Some Classic Concurrency Problems

An Introduction

Joseph Kehoe[1]

[1] Department of Computing and Networking
Institute of Technology Carlow

CDD101, 2017

Mutual Exclusion

- How do we stop more than one thread accessing a variable at the same time?
  1. int count=0
  2. Start Critical Section
  3. count++;
  4. End Critical Section
- Show how this can be done using a Semaphore

Multiplex

- Generalise mutual exclusion solution to allow a maximum of N threads access a critical section
    - Critical section is a block of code that we need to restrict concurrent access to
    - Only N threads can be in critical section at any one time (N>0)
- We must show that Deadlock cannot occur under any circumstances
- We should also show fairness

Barrier

- Rendezvous only works for two threads
- A barrier works for N threads
  - When first N-1 threads arrive at barrier they are blocked
  - When the Nth thread arrives they all continue

Hints on next slide!

Barrier Hint

1. We will need to keep track of the number of threads who have reached the barrier
2. Use a counter
3. counter shared by all threads