

Assignment 12

Operating System Lab (CS342)

Department of CSE, IIT Patna

Date:- 9-Apr-2019

Time:- 3 hours

Instructions:

1. All the assignments should be completed and uploaded by **9-Apr-2019, 5pm.**
2. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
3. **Proper indentation and appropriate comments are mandatory.**
4. You should zip all the required files and name the zip file as ***roll_no.zip***, eg. **1601cs11.zip**.
5. Upload your assignment (**the zip file**) in the following link:
<https://www.dropbox.com/request/erzqhO6Tesfylc5HxGZh>

Q1) Reader-Writer problem

In this assignment, you are asked to implement the Reader-Writer problem with Reader's priority. In this system, you have two readers who read from a shared buffer and one writer who updates the buffer. The readers are to be implemented with the help of child processes and the writer as the parent process. Implement the shared buffer with the help of a shared file called buffer.txt. The buffer contains a variable VAR which is initialized to zero. After each access, the writer reads the content of the buffer, prints it (along with its own PID), and increments the content by one. On the other hand, each reader periodically reads the content of the buffer, and prints the current value along with its own identity (PID). Use semaphore(s) in order to ensure proper synchronization between the Reader and Writer processes. Print appropriate messages whenever a process (reader or write) gains control of the buffer. Insert appropriate delays (sleep() or usleep()) so that the messages become visible. The program terminates when the writer writes MAX to VAR (fix a value in MAX at the beginning). After this happens, the parent kills the reader

processes, and itself terminates. Notice that in the Reader-Priority mode, both the Readers can access the shared content together. The Reader who first makes a read operation should lock the shared content for reading. During the time the shared content is locked for reading, the other Reader can read it. The last Reader done with reading releases the lock. The Writer can write only when no Reader is reading the shared content. Finally, before the Writer finishes writing, no Reader can read.

Sample run (Parent ID: 123, Child IDs: 124, 125)

File created. VAR is 0

First Reader enters

Reader (124) reads the value 0

Last Reader leaves

Writer enters

Writer (123) writes the value 1

Writer leaves

First reader enters

Reader (125) reads the value 1

Reader (124) reads the value 1

Last Reader leaves