CS0: Get-started Questions (DJ Greaves) (Rev B)

The questions on this preliminary sheet are mostly open-ended and their purpose is mainly for discussion in the first supervision. So do not worry about generating complete answers in advance.

* Star denotes optional/advanced exercise.

Q0 Parallel Programming

List or tabulate the essential similarities and/or differences between parallel programming and distributed systems.

Can you think of an application or algorithm where the shared memory is **not** just being used for some form of 'message passing'?   ## Q1 Non-deterministic Schedulling

Why might the output from a concurrent program vary on different runs? What is one advantage of allowing this? What is one disadvantage? [Do not consider programs that read the RTC (real-time clock), use random number generators or read a different input data in different runs!]

Q2 Operating System Fundamental Abstractions.

Early versions of the Windows Operating System (before circa 1998) lacked most features that would be expected to be found in an operating system: it was essentially a GUI-controlled command shell. List the minimal abstractions expected from a proper operating system. Windows did provide some basic, non-preemptive threads (using co-routines). What benefit did having threads bring and what problem arose from them being non-preemptive? What is the essential difference between a thread and a process?

Q3 Atomic Hardware Operations

This is a question to think about at the start of this CC/DS course and which you should probably be able to answer with confidence by start of Lent term!

Which of the following operations can be considered atomic on a modern digital computer: Store of a character? Store of a 32-bit word? Store of a 64-bit word? Atomic compare-and-swap? Write of a disk sector? A system call? A floating-point division? Sending a network packet? Signalling a semaphore? An inter-processor interrupt (IPI) aka inter-core interrupt (ICI) ?

Make sure you understand the difference between *cache consistency* and *sequential consistency* by the end of the Computer Design course. You may find this book helpful: *'Modern SoC Design on Arm'* [2021, DJ Greaves].

Q4 CBMC Example (*)

See if you can reproduce the CBMC model checker with the Beer Fridge Stocking problem or

whatever else was shown covered in the first Examples Class.