
Ano Letivo 2019-20

Unidade Curricular ARQUITETURA DE COMPUTADORES

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14781042

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Português

Modalidade de ensino Presencial

Docente Responsável Peter Stallinga

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Peter Stallinga	T; TP	T1; TP1; TP2	30T; 60TP
João Miguel Gago Pontes de Brito Lima	TP	TP3	30TP

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
1º	S2	30T; 30TP	168	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Não se aplica

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

Após a conclusão da disciplina, os alunos deverão compreender os princípios básicos da arquitetura, dos sistemas mais simples até os sistemas mais avançados, para além do comportamento e desempenho desses sistemas em relação ao software e aplicações reais (incluindo alguns aspectos da otimização de código e do desenvolvimento de código em assembly e C).

Conteúdos programáticos

História: do IAS (1946) até o CRAY T3E (1996) etc., desempenho, benchmarks (Linpack e Spec), a lista TOP500 do Dongarra

Arquiteturas simples: unidades, buses, registos, periféricos, a decodificação de instruções, tipos de controlo hardware, e a programação aos níveis micro e nano
Gestão de memória: virtual, caching, interleaving, a hierarquia de memória

Gestão de periféricos: controladores DMA e interrupções, comunicação série e paralela

Periféricos e memória: RAM estática e dinâmica, fitas, discos, um terminal, teclado e monitor

Arquiteturas avançadas: RISC, pipelining, vector, superescalar, R10000 da MIPS, PVP, SMP, MPP, NUMA, GPGPU

Introdução aos processadores da Intel (8085 e família x86), AMD etc.

Programação de um processador da MIPS, instruções e linguagem assembly. Programação de dispositivos simples de entrada e saída e um device driver.

Demonstração da coerência dos conteúdos programáticos com os objetivos de aprendizagem da unidade curricular

O objetivo desta uc é o desenvolvimento de uma visão geral e compreensão da arquitetura de computadores, dos sistemas simples até os sistemas mais avançados (incluindo a programação ao nível mais baixo: assembly). Os conteúdos programáticos cobrem toda a matéria necessária, com uma organização lógica e sequencial

Também a noção de desempenho será abordada. A parte pratica será uma introdução à programação ao nível de assembler (RISC) e da sua interface com C, assim como a programação das entradas (teclado), saídas (VGA) e de um driver adicional. A sebenta da parte teórica cobre uma larga gama de tópicos, todos relacionados com arquitetura em geral, enquanto a sebenta de TP foca em programação assembler ao nível introdutório, por questões de simplicidade reduzido à arquitetura RISC e interface de código em assembler com código em C. Dado que poucos estudantes alguma vez terão a necessidade de programar em assembler numa arquitetura CISC, este tópico não se encontra coberto na sebenta

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas são introduzidos muitos princípios básicos através de sistemas reais. Existe muita informação disponível em páginas web, inclusive os últimos desenvolvimentos. Além de participar nas aulas T (discussão sobre a matéria exposta, dúvidas e ideias), o aluno deve explorar a informação online para aprofundar conhecimentos.

Nas aulas TP são introduzidos os conceitos básicos da programação ao nível mais baixo: assembler. Utilizando um emulador do conjunto de instruções dos processadores RISC da empresa MIPS, como o MARS, o aluno aprende a escrever e testar programas simples até programas com recursividade (inclui interface entre linguagem C e código assembly). A programação de dispositivos de entrada e saída, e de um device driver em C é abordada.

Exame sem consulta, a parte T (60%) sendo obrigatória. Avaliação contínua da parte TP (40%), com a possibilidade de melhorar a nota TP nas provas escritas. Aprovação do aluno: nota TP superior a 4,0 AND nota T+TP superior a 9,5

Demonstração da coerência das metodologias de ensino com os objetivos de aprendizagem da unidade curricular

Esta unidade curricular é leccionada no segundo semestre do primeiro ano, logo depois de Sistemas Digitais em que a tecnologia digital foi introduzida inclusive um microprocessador da primeira geração. Nas aulas teóricas, muitos acetatos que mostram muitas arquiteturas e sistemas são utilizados para explicar aspetos básicos e soluções diferentes, e também para discutir vantagens específicas e a integração destas em computadores e Pcs. Pensamos que, para a grande maioria dos alunos, a apresentação de uma visão geral é mais importante do que a análise pormenorizada de uma arquitetura. No que respeita à parte TP, é introduzida passo a passo uma arquitetura RISC (conjunto de instruções MIPS), permitindo aos alunos a programação ao nível assembler utilizando um emulador como MARS. Portanto, os alunos podem desenvolver uma visão alargada em geral para compreender novos desenvolvimentos tecnológicos, para além de uma visão

Bibliografia principal

Computer architecture and organization. J.P. Hayes, McGraw-Hill 1988 and new edition

The 8086/8088 Family: Design, Programming, and Interfacing. J. Uffenbeck, Prentice-Hall 1987

Computer architecture, a quantitative approach. J.L. Hennessy and D.A. Patterson, Morgan Kaufmann, 1990 and later

The Intel microprocessors 8086. 80486: Architecture programming and interfacing. 3rd Ed. B.B. Brey, Prentice-Hall/Macmillan, 1994

8086 ... 80486 Assembly language programming. B.B. Brey, Prentice-Hall/Macmillan, 1994

Microprocessors. Theory and applications. M. Rafiqzaman, Prentice-Hall, 1992

The 68000 microprocessor. Hardware and software principles and applications. J.L. Antonakos, Macmillan, 1993, 2nd Ed

Conhecendo a família 80486. B. Segal et al., Livros Erica Edit, 1992

Computer organization and architecture. Principles of structure and function. W. Stallings, Macmillan, 1993, 3rd Ed

Sistemas digitais. Antonio J.G. Padilla, McGraw-Hill, 1993

All INTEL books, like i486 Microprocessor

Academic Year 2019-20

Course unit COMPUTER ARCHITECTURE

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School FACULTY OF SCIENCES AND TECHNOLOGY

Main Scientific Area CIÊNCIA DE COMPUTADORES

Acronym

Language of instruction Portuguese

Teaching/Learning modality Face to face learning

Coordinating teacher Peter Stallinga

Teaching staff	Type	Classes	Hours (*)
Peter Stallinga	T; TP	T1; TP1; TP2	30T; 60TP
João Miguel Gago Pontes de Brito Lima	TP	TP3	30TP

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	30	0	0	0	0	0	0	168

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Not applicable

The students intended learning outcomes (knowledge, skills and competences)

After finishing this subject, students must understand the basic principles of architecture, from the simplest systems to the more advanced, the behaviour and performance of these in relation to software and real applications, including some aspects of code optimisation and developing code at assembly level.

Syllabus

History: from the IAS (1946) to the CRAY T3E (1996) etc., performance, benchmarks (Linpack and SPEC), Dongarra's TOP500 list

Simple architectures: units, busses, registers, peripherals, instruction decoding, types of hardware control with micro and nano programming

Memory management: virtual, caching, interleaving, memory hierarchy

Management of peripherals: DMA and interrupt controllers, serial and parallel communication

Peripherals and memory: static and dynamic RAM, tapes, disks, a terminal, keyboard and monitor

Advanced architectures: RISC, pipelining, vector, superscalar, MIPS R10000, PVP, SMP, MPP, NUMA, GPGPU

Introduction of Intel processors (8085 and x86 family), AMD etc.

Programming of a MIPS RISC processor, instruction set and assembly language. Programming of simple input (keyboard) and output (VGA) devices, and a device driver.

Demonstration of the syllabus coherence with the curricular unit's learning objectives

The main goal of the theoretical (T) lectures is to understand the technological developments during the evolution of computers, and how microprocessors and some input and output devices work.

Also the notion of benchmarking will be approached. The practical part (TP) is an introduction to programming at assembly level (RISC) and the interface with C, as well as the programming of input (keyboard) and output (VGA) devices and the use of a device driver (for example the realtime clock). The syllabus of the theoretical part covers a wide range of topics, all related to architecture in general, whereas the TP syllabus targets assembly programming at an introduction level, for simplicity restricted to a RISC architecture, and interfacing assembly code with C code. Since few students will ever need to program in assembly on a CISC architecture, this topic is not covered by the syllabus.

Teaching methodologies (including evaluation)

In the theoretical lectures the basic principles are introduced using real systems. Apart from attending classes, with discussions about the different aspects, students are encouraged to use online information for developing a broader view. There is time to discuss problems and doubts in the theoretical lectures.

In the TP classes the basic programming concepts at the lowest level are introduced: assembly. Using an emulator of the MIPS RISC instruction set, like MARS, students learn to write and test simple programs including recursivity, also interfacing assembly with C. Programming of input and output devices and a device driver is studied.

Evaluation: written examinations in which no notes and other material (60%T, 40%TP). Both parts T and TP are compulsory. The students can opt for TP continuous evaluation (exercises), and can improve the TP mark in the examination. Approval: TP mark greater than 4.0 AND T+TP mark greater than 9.5 (0 to 20 scale).

Demonstration of the coherence between the teaching methodologies and the learning outcomes

In the theoretical lectures, many slides illustrating many different architectures and systems are used in order to show basic aspects and solutions, also to discuss specific advantages and the integration of these in modern computers. We think that, for most students, presenting a general overview is much more important than analysing one architecture in all its details, because of the fast developments which will continue in the future. Concerning the TP part, a RISC architecture (MIPS instruction set) is introduced, step by step, with hands-on experience in assembly programming by using an emulator like MARS. Hence, the students will get a wide overview in general for understanding new technological developments, plus a detailed view of one RISC architecture

Main Bibliography

Computer architecture and organization. J.P. Hayes, McGraw-Hill 1988 and new edition

The 8086/8088 Family: Design, Programming, and Interfacing. J. Uffenbeck, Prentice-Hall 1987

Computer architecture, a quantitative approach. J.L. Hennessy and D.A. Patterson, Morgan Kaufmann, 1990 and later

The Intel microprocessors 8086. 80486: Architecture programming and interfacing. 3rd Ed. B.B. Brey, Prentice-Hall/Macmillan, 1994

8086 ... 80486 Assembly language programming. B.B. Brey, Prentice-Hall/Macmillan, 1994

Microprocessors. Theory and applications. M. Rafiquzzaman, Prentice-Hall, 1992

The 68000 microprocessor. Hardware and software principles and applications. J.L. Antonakos, Macmillan, 1993, 2nd Ed

Conhecendo a familia 80486. B. Segal et al., Livros Erica Edit, 1992

Computer organization and architecture. Principles of structure and function. W. Stallings, Macmillan, 1993, 3rd Ed

Sistemas digitais. Antonio J.G. Padilla, McGraw-Hill, 1993

All INTEL books, like i486 Microprocessor