

Introduction to Computer Systems (INF2C-CS)

Proposal for Course Restructuring

Course organizer: Boris Grot

1 Introduction

This is a proposal to restructure INF2C-CS (INFR08018) from a 10-credit course to a 20-credit course. The current structure and topics are working well in covering the core introductory topics in computer systems. Despite being quite intensive and having demanding coursework assignments, the course counts for only 10 credits. The reasons for this are mostly historical, and do not fairly reflect the actual number of hours students spend on the course. The following proposal aims to correct the existing imbalance by increasing the number of credits to 20. The proposal also seeks to increase course engagement for students through additional lab exercises and tutorials, as well expand a few of the topics covered in the course while maintaining the current syllabus content.

2 Coursework

Present: the course has 2 summatively-assessed courseworks, which exceeds what is allowed for 10-credit courses, but meets the 20-credit course guideline.

Proposed for the 20-credit course: will continue having 2 summatively-assessed courseworks. Will introduce one piece of formative assessment with feedback; most likely, this will be one of the lab exercises.

3 Course Workload

Current: the course has 2 lectures a week, 4 tutorials, and 1 lab. Together with the two rather intensive courseworks and the readings, this translates into approximately 8-10 hours per week.

Proposed for the 20-credit course: will maintain 2 lectures per week. Will also make the following additions:

- 1 additional lecture (for the course, not per week) to accommodate small additions to a topic already covered in the course.
- 1 additional tutorial to expand the set of lecture topics covered by tutorials.
- 2 additional labs to provide students with hands-on experience in applying classroom knowledge and to better prepare them for the courseworks.

Item	Hours for each	Total
Lectures x22 (11 weeks, 2 per week)	1	22
Preparation for lectures	1.5	33
Tutorials (inc. preparation) x5	3	15
Labs x3	5	15
Courseworks x2	20	40
Revision + Exam	34	34
Programme level activities	20	20
Total		179

4 Learning Outcomes

1. Describe the trade-offs in different binary representation systems.
2. Explain the principles of: instruction set architecture, digital logic design, cache hierarchy, virtual memory, I/O devices, exceptions and processor management.
2. Demonstrate an understanding of how a high-level programming language (C) maps to the assembly code by converting a simple C program to MIPS assembly.
3. Sketch the design of a simple single- and multi-cycle processor and explain how it operates by combining the knowledge of the logic design basics with that of the MIPS instruction set architecture.

3 Syllabus

No changes to the syllabus are being proposed. The set of topics covered in class is as follows:

- Introduction and background
- Data representation
- Instruction set and assembly language
- C programming language
- Logic design
- Single- and multi-cycle processor
- Memory hierarchy and caches
- Virtual memory
- Exceptions and processor management
- I/O