Concurrency

COSC-361

Stephen Marz

TICKLE COLLEGE OF ENGINEERING

MIN H. KAO DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

1

Definitions

- **Parallel** - two or more computational units capable of executing simultaneously.
- **Concurrency** - ability of multiple parts of a program to be executed out of order and/or in parallel.
- **Critical section** - instructions where preemption would cause failure.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

2

Multi-user / Multi-process

- Processes compete for limited resources.
- A process can be interrupted (preempted) at any moment.
- Sometimes, a request must be finished before another section can start.
  - This type of section is called a **critical section**

3

Critical Sections

1. Guarantee mutual exclusion
2. Prevent lockout
3. Prevent starvation
4. Prevent deadlock

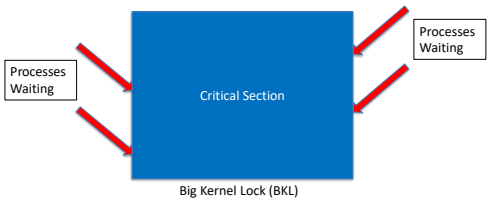
4

Naive Locking

- **Big Kernel Lock (BKL)**
  - Prevent any preemption at all
  - Very slow, things that have nothing to do with another can't run.
- Linux had BKL for quite a while.

5

Big Kernel Lock



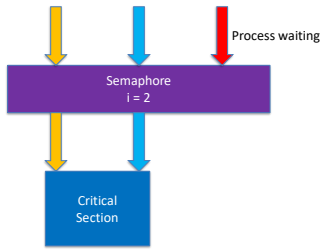
6

Locking: Semaphore

- A positive integer.
  - When the integer reaches 0, it blocks until the integer is increased again.
- Functions
  - ***Down***: lock
    - Decreases the integer.
    - If integer is already 0, it waits.
  - ***Up***: unlock
    - Increases the integer.
    - Notifies that resources are again available.

7

Semaphore



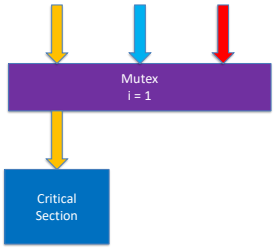
8

Locking: Mutex

- Mutual Exclusion (Mutex)
- A binary semaphore
  - Integer is either 0 (locked) or 1 (unlocked)
- Functions
  - Lock: sets the mutex to 0
    - If lock is already set, it waits.
  - Unlock: sets the mutex to 1.

9

Mutex



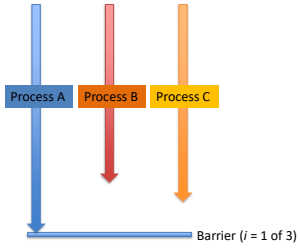
10

Locking: Barrier

- Barriers are a finish line for processes
- Processes wait at the end of the barrier until ALL processes make it to the barrier.

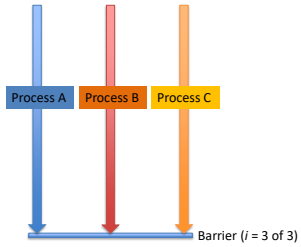
11

Barriers: Waiting State



12

Barriers: Unblocked State



13

Synchronization Problem Examples

14

Reader's/Writer's

- Two types of processes
  - ***Readers*** - want to GET data
  - ***Writers*** - want to SET data
- Critical Section
  - May have multiple readers
  - May have ONLY ONE writer
    - No readers allowed while a writer is in critical section.

15

R/W Rules

1. A reader may enter critical section if other readers are in it.
   a) when last reader exits, writers are allowed in.

2. All readers that arrive when a writer is in the critical section enter before the next writer.

16

Synchronization without locks

17

Read-Copy-Update (RCU)

- Primitive without need for locks.
- Locks have an acquire (lock) and release (unlock) cost.
- Typically implemented as a singly linked list

18


Concurrency
COSC-361
Stephen Marz
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

19