
1. Designação da unidade curricular

[4448] Programação de Computadores e Sistemas / Computer and Systems Programming

2. Sigla da área científica em que se insere IC

3. Duração Unidade Curricular Semestral

4. Horas de trabalho 162h 00m

5. Horas de contacto Total: 67h 30m das quais T: 27h 30m | TP: 27h 30m | P: 12h 30m

6. % Horas de contacto a distância Sem horas de contacto à distância

7. ECTS 6

8. Docente responsável e respetiva carga letiva na Unidade Curricular [818] Mário Henrique Carrasqueira Simões | Horas Previstas: N/D

9. Outros docentes e respetivas cargas letivas na unidade curricular Não existem docentes definidos para esta unidade curricular

10. Objetivos de aprendizagem e a sua compatibilidade com o método de ensino (conhecimentos, aptidões e competências a desenvolver pelos estudantes).

Os estudantes que terminam com sucesso esta unidade curricular serão capazes de:

1. Implementar, testar e corrigir programas de complexidade média, organizados em módulos escritos em linguagem C, podendo envolver o uso de bibliotecas, normalizadas ou outras;
2. Relacionar os mecanismos da linguagem C com o modelo computacional que suporta a execução;
3. Integrar módulos em assembly nos programas desenvolvidos em linguagem C;
4. Descrever como os programas são compilados, ligados, carregados e executados, incluindo o uso de bibliotecas com os métodos de ligação estática ou dinâmica, e a criação de novas bibliotecas.
5. Identificar o contributo dos mecanismos de cache para a eficiência da execução.

10. Intended Learning objectives and their compatibility with the teaching method (knowledge, skills and competences by the students).

Students who successfully complete this course unit will be able to:

1. Implement, test, and debug medium-complexity programs, organized into modules written in C, which may involve the use of standard or external libraries.
2. Relate the mechanisms of the C language to the computational model that supports execution.
3. Integrate assembly modules into programs developed in C language.
4. Describe how programs are compiled, linked, loaded, and executed, including the use of libraries with static or dynamic linking methods, and the creation of new libraries.
5. Identify the contribution of caching mechanisms to execution efficiency.

11. Conteúdos programáticos

- I. Caracterização da linguagem C. Mecanismos de controlo. Tipos básicos. Constantes e variáveis. Expressões. Tipos struct e arrays. Ponteiros e sua aritmética. Biblioteca padrão.
- II. Gestão da memória nos programas escritos em linguagem C (alojamento permanente, variáveis automáticas e alojamento dinâmico). Programação genérica.
- III. Organização dos programas em módulos. Compilação, ligação e execução de programas. Conceito de pré-compilação. Criação e utilização de macros e de ficheiros para inclusão. Automatização do processo de compilação e ligação.
- IV. Escrita de programas em assembly. Convenções de chamada a funções; compatibilidade com a linguagem C. Conceito de stack frame e sua manipulação. Programas envolvendo código em C e assembly. Automatização do processo de compilação e ligação.
- V. Ligação estática e dinâmica. Uso de bibliotecas externas e criação de bibliotecas.
- VI. Hierarquia de memória. Sistemas de cache e o seu impacto no desempenho dos programas.

11. Syllabus

- Characterization of the C language. Control mechanisms. Basic types. Constants and variables. Expressions. Struct and array types. Pointers and their arithmetic. C standard library.
- II. Memory management in programs written in C (static allocation, automatic variables, and dynamic allocation). Generic programming.
- III. Organization of programs into modules. Compilation, linking and execution of programs. Pre-compilation concept. Creation and use of macros and files for inclusion. Automation of the compilation and linking process.
- IV. Writing programs in assembly language. Function calling conventions: C language calling conventions. Programs involving both C and assembly code. Traversal and manipulation of stack frames.
- V. Static and dynamic linking. Use of external libraries and creation of libraries.
- VI. Memory hierarchy. Cache systems and their impact on program performance.

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

- Os temas são introduzidos por ordem crescente de complexidade.
- Os objetivos 1 é suportado pelo somatório dos conhecimentos adquiridos em todos os temas.
- O objetivo 2 é suportado pelos temas II e IV.
- O objetivo 3 é suportado pelo tema IV.
- O objetivo 4 é suportado pelos temas III e V.
- O objetivo 5 é suportado pelo tema VI.

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

Themes are introduced in order of increasing complexity.

Objective 1 is supported by the sum of knowledge acquired in all topics.

Objective 2 is supported by topics II and IV.

Objective 3 is supported by topics IV.

Objective 4 is supported by topics III and V.

Objective 5 is supported by topics VI.

13. Metodologias de ensino e de aprendizagem específicas da unidade curricular articuladas com o modelo pedagógico

O ensino é composto pelas partes teórica (T), teórico-prática (TP) e prática laboratorial (PL), não havendo uma separação estanque entre as aulas dedicadas a cada uma destas partes. Em cada tema, há exposição de conceitos (T) e respetivos casos de aplicação, acompanhados por exercícios exemplificativos (TP), complementando com exercícios de consolidação (PL) realizados pelos alunos. A fração do tempo atribuída a PL inclui o apoio aos alunos para a realização dos trabalhos práticos.

13. Teaching and learning methodologies specific to the curricular unit articulated with the pedagogical model

Teaching is made up of theoretical (T), theoretical-practical (TP) and laboratory practice (PL) parts, with no clear separation between the classes dedicated to each of these parts. In each topic, there is an exposition of concepts (T) and respective application cases, accompanied by exemplary exercises (TP), complemented by consolidation exercises (PL) carried out by the students. The fraction of time allocated to PL includes support for students to carry out practical work.

14. Avaliação

A avaliação tem as componentes prática (P) e escrita (E).

A componente P, pedagogicamente fundamental, é baseada em três trabalhos práticos realizados ao longo do tempo letivo, cujas classificações são designadas por P1, P2 e P3. Os alunos realizam os trabalhos em grupo com o máximo de três elementos, fora das horas de contacto. As classificações P1 e P2 são atribuídas pela apreciação dos respetivos trabalhos submetidos pelo grupo de alunos. A classificação P3 é obtida por discussão, realizada em grupo, sendo atribuídas classificações individuais. A ponderação é $P = 0,2\text{ P1} + 0,2\text{ P2} + 0,6\text{ P3}$; Os mínimos para prosseguir a avaliação são: $\text{P1} \geq 8,0$; $\text{P2} \geq 8,0$; $\text{P3} \geq 8,0$; $P \geq 9,5$.

A componente E obtém-se por exame final, sendo admitidos os alunos aprovados na parte P. O exame é realizado na época normal de exame (EN) ou na época de recurso (ER). Se o aluno, aprovado em P, reprovar no exame de EN é admitido à ER. O mínimo para aprovação é $E \geq 9,5$.

A classificação final é: $CF = 0,5\text{ P} + 0,5\text{ E}$.

14. Assessment

The assessment has practical (P) and written (E) components.

The P component, pedagogically fundamental, is based on three practical works carried out throughout the academic period, whose classifications are designated as P1, P2 and P3. Students perform the work in group with a maximum of three members, outside contact hours. Classifications P1 and P2 are awarded based on the assessment of the respective works submitted by the group of students. The P3 classification is obtained by discussion, carried out in groups, with individual classifications being assigned. The weighting is $P = 0.2 P1 + 0.2 P2 + 0.6 P3$; The minimum values to continue the evaluation are: $P1 \geq 8.0$; $P2 \geq 8.0$; $P3 \geq 8.0$; $P \geq 9.5$.

Component E is obtained through a final exam, with students approved in part P being admitted. The exam is carried out during the normal exam period (EN) or during the recurrent exam period (ER). If the student, who passed P, fails the EN exam, will be admitted to ER. The minimum for approval is $E \geq 9.5$.

The final classification is: $CF = 0.5 P + 0.5 E$.

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

Os estudantes obtiveram, em unidades curriculares anteriores: na vertente de software, os conceitos gerais de programação, algoritmia e estruturas de dados; na vertente de hardware, os conceitos de arquitetura de computadores.

Nas aulas são apresentados os temas, de I a VII, com vista ao domínio dos objetivos referidos. A exposição é complementada, em interação com os estudantes, pela realização, teste e correção de programas, em exercícios exemplificativos e de consolidação dos temas expostos. Estes programas, além de concretizarem os conceitos de programação lecionados, são usados para explorar as ferramentas de programação, com ênfase na metodologia de depuração de erros.

Os trabalhos práticos, têm complexidade crescente e são alinhados com os temas das aulas, que fornecem conceitos de base para a sua realização. Nos trabalhos, os estudantes confrontam-se com problemas e dúvidas que, ao elaborar soluções, promovem o desenvolvimento das suas competências.

O plano de trabalhos práticos está estruturado em três séries de exercícios (SE1, SE2 e SE3) organizadas por grupos de temas. A escolha dos temas agrupados nas sucessivas séries de exercícios é orientada para promover os objetivos de aprendizagem de acordo com a relação indicada em 12.

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

Students obtained, in previous curricular units: in the software aspect, the general concepts of programming, algorithms and data structures; on the hardware side, the concepts of computer architecture.

In classes, themes from I to VII are presented, with a view to mastering the aforementioned objectives. The exposition is complemented, in interaction with students, by carrying out, testing and correcting programs, in exemplary and consolidating exercises, the exposed themes. These programs, in addition to implementing the programming concepts taught, are used to explore programming tools, with an emphasis on error debugging methodology.

The practical work has increasing complexity and is aligned with the themes of the classes, which provide basic concepts for its implementation. In their work, students are faced with problems and doubts that, when developing solutions, promote the development of their skills.

The practical work plan is structured into three series of exercises (SE1, SE2, and SE3) organized by groups of themes. The choice of themes grouped in the successive series of exercises is aimed at promoting the learning objectives in accordance with the list indicated in 12.

16. Bibliografia de consulta/existência obrigatória

C Programming - A Modern Approach, 2nd Ed.; K. N. King; W. W. Norton, 2008; ISBN 9780393979503

The C Programming Language, 2nd Ed.; Dennis M. Ritchie, Brian W. Kernighan; Prentice Hall, 1988; ISBN: 9780133086249/9780131103627

Low-Level Programming: C, Assembly, and Program Execution on Intel® 64 Architecture; Igor Zhirkov; Apress, 2017; ISBN: 9781484224021/9781484224038

17. Observações

Unidade Curricular Obrigatória

Data de aprovação em CTC:

Data de aprovação em CP: