

## EG 301 Operating Systems

### List of Lab Exercises – Only File and Process Management

---

1. Create the following types of a files using (i) shell command (ii) system call
  - a. soft link (symlink system call)
  - b. hard link (link system call)
  - c. FIFO (mkfifo Library Function or mknod system call)
2. Write a simple program to execute in an infinite loop at the background. Go to /proc directory and identify all the process related information in the corresponding proc directory.
3. Write a program to create a file and print the file descriptor value. Use creat ( ) system call
4. Write a program to open an existing file with read write mode. Try O\_EXCL flag also.
5. Write a program to create five new files with infinite loop. Execute the program in the background and check the file descriptor table at /proc/pid/fd.
6. Write a program to take input from STDIN and display on STDOUT. Use only read/write system calls
7. Write a program to copy file1 into file2 (\$cp file1 file2).
8. Write a program to open a file in read only mode, read line by line and display each line as it is read. Close the file when end of file is reached.
9. Write a program to print the following information about a given file.
  - a. inode
  - b. number of hard links
  - c. uid
  - d. gid
  - e. size
  - f. block size
  - g. number of blocks
  - h. time of last access
  - i. time of last modification
  - j. time of last change
10. Write a program to open a file with read write mode, write 10 bytes, move the file pointer by 10 bytes (use lseek) and write again 10 bytes.
  - a. check the return value of lseek
  - b. open the file with od and check the empty spaces in between the data.
11. Write a program to open a file, duplicate the file descriptor and append the file with both the descriptors and check whether the file is updated properly or not.
  - a. use dup
  - b. use dup2
  - c. use fcntl

12. Write a program to find out the opening mode of a file. Use `fcntl`.
13. Write a program to wait for a STDIN for 10 seconds using `select`. Write a proper print statement to verify whether the data is available within 10 seconds or not (check in `$man 2 select`).
14. Write a program to find the type of a file.
  - a. Input should be taken from command line.
  - b. program should be able to identify any type of a file.
15. Write a program to display the environmental variable of the user (use `environ`).
16. Write a program to perform mandatory locking.
  - a. Implement write lock
  - b. Implement read lock
17. Write a program to simulate online ticket reservation. Implement write lock  
Write a program to open a file, store a ticket number and exit. Write a separate program, to open the file, implement write lock, read the ticket number, increment the number and print the new ticket number then close the file.

18. Write a program to perform Record locking.
  - a. Implement write lock
  - b. Implement read lock

Create three records in a file. Whenever you access a particular record, first lock it then modify/access to avoid race condition.

### **Process Management**

19. Write a program to find out time taken to execute `getpid` system call. Use time stamp counter.
20. Find out the priority of your running program. Modify the priority with `nice` command.
21. Write a program, call `fork` and print the parent and child process id.
22. Write a program, open a file, call `fork`, and then write to the file by both the child as well as the parent processes. Check output of the file.
23. Write a program to create a Zombie state of the running program.
24. Write a program to create an orphan process.
25. Write a program to create three child processes. The parent should wait for a particular child (use `waitpid` system call).
26. Write a program to execute an executable program.
  - a. use some executable program
  - b. pass some input to an executable program. (for example execute an executable of `./a.out` name)
27. Write a program to execute `ls -l` by the following system calls
  - a. `execl`
  - b. `execlp`
  - c. `execle`

- d. `execv`
- e. `execvp`

28. Write a program to get maximum and minimum real time priority.

29. Write a program to get scheduling policy and modify the scheduling policy (`SCHED_FIFO`, `SCHED_RR`).

30. Write a program to run a script at a specific time using a Daemon process.