

NWEN303 Concurrent Programming (2016)

Assignment 2 : Shared Memory Concurrency

(Due midnight, Sunday 14 August)

This assignment is intended to help you to better understand some of the topics we have covered in lectures. You may need to do some additional reading in order to answer these questions — it is essential that you acknowledge any sources you referred to, and that whatever you submit is written in your own words.

You should submit your work using the online submission system, giving your answers as a pdf document (scan it if necessary).

Note that the marks shown are provided as a guideline, and may be changed.

1. Readers and Writers [30 marks]

Modify the solution to the readers and writers problem given in lectures to implement (separately) each of the following rules:

- (a) If there are reading processes, a new reader may commence reading even if there are writers waiting.
- (b) If there are waiting writers, they receive precedence over all waiting readers.
- (c) If there are waiting writers, no more than two writers will write before a reader is allowed to read.

In each case, you should explain how your solutions enforces the relevant rule.

2. Critical Section Problem [20 marks]

Consider a variation of the Critical Section Problem where there are N processes, and at most M processes can enter their critical region at the same time, for some M where $1 < M < N$. (We will see later that this provides one way of avoiding deadlock in the Dining Philosophers Problem.)

- (a) Develop a solution to this problem using monitors.
- (b) Develop a solution to this problem using semaphores.

In each case, you should explain how your solution works, and how it satisfies the relevant correctness properties.

3. Producers and Consumers revisited [30 marks]

Consider a variation of the Producers and Consumers problem, where there are M producers and N consumers, and multiple processes can be writing to or reading from the buffer as long as they are reading or writing on different locations in the buffer. As before, a producer grabs a location at the back of the queue to write to and a consumer grabs a location at the front of the queue to read from, but now they don't have to wait for previous processes to finish.

Develop a solution to this problem, using a monitor. Explain how your solution works, and how it satisfies the relevant correctness properties.

4. **Double Buffering** [20 marks]

Consider a variation of the Producers and Consumers problem, where a buffer is a single variable (rather than a queue), and there are two buffers, so that one process can be writing into one buffer while the other process is reading from the other, then they swap over, etc.

This used to be a common way of dealing with I/O. One thread would read a block of data from a device such as a disk or magnetic tape, while another thread was processing the previously read block.

Develop a solution to this problem, using any of the synchronisation mechanisms we have discussed. Explain your choice of synchronisation mechanism, and how your solution satisfies the relevant correctness properties.

Total possible: 100 marks