Semaphores

An Overview

Joseph Kehoe[1]

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

CDD101, 2017

What is a Semaphore?

Invented by Edsgar Dijkstra

- A Semaphore is an integer with the following properties
  1. When you create a semaphore you can initialise it to any integer value but after that you can only perform two operations on it.
  2. You can increment by one or decrament it by one.
  3. You cannot read the current value of a semaphore
  4. When a thread decrements a semaphore, if the result is negative, the thread blocks itself and cannot continue until the semaphore value is no longer negative
  5. When a thread increments a semaphore, if there are other threads waiting on that semaphore then one of them becomes unblocked

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Consequences of Definition

- When you signal a semaphore you do not necessarily know whether another thread is waiting
  - so the number of unblocked threads may be zero or one
- In general there is no way of knowing whether a thread will block on a decrement operation
- After an increment operation both the incrementing thread and one waiting thread can run concurrently
  - but there is no way of knowing which (if either) will continue immediately

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Consequences

The value of a semaphore indicates:

- Positive integer represents the number of threads that can decrement without blocking
- Negative integer represents the number of waiting (blocked) threads
- Zero means no threads are waiting

But you are not allowed to ask a semaphore what its value is!

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Consequences

Only two operations are allowed on a semaphore:

1. Decrement the counter
   - Called dec(), wait() or P()
2. Increment the counter
   - Called inc(), signal() or V()

But you are not allowed to ask a semaphore what its value is!

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Why use Semaphores

- They impose constraints that help programmers avoid errors
- Code using semaphores tends to be clean and organised
- Semaphores have efficient implementations
- Mainly we use them to force you to think clearly about the issues of concurrency

Lessons learned here will be applicable to any concurrency programming model you use in the future

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Creating Semaphores in C++

Look at the following files online:

- Semaphore.h
- Semaphore.cpp

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Signalling with Semaphores

- A single semaphore can be used to send a signal from one thread to another
- To indicate something has happened
- Use a semaphore initialised with value 0
  - Thread waiting for signal calls wait
  - Thread sending signal calls signal

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Simple Rendezvous

Generalised Signal Pattern

- Thread 1 has to wait for thread 2
- Thread 2 has to wait for thread 1
- Both have to arrive at a certain point before either proceeds
- Thread A
  - A1;
  - A2;
- Thread B
  - B1;
  - B2;
- A1 must finish before B2 starts
- B1 must finish before A2 starts

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Exercise

- Implement both solutions in C++
- Create Makefile
- Document with Doxygen
- Put solution up on github

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster