8 Concurrent and Distributed Systems (avsm2)

(*a*) Programs with concurrency are vulnerable to classes of problems that are not exhibited in single-threaded programs.

(*i*) Explain the concepts of *deadlock* and *livelock* in a multithreaded program. [2 marks]

(*ii*) Explain the four conditions required for deadlock. A sentence explaining each condition is sufficient. [4 marks]

(*b*) Modern instruction set architectures provide instructions for performing atomic operations over memory locations.

(*i*) One class of instructions are generically referred to as Compare And Swap (CAS). Describe how the CAS instruction on the x86 architectures is used to perform an atomic operation. Briefly explain how the instruction uses the instruction operands when executed. [3 marks]

(*ii*) Databases often utilise a technique known as *write-ahead logging* to provide durability guarantees. Describe how a disk-based transaction log might be implemented, and what the atomic operation used in this technique is. [3 marks]

(*c*) The below snippet of C code uses `pthreads` for concurrent execution. It uses a mutex `M` and a condition variable `C` to ensure that the `run_critical_code` function only executes when the `condition` boolean is `true`.

```
L1:    pthread_mutex_lock(&M);
L2:    run_critical_code ();
L3:    if (!condition)
L4:      pthread_cond_wait(&C, &M);
L5:    if (!condition)
L6:      pthread_cond_broadcast(&C);
L7:    pthread_mutex_lock(&M);
```

Unfortunately, there are four bugs in the code that prevent it from working correctly. List each of the four bugs and describe how each bug affects programme execution. Write down a new version of the code snippet with the four bugs corrected. [8 marks]