semaphores, shared memory, and fork.
25. Write a C program to simulate producer and consumer problem using semaphores, shared memory, and pthread\_create.
26. Write a C program to simulate producer and consumer problem using muexes, shared memory, and threads.
27. Write socket Programs in C for Echo/Ping/Talk Commands.
28. Create a Socket (TCP) between two computers and enable file transfer between them.
29. Write a Program to implement Remote Command Execution.
30. Write a code simulating ARP/RARP.