System Software-II

List of Lab Exercises

1. Write a separate program (for each time domain) to set a interval timer in 10sec and 10micro second
   a. ITIMER_REAL
   b. ITIMER_VIRTUAL
   c. ITIMER_PROF

2. Write a program to print the system resource limits. Use getrlimit system call.

3. Write a program to set (any one) system resource limit. Use setrlimit system call.

4. Write a program to measure how much time is taken to execute 100 getppid ( ) system call. Use time stamp counter.

5. Write a program to print the system limitation of
   a. maximum length of the arguments to the exec family of functions.
   b. maximum number of simultaneous process per user id.
   c. number of clock ticks (jiffy) per second.
   d. maximum number of open files
   e. size of a page
   f. total number of pages in the physical memory
   g. number of currently available pages in the physical memory.

6. Write a simple program to create three threads.

7. Write a simple program to print the created thread ids.

8. Write a separate program using signal system call to catch the following signals.
   a. SIGSEGV
   b. SIGINT
   c. SIGFPE
   d. SIGALRM (use alarm system call)
   e. SIGALRM (use setitimer system call)
   f. SIGVTALRM (use setitimer system call)
   g. SIGPROF (use setitimer system call)

9. Write a program to ignore a SIGINT signal then reset the default action of the SIGINT signal - Use signal system call.

10. Write a separate program using sigaction system call to catch the following signals.
   a. SIGSEGV
   b. SIGINT

c. SIGFPE

11. Write a program to ignore a SIGINT signal then reset the default action of the SIGINT signal - use sigaction system call.

12. Write a program to create an orphan process. Use kill system call to send SIGKILL signal to the parent process from the child process.

13. Write two programs: first program is waiting to catch SIGSTOP signal, the second program will send the signal (using kill system call). Find out whether the first program is able to catch the signal or not.

14. Write a simple program to create a pipe, write to the pipe, read from pipe and display on the monitor.

15. Write a simple program to send some data from parent to the child process.

16. Write a program to send and receive data from parent to child vice versa. Use two way communication.

17. Write a program to execute ls -l | wc.
  a. use dup
  b. use dup2
  c. use fcntl

18. Write a program to find out total number of directories on the pwd.
  execute ls -l | grep ^d | wc ? Use only dup2.

19. Create a FIFO file by
  a. mknod command
  b. mkfifo command
  c. use strace command to find out, which command (mknod or mkfifo) is better.
  c. mknod system call
  d. mkfifo library function

20. Write two programs so that both can communicate by FIFO -Use one way communication.

21. Write two programs so that both can communicate by FIFO -Use two way communications.

22. Write a program to wait for data to be written into FIFO within 10 seconds, use select system call with FIFO.

23. Write a program to print the maximum number of files can be opened within a process and size of a pipe (circular buffer).

24. Write a program to create a message queue and print the key and message queue id.

25. Write a program to print a message queue's (use msqid_ds and ipc_perm structures)
  a. access permission
  b. uid, gid
  c. time of last message sent and received
  d. time of last change in the message queue
  d. size of the queue
  f. number of messages in the queue
  g. maximum number of bytes allowed
  h. pid of the msgsnd and msgrcv

26. Write a program to send messages to the message queue. Check $ipcs -q

27. Write a program to receive messages from the message queue.
  a. with 0 as a flag
  b. with IPC_NOWAIT as a flag

28. Write a program to change the exiting message queue permission. (use msqid_ds structure)

29. Write a program to remove the message queue.

30. Write a program to create a shared memory.
  a. write some data to the shared memory
  b. attach with O_RDONLY and check whether you are able to overwrite.
  c. detach the shared memory
  d. remove the shared memory

31. Write a program to create a semaphore and initialize value to the semaphore.
  a. create a binary semaphore
  b. create a counting semaphore

32. Write a program to implement semaphore to protect any critical section.
  a. rewrite the ticket number creation program using semaphore
  b. protect shared memory from concurrent write access
  c. protect multiple pseudo resources (may be two) using counting semaphore
  d. remove the created semaphore

33. Write a program to communicate between two machines using socket.

34. Write a program to create a concurrent server.
   a. use fork
   b. use pthread_create