
1. Caracterização da Unidade Curricular

1.1 Designação

[2226] Arquitetura de Computadores / Computers Architecture

1.2 Sigla da área científica em que se insere

IC

1.3 Duração

Unidade Curricular Semestral

1.4 Horas de trabalho

160h 00m

1.5 Horas de contacto

Total: 67h 30m das quais T: 45h 00m | P: 22h 30m

1.6 ECTS

6

1.7 Observações

Unidade Curricular Obrigatória

Unidade Curricular comum ao(s) curso(s) de LEIC, LEIRT

2. Docente responsável

[1743] Tiago Miguel Braga da Silva Dias

3. Docentes e respetivas cargas letivas na unidade curricular

Não existem docentes definidos para esta unidade curricular

4. Objetivos de aprendizagem (conhecimentos, aptidões e competências a desenvolver pelos estudantes)

Os estudantes que terminarem com sucesso esta UC serão capazes de:

1. Demonstrar conhecimento em arquiteturas de computadores;
2. Demonstrar conhecimento na estrutura interna de um processador;
3. Escrever programas de pequena complexidade usando a linguagem assembly;
4. Utilizar ferramentas para produção de programas escritos em linguagem assembly e para a depuração dos seus erros;
5. Demonstrar conhecimento no mapeamento de memória física no espaço de endereçamento do processador e na sua utilização;
6. Demonstrar conhecimento no mapeamento de dispositivos periféricos no espaço de endereçamento do processador e na sua utilização, tais como portos paralelos de entrada/saída e temporizadores/contadores;
7. Demonstrar conhecimento na utilização do sistema de interrupção externa dos processadores.



**4. Intended learning outcomes
(knowledge, skills and
competences to be developed
by the students)**

Students who successfully complete this course will be able to:

1. Demonstrate knowledge of computer architectures;
2. Demonstrate knowledge of the internal structure of a processor;
3. Write low complexity programs using the assembly language;
4. Use software development tools for production and debugging of assembly programs;
5. Demonstrate knowledge of physical memory mapping onto the address space of a processor and its usage;
6. Demonstrate knowledge about peripheral usage and its mapping onto the address space of a processor, namely input/output parallel ports and timers/counters;
7. Demonstrate knowledge of the processor interrupt subsystem and its usage.

5. Conteúdos programáticos

- I. Introdução: áreas de aplicação de computadores; conceito de programa e modelo de execução; componentes de um computador; evolução tecnológica.
- II. Programação assembly: arquitetura do conjunto de instruções, tipos de instruções e operandos, modos de endereçamento; rotinas e convenções de utilização; pilha e sua utilização; produção e teste de programas.
- III. Arquitetura de um processador: codificação de instruções; unidade de processamento de dados, unidade lógico-aritmética e tipos de registos; unidade de controlo; modelos Harvard e Von-Neumann.
- IV. Subsistema de memória: tecnologias de memórias; espaço de endereçamento e barramentos de dados, endereço e controlo; organização de módulos de memória; hierarquia de memória.
- V. Subsistema de entrada/saída: tipos de dispositivos e suas características; portos paralelos; temporizadores/contadores.
- VI. Mecanismo de interrupção: conceitos de exceção e interrupção; vetorização e níveis de prioridade; processamento de pedidos de interrupção.

5. Syllabus

- I. Introduction: application areas of computers; concept of computer program and its execution model; components of a computer; technological evolution of processors and memories.
- II. Assembly programming: instruction set architecture, types of instructions, operands and addressing modes; concept of routine and calling conventions; concept of stack and its implementation; tools for the development and test of programs.
- III. Processor architecture: instruction coding; datapath, arithmetic and logic unit and types of registers; control unit; the Harvard and Von Neumann models.
- IV. Memory subsystem: memory technologies, processor address space; data, address and control buses; organization of memory modules; memory hierarchy.
- V. Input/output subsystem: types and characteristics of input/output devices; parallel ports; counters/timers;
- VI. Interrupt subsystem: concepts of exception and interrupt; interrupt vectorization and priority levels; processing of interrupt requests.

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

Esta UC tem como objetivos principais habilitar os alunos para compreenderem e utilizarem sistemas computacionais baseados em microprocessadores. Começa-se por apresentar o conceito de sistema computacional baseado em microprocessador e os seus principais componentes (I). Depois, introduz-se um processador didático de 16 bits que suporta o estudo dos restantes tópicos. Tendo em conta as características deste processador, abordam-se os tópicos da programação em assembly e os processos de geração, carregamento em memória e execução de programas (II). São analisados os principais módulos funcionais do processador e identificadas as preocupações no seu desenho (III). No âmbito dos outros subsistemas, é abordada a interface e a descodificação de memória física mapeada no espaço de endereçamento do processador (IV) e são introduzidos os periféricos fundamentais, i.e. portas paralelas e temporizadores/contadores (V). Por último, é estudado o subsistema de interrupções deste processador (VI).

6. Evidence of the syllabus coherence with the curricular unit's intended learning outcomes

The main goal of this CU is to enable students to understand and use microprocessor-based computer systems. First, the concept of a computer system based on a microprocessor and the main components of such systems are presented (I). Then, a 16-bit didactic processor is introduced. This processor supports the study of the remaining course topics. Such study starts by addressing assembly programming. This includes the program development process, as well as the loading of the programs into memory and its execution and debugging procedures (II). Then, the main functional modules of the processor are discussed (III). The memory and I/O subsystems are studied afterwards. Such study covers not only the mapping of the physical memory and peripherals onto the processor address space but also the corresponding interfaces (IV). Moreover, two fundamental peripherals are also studied: parallel I/O ports and timers/counters (V). Finally, the processor interruption subsystem is analyzed (VI).

7. Metodologias de ensino (avaliação incluída)

Ensino teórico e prático com recurso ao laboratório para aplicação dos principais conceitos. As aulas visam a apresentação e explicação dos temas, sendo os alunos também confrontados com exercícios e trabalhos em laboratório para experimentação das valências adquiridas. Ademais, durante o semestre são realizados trabalhos práticos para consolidação da aprendizagem. Esses trabalhos são realizados em grupo. Os resultados da aprendizagem (1) a (7) são avaliados através dos relatórios dos trabalhos práticos (T) realizados no semestre e do projeto final (P) e, sempre que necessário, de uma prova oral de validação da contribuição de cada aluno para o trabalho. P é considerado pedagogicamente fundamental por agregar em contexto prático todos os tópicos tratados na UC. Os resultados da aprendizagem (2), (3), (5), (6) e (7) são também avaliados através de exame final escrito (E). A nota final é definida por $0,3*T+0,3*P+0,4*E$. A classificação mínima exigida em P e E é 9,5 valores.

**7. Teaching methodologies
(including assessment)**

Theoretical and practical teaching supported by laboratory classes for the demonstration of the main syllabus topics.

The classes are devoted to the presentation and the explanation of the syllabus contents. Also, the students are asked to solve problems and to conduct hands-on activities on the laboratory to consolidate knowledge about specific syllabus topics. Such activities are performed in groups.

The learning outcomes (1) to (7) are assessed through the reports of the course assignments (T) and the final project (P) and, whenever necessary, a viva voce examination to validate the individual contribution of each student. P is considered to be a pedagogically fundamental activity. The learning outcomes (2), (3), (5), (6) and (7) are further assessed in a final written exam (E).

The final grade is calculated according to the following formula: $0.3 \cdot T + 0.3 \cdot P + 0.4 \cdot E$. A minimum classification of 9.5 values is required in P and E.

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

Nas aulas teóricas e práticas são apresentados e discutidos os conceitos teóricos de forma interativa. Com o propósito de facilitar a compreensão dos conteúdos programáticos, também são analisados casos de estudo relevantes e resolvidos exercícios sobre problemas concretos. Os alunos são incentivados a participar, quer respondendo a questões colocadas pelo docente, quer colocando questões ou dando contributos para a resolução coletiva dos problemas propostos. As atividades laboratoriais visam ajudar os alunos a consolidar os conceitos teóricos apresentados nas aulas e a apoiar a realização do trabalho de projeto da UC, em que os alunos são confrontados com a especificação de um sistema para desenhar e implementar uma solução. Esta abordagem, com momentos de exposição da teoria seguidos de momentos de resolução de exercícios e em laboratório conduz a que os alunos assimilem mais facilmente os tópicos tratados na UC.

Os trabalhos práticos de avaliação realizados durante o semestre e o exame escrito final permitem aferir a qualidade da aprendizagem desses conceitos, enquanto o trabalho de projeto permite aferir as boas práticas no desenvolvimento de programas em linguagem assembly e avaliar a capacidade dos alunos para utilizarem microprocessadores no desenvolvimento de sistemas computacionais de baixa complexidade. A realização de uma prova oral dos trabalhos práticos e do projeto visa aferir, com maior rigor, o nível de conhecimento e o envolvimento de cada aluno sobre os trabalhos realizados em grupo.

**8. Evidence of the teaching
methodologies coherence with
the curricular unit's intended
learning outcomes**

The theoretical concepts of the syllabus contents are presented in both the theoretical and the practical classes by using an interactive methodology. Moreover, relevant case studies are discussed with the students and problem-solving activities are carried out to facilitate their understanding of the most important syllabus topics. Students are encouraged to participate by being summoned by the teacher to answer some questions and by providing contributions to the resolution of the proposed problems. The goal of the laboratory activities is to help students consolidate the theoretical concepts and support the realization of the course project. This project presents students with the specification of a dedicated computational system that they must design and implement. Altogether this approach allows the students to more easily acquire the main topics covered in this course.

The written reports of the assignments conducted during the semester and the final written exam allow assessing the quality of the learning outcomes, while the project assignment further allows evaluating the students' skills to write good computer programs using the assembly language and their ability to use microprocessors to develop low complexity computational systems. The realization of viva voce examinations of the practical assignments and the course project enables evaluating more accurately not only the skills and the competences acquired by the students with the group assignments but also the individual contribution of each student to the realization of such assignments.

**9. Bibliografia de
consulta/existência obrigatória**

- Harris, S. L. & Harris, D. M. (2015). Digital Design and Computer Architecture: ARM Edition, Waltham, MA: Morgan Kaufmann Publishers, ISBN: 978-0128000564.
- Patterson, D. A. & Hennessy, J. L. (2016). Computer Organization and Design: The Hardware/Software Interface - ARM Edition, Cambridge, MA: Morgan Kaufmann Publishers, ISBN-13: 978-0128017333.
- Mano, M. & Kime, C. (2015). Logic and Computer Design Fundamentals, 5ª Edição, Upper Saddle River, N.J.: Pearson Education, ISBN-13: 978-0133760637.

10. Data de aprovação em CTC 2024-07-17 2024-07-17 2024-07-17

11. Data de aprovação em CP 2024-06-26 2024-06-26 2024-06-26