Indian Statistical Institute

Operating Systems

Lab 2: Synchronisation

> Make sure man pages for system calls are installed. If you are using a Fedora based distribution, you will need to install `man-pages`; for an Ubuntu based distribution, please install `manpages-dev`.

Man pages:

`semget(2)`, `semop(2)`, `semctl(2)`

`sem_overview(7)`, `sem_open(3)`, `sem_init(3)`, `sem_wait(3)`, `sem_post(3)`, `sem_destroy(3)`, `sem_close(3)`, `sem_unlink(3)`

`shmget(2)`, `shmat(2)`, `shmdt(2)`

`shm_overview(7)`, `shm_open(3)`, `ftruncate(2)`, `mmap(2)`, `munmap(2)`, `shm_unlink(3)`

1. Go through the example program at `https://www.dropbox.com/s/pteg8junueiq9ei/posix-semaphores.c`.

2. Write a program to simulate the Dining Philosophers Problem using SysV semaphores. Your program should take the number of philosophers as the only (compulsory) command line argument.

3. Write a program to simulate the Producer-Consumer Problem using POSIX semaphores and shared memory.

4. Write a program that provides a starvation-free simulation of the Readers-Writers Problem using POSIX semaphores and shared memory.

5. [EXTRA] Write a program that provides a starvation-free simulation ofa traffic crossing. Vehicles move through a crossing in one of 4 directions: North-South, South-North, East-West, West-East. At a time, vehicles moving in parallel directions (e.g. N-S and S-N) may pass through the crossing, but vehicles moving in perpendicular directions are not permitted to simultaneously pass through the crossing.

Your programs should be designed so that it is possible to easily turn off the use of synchronisation and demonstrate race conditions (e.g., violation of the mutual exclusion property).