

Concurrent and Distributed Systems: Supervision 1

Lectures covered by the supervision: <https://www.cl.cam.ac.uk/teaching/2627/ConcDisSys/>

- Lecture 1: Introduction to concurrency, threads, and mutual exclusion
- Lecture 2: Hardware, OS and automaton Views
- Lecture 3: Mutual exclusion, semaphores, and producer-consumer relationships
- Lecture 4: CCR, monitors, and concurrency in practice

Previous exams.

Supervision questions:

1. Explain differences between concurrency and parallelism? What is motivating engineers to use parallelism?
2. Discuss how concurrency and parallelism influence software engineers.
 - a. Do software engineers explicitly decide what will be a process and what will be a thread?
 - b. Does OS differentiate between threads and processes and why?
 - c. Does CPU differentiate between threads and processes and why?
3. Create two examples to demonstrate influence of parallelism on software quality (e.g., performance, testability). In the first example, argue why concurrency is beneficial. In the second example, argue why concurrency is a bad choice.
4. Sketch an example where 4 threads try to find prime numbers between 0 and 1 000 000. A main thread creates these threads and should print, at the end, the number of primes. Sketch how stack and heap of these threads look like.
5. What are Heisenbugs? Sketch an example with demonstrating the possibility for heisenbugs to appear.
6. Compare single threaded and multithreaded software.
 - a. Create an example with pseudo code that you parallelise to take advantage of multiple cores.
 - b. What determines intermediate states and output of a sequential program?
 - c. What determines intermediate states and output of a multithreaded program?
7. What is mutual exclusion aiming to solve? What causes that problem? Can we have multithreaded software without a need to introduce mutual exclusion?
8. Explain atomicity.
 - a. Definition
 - b. Importance
 - c. Application
9. Compare these synchronisation mechanisms (implementation complexity, performance, useability, influence on progress of threads, intended use) – discuss what kind of synchronisation intentions these mechanisms mimic:
 - a. Compare-and-swap (CAS)
 - b. Load Linked, Store Conditional (LL/SC)
 - c. Producer-consumer
 - d. Multiple-Readers Single-Writer
 - e. Semaphore
 - f. Condition variable

- g. Conditional Critical Regions
 - h. Monitors
10. Explain Dining Philosophers problem (use FSM for 2 philosophers).
 11. Sketch an example (pseudocode) for the following concurrency bugs:
 - a. Data-related bugs: Data race, multivariable data race, atomicity violation
 - b. Liveness: live lock, deadlock.
 12. Apply the Lamport Bakery Algorithm in your examples above (with data races and atomicity violation).
 13. Explain Multiple-Reader-Single-Writer synchronisation mechanism. Use pseudo code.
 14. Compare Conditional Critical Regions and Monitors - advantages/disadvantages. Illustrate monitor with pseudo code.
 15. Compare mentioned synchronisation mechanisms in terms of:
 - a. What kind of communication patterns between threads they aim to handle?
 - b. Performance
 - c. Usability
 - d. Prone-to-bugs
 16. Explain signal-and-wait and signal-and-continue synchronisation mechanisms.
 17. Summarize in 1-3 sentences Lesson 1.
 18. Summarize in 1-3 sentences Lesson 2.
 19. Summarize in 1-3 sentences Lesson 3.
 20. Summarize in 1-3 sentences Lesson 4.

BONUS: Assume a long array of integers where elements of the array have values between 1 and 4. It is necessary to calculate the square of each element, and the sum of such squared elements. Create 2 versions of a program: i) sequential and ii) concurrent with threads. Use any synchronisation mechanism that you consider is the best for the performance. Create programs with pthreads (C++) and Java. Comment on the performance (e.g., to measure performance use `time ./programName`) on Linux. Commit the solution to Github - provide a link.

Save your answers into MS Teams or email them to me. Please use the following naming pattern:

CDS_Supervision_1_Answers_<last name>_<first name>_Michaelmas_2026

Send your answers as a pdf, doc, image, or any other format of a document for which there exists an easily available software to open.

Jasmin JAHIC
jj542@cam.ac.uk
<https://www.cl.cam.ac.uk/~jj542/>