# Tutorial #5 Shared Memory & Semaphores

#### Anirban Lahiri & Prashant Agrawal Department of Computer Science & Engineering IIT Kharagpur

## **Outline**

- Accessing Shared Memory
- Race Condition
- Related Functions for Shared Memory
- Critical Section Problem
- Structure of a Process with Critical Section
- Semaphores
- Using Semaphores
- Related Functions for Semaphores

## **Accessing Shared Memory**





Computer Architecture & Operating System Course Anirban Lahiri & Prashant Agrawal under guidance of : Profs. D. Sarkar, R. Mall, D. R. Chowdhury, P. S. Dey " It is a situation where several processes access and manipulate the same data concurrently, and the outcome of the execution depends on the particular order in which the access takes place. "

Ref: Operating System Concepts, Silberschatz & Galvin (5th edition); Pg. 157

## **Race Condition**

#### E.g.



#### **Related Functions for Shared Memory**

shmget()

- To create a shared memory segment
- shmat()
  - To attach a shared memory segment to a process
- shmdt()
  - To detach a shared memory segment from a process
- shmctl()
  - To receive information on a shared memory segment, set the owner, group, and permissions of a shared memory segment, or destroy a segment

## **Critical Section Problem**

- In a multi-process system, each process has a segment of code, called a critical section, in which the process may be changing some common data.
- This common data may be one or more variables, tables, files, etc



## **Critical Section Problem**

- The execution of critical sections by the processes should be mutually exclusive in time
  - i.e. when one process is executing in its critical section then no other process is allowed to execute in its critical section
- □ This is referred to as the critical section problem

#### Structure of a Process with Critical Section



#### Figure: Structure of a process P<sub>i</sub> with critical section

#### Semaphores

- A semaphore is a protected variable that is used to synchronize access to shared data in a multi-process environment
- A semaphore can be accessed only by the following atomic operations



## **Using Semaphores**

A semaphore may be used by a number of processes to synchronize access to a shared data



repeat



critical section

SIGNAL(S)

remainder section

until false;

Computer Architecture & Operating System Course Anirban Lahiri & Prashant Agrawal under guidance of : Profs. D. Sarkar, R. Mall, D. R. Chowdhury, P. S. Dev

25 Oct. 2005 11

## **Related Functions for Semaphores**

semget()

- To create a semaphore
- semop()
  - To perform wait and signal operations on a semaphore
- semctl()
  - To perform control operations on the semaphore , e.g. destroying a semaphore

#### References

- Operating System Concepts by Silberschatz and Galvin
- http://www.ecst.csuchico.edu/~beej/guide/ipc/semaphores.html
- http://www.csm.astate.edu/~rossa/semab.html
- http://www.cim.mcgill.ca/~franco/OpSys-304-427/lecture-notes/node31.html
- Linux man pages

Computer Architecture & Operating System Course Anirban Lahiri & Prashant Agrawal under guidance of : Profs. D. Sarkar, R. Mall, D. R. Chowdhury, P. S. Dey

#### "The most important thing in communication is to hear what isn't being said" - Peter Ducker