Indian Institute of Technology Tirupati

Renigunta Road, Tirupati 517506, AP

September 5, 2019

CS3500 Operating Systems Lab4 - Process Synchronization

1. (5 marks) Folder name is rollnumber_lab4. In rollnumber_lab4a.c file, implement two concurrent processes in Xinu that prints their successive names, say, Process-A and Process-B. Check if they are printed in expected manner, or being disturbed by context switch. Attach the output screenshot in your submission.

2. (5 marks) In rollnumber_lab4b.c file, implement the discussed producer and consumer global sharing problem and verify the occurence of race condition. Attach the appropriate output screenshot in your submission, and discuss the race condition occured in a simple rollnumber_lab4b_readme.txt file. By any chance, if the context switch is not happening to create a race condition, try producing a sequence of more data (at producer), and consuming a sequence of more data (at consumer).

3. (10 marks) Problem-1 extension to rollnumber_lab4a-ext.c, Read and understand the the Semaphores (a brief would be given in this lab) implemented the given Xinu system calls namely, wait.c and signal.c, and the associated header file semaphore.h. Use this semaphore, to toggle the printing of each processes' names.

4. (10 marks) Problem-2 extension to rollnumber_lab4b-ext.c, Read and understand the Semaphores (a brief would be given in this lab) implemented in the given Xinu system calls namely, wait.c and signal.c, and the associated header file semaphore.h. Use this semaphore, to mutally exclude the critical section.

5. (10 marks) In rollnumber_4c.c, implement a scenario with semaphores exhibiting the existence of deadlock.

Note: In the main folder of rollnumber_lab4, attach all the .c, .txt files, and screenshot images. Proper naming and explaining your implementation in readme file carries additional 10 marks.