Some Classic Concurrency Problems

An Introduction

Joseph Kehoe[1]

[1] Department of Computing and Networking
SETU, Carlow Campus

Concurrency, 2023

Mutual Exclusion

Mutual Exclusion Prevent threads simultaneous access of a variable

1. Use a Mutex;
2. Use a Semaphore initialised to 1;
3. Use a buffered channel of size 1.

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

Use a semaphore (initialised to N) or a buffered Channel (of size N)

Rendezvous

- Two threads much reach a certain point before either can continue
  - Each must signal the other that it has arrived and
  - Wait until it gets a signal that its partner has also arrived
- We must show that Deadlock cannot occur under any circumstances
- We should also show fairness

Use Mutexes or Semaphores or Channels

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

Discussion

do it now!

1. Break into groups. Each group must come up with a solution!