Engineering Tripos Part IIB, 4F14: Computer Systems, 2024-25

Module Leader

Prof Andrew Gee [1]

Lecturers

Prof Andrew Gee and Prof Per Ola Kristensson [2]

Timing and Structure

Lent Term. 12 lectures and 2 examples classes. 75% exam / 25% coursework. Lectures will be recorded.

Prerequisites

Part 1 Digital Circuits and Computing, including an understanding of C++ as taught in the Mars Lander project and the Part 1B Device Programming exercise. Students will be expected to write small programs in C++.

Aims

The aims of the course are to:

- Describe the computer hardware that underlies modern information processing systems.
- Explain how to write multithreaded software that runs on such hardware.

Objectives

As specific objectives, by the end of the course students should be able to:

- Appreciate the basic components needed to construct a computer and the different ways to interconnect these components, including the various ways of exploiting parallelism.
- Compare the instruction sets, implementation issues and performance of CISC and RISC architectures.
- Design efficient hardware for computer arithmetic.
- Understand the operation of pipelined datapaths.
- Describe memory organisation, addressing schemes and the use of caches; and their effects on performance.
- Compare the various ways of handling input and output in a computer system.
- Understand the concept of a memory model.
- Understand basic concurrency concepts.
- Design and implement thread-safe algorithms in C++.

Content

Computer Systems (8 lectures and 2 examples classes, Prof Andrew Gee)

- Computer architecture, historical perspectives.
- Instruction set architectures, RISC vs CISC.
- ALU design, datapaths and control, pipelining.
- Memory hierarchy, caches, virtual memory.

- Input/output, bus organization, polling and interrupt-driven I/O, DMA.
- Parallel processing, SIMD and MIMD architectures.

Assessment: examination (75%), candidates to attempt two questions from a choice of three

Parallel Programming (4 lectures, Prof Per Ola Kristensson)

- C++ memory model.
- Race conditions, mutual exclusion, synchronization, starvation.
- Thread-safe data structures.
- C++ threading library.

Assessment: coursework (25%)

Coursework

Multithreaded programming using the C++ memory model and threading libraries. The programming exercise is an opportunity to experience how theoretical concepts from the lectures translate into actual working code using a state-of-the-art industry standard threading library. Time required: 4-8 hours programming plus 15 minutes demonstrating and discussing your code with an assessor. Please note that coursework assessment is not anonymous.

Coursework	Format	Due date & marks
Multithreaded programming Learning objectives: • To gain practical experience with the C++ threading library. • To design and implement thread-safe data structures. • To practice concurrency control so as to avoid race conditions and starvation.	Individual Demonstrating your software Not anonymously marked	Software to b Lent Term Assessment Term [15/60]

Booklists

Please refer to the Booklist for Part IIB Courses for references to this module, this can be found on the associated Moodle course.

Examination Guidelines

Please refer to Form & conduct of the examinations [3].

Last modified: 31/05/2024 10:08

Source URL (modified on 31-05-24): https://teaching25-26.eng.cam.ac.uk/content/engineering-tripos-part-iib-4f14-computer-systems-2024-25

Links
[1] mailto:ahg13@cam.ac.uk

[2] mailto:ahg13@cam.ac.uk, pok21@cam.ac.uk
[3] https://teaching25-26.eng.cam.ac.uk/content/form-conduct-examinations