COURSE DATA

Data Subject	
Code	34659
Name	Computer architecture
Cycle	Grade
ECTS Credits	6.0
Academic year	2023 - 2024

Study (s)

Degree	Center	Acad. year	Period
1400 - Degree in Computer Engineering	School of Engineering	3	Second term

Subject-matter

Degree	Subject-matter	Character
1400 - Degree in Computer Engineering	6 - Computer engineering	Obligatory

Coordination

Name	Department
PARDO CARPIO, FERNANDO	240 - Computer Science

SUMMARY

"Computer Architecture" is part of the "Computer Engineering" matter, it is a compulsory subject in the third year of the Degree of Computer Engineering, taught in the second half, which has a load of 6 ECTS.

This subject completes the knowledge of the technology and performance of computers started in previous subjects of "Computer Technology" and "Basis of computers" in the first year, and continued in the subjects, "Structure of Computers" and "Computer Organization", taught in second year.

This subject completes the study of the architectural paradigms of current computers initiated in previous courses. Therefore, after a review of key concepts studied in the course of Computer Organization, the subject focuses on programming models for parallel and distributed computers. In particular, it is focused on current "de facto" standards, such as OpenMP and MPI.

The networks of parallel computers, multiprocessors and multicomputers, are reviewed in detail, studying their most important features: topology, routing, switching mechanisms, virtual channels, etc.

Throughout the subject, techniques and models to evaluate the features and performance of parallel computers are studied. These are based on the characteristics of these sort of computers, in order to establish the best solution for a given problem.

PREVIOUS KNOWLEDGE

Relationship to other subjects of the same degree

There are no specified enrollment restrictions with other subjects of the curriculum.

Other requirements

In order to properly study this course, students should be seated knowledge and learning outcomes that have been in the degrees related to programming, the structure and functioning of the computer. In particular, it is recommended that you have completed the previous courses that are part of this matter ("Computer Structure" and "Computer Organization") and the subjects of "Programming", "Data Structures and Algorithms" and "Languages and Programming paradigms "

OUTCOMES

1400 - Degree in Computer Engineering

- G1 - Ability to design, write, organise, plan, develop and sign projects in the field of computer engineering aimed at the design, development or exploitation of computer systems, services and applications.
- G4 - Ability to define, evaluate and select hardware and software platforms for the development and implementation of computer systems, services and applications, in accordance with both the knowledge and the specific skills acquired in the degree.
- G6 - Ability to design and develop computer systems and centralised or distributed computer architectures which integrate hardware, software and networks, in accordance with both the knowledge and the specific skills acquired in the degree.
- G8 - Knowledge of basic subject areas and technologies that serve as a basis for learning and developing new methods and technologies, and of those which provide versatility to adapt to new situations.
- G10 - Knowledge to perform measurements, calculations, assessments, appraisals, surveys, studies, reports, scheduling and other similar work in the field of computer engineering, in accordance with both the knowledge and the specific skills acquired in the degree.

- R1 - Ability to design, develop, select and evaluate computer applications and systems while ensuring their reliability, safety and quality, according to ethical principles and current legislation and regulations.
- R4 - Ability to draw up the technical specifications of a computer system, according to standards and regulations.
- R6 - Knowledge and application of basic algorithmic procedures of computer technology to design solutions to problems, by analysing the suitability and complexity of the algorithms proposed.
- R7 - Knowledge, design and efficient use of the types and structures of data most suitable for solving a problem.
- R14 - Knowledge and application of the fundamental principles and basic techniques of parallel, concurrent, distributed and real-time programming.
- TI2 - Ability to select, design, implement, integrate, evaluate, build, manage, exploit and maintain hardware, software and network technologies, within adequate cost and quality thresholds.
- IC1 - Ability to develop specific processors and embedded systems, and to develop and optimise software for such systems.
- IC3 - Ability to analyse, evaluate, select and configure hardware platforms for the development and implementation of computer applications and services.

LEARNING OUTCOMES

This course allows for the following learning outcomes:

• Ability to calculate the relevance of scalability in the performance of a system.

• Assess the advantages of parallel processing, establishing metrics for comparison.

• Assess the performance impact of the interconnection network of a parallel system according to their different designs.

• Effective use multiprocessors in response to the memory organization.

• Assess the advantages and disadvantages of different parallel architectures.

• Design simple programs on a system with multiple processing elements.

• Work together to make necessary designs and configurations, distributing the workload to address complex problems.

To complement the above results, this subject also to acquire the following skills and social skills:

• Logical reasoning.

• Analysis and synthesis problems.

• Oral and written expression.

• Individual work capacity.

• Ability to work in groups.

DESCRIPTION OF CONTENTS

1. Parallel architectures and performance analysis

Review of computer architectures.
Parallel computers: Multiprocessors and multicomputers.
Challenges, constraints and future opportunities.
Magnitudes and performance models
Labs:
L1. Serial programing. Performance analysis.

2. Parallel programming

Programming models in parallel computers.
Shared (SIMD, SPMD) and distributed (MIMD) memory programming.
Programming standards: OpenMP and MPI.

Labs:
L2. Introduction to OpenMP. Basic programming examples.
L3-4. Application example in OpenMP. Programming of a real application.
L5. Introduction to MPI. Basic programming examples.
L6-7. Application example in MPI. Programming of a real application.

3. Interconnection networks

Rating: definitions, properties and problems (delay, power, noise, scalability and reliability).
Routers: Elements and features.
Switching: circuit and packet switching (Store & Forward), VCT and Wormhole.
Flow Control: Features, Type: HW, Stop & Go, based on credits.
Dynamic networks: Features. Topologies: crossbar and MIN (Multiple Interconnection Network).
Static Networks: Characteristics. Regular and irregular topologies: bus, ring, mesh, torus, hypercube, d-dimensional mesh, d-dim k-ary and d-cube k-ary.
Routing Algorithms. Features: livelock, starvation and deadlock. Types: Based on origin, centralized, distributed (based on tables, arithmetic). Deterministic / semi-adaptive / adaptive. Deadlock-free algorithms. Testing Techniques (GDC). Virtual channels.
Current architectures and implementations: CMP, NoC, Clusters, etc.

Labs:
L8. Multicomputer networks

WORKLOAD

ACTIVITY	Hours	% To be attended
Theory classes	30,00	100
Laboratory practices	20,00	100
Classroom practices	10,00	100
Development of group work	20,00	0
Development of individual work	10,00	0
Study and independent work	10,00	0
Readings supplementary material	5,00	0
Preparation of evaluation activities	10,00	0
Preparing lectures	10,00	0
Preparation of practical classes and problem	15,00	0
Resolution of case studies	10,00	0
TOTAL	**150,00**	

TEACHING METHODOLOGY

Theoretical activities.

Description: The lectures will develop the issues by providing a global and inclusive vision, analyzing in detail the key issues and more complex, encouraging at all times, participation of students. The workload for students in this section of the total charge of the matter is 19%.

Practical activities.

Description: Complementing theoretical activities in order to apply the basics and expand the knowledge and experience to be acquired in the course of the work proposed. They include the following types of classroom activities:

- Classes of problems and issues in the classroom
- Regular discussion and resolution of problems and exercises that the students have previously worked
- Labs
- Oral presentations
- Tutoring schedule (individualized or group)
- Conducting individual evaluation questionnaires in the classroom with the presence of teachers.

The workload for students on the total charge of the matter is 21%.

Personal work.

Description: Realization (outside the classroom) of monographs, literature search directed, issues and problems as well as the preparation of classes and exams (study). This is done individually and tries to promote self-employment. The workload for students on the total charge of the matter is 45%

Working in small groups.

Description: Realization, by small groups of students (2-4) of work, issues, problems outside the classroom. This work complements the work and encourages individual ability to integrate into working groups. The workload for students on the total charge of the matter is 15%.

It will use the platform of e-learning (virtual classroom) of the University of Valencia in support of communication with students. Through it you will have access to course materials used in class as well as solve problems and exercises.

EVALUATION

The assessment of the subject will be the sum of the following sections:

Continuous assessment (P), based on attendance and participation in the teaching-learning process. This section will be proposed classroom activities and resolving issues and problems along the course. This part will count 10% of the final mark and will consist of assistance and delivery by the student, individually or in groups, exercises and questions raised as a result of preparation or attendance and participation sessions in solving problems through the forum of the course or schedule tasks, answering to the proposed problems. The student must have attended and/or delivered 60% of these works to score in this part and the score for each exercise or question will depend on its complexity. If the 60% is not reached, the student will obtain a Not Present mark in the first call. This activity will not be recoverable in second call.

Evaluation of laboratory activities (L) from the achievement of objectives in group (LG) and individual tests (LI) at the end of the laboratory sessions. These activities will be carried out individually and/or in group, and its weight is 25% of the final grade. To score in this section, the student should obtained at least a 4 mark in all activities where this condition is explicitely stated, or an average of 6 otherwise. The mark of this part will be computed as the weighted average of LG and LI. This activity will not be recoverable in second call.

Individual objective tests, consisting of several exams or knowledge tests, which consist of both theoretical and practical issues as problems that will occur in the middle of the semester (called T1), and out of the teaching time in the examination period (called T2). This part of the evaluation (T) will count 65% of the final grade and the distribution of each individual test is as follows:

T = max{0.3 * T1 + 0.7 * T2,T2}.

Each one of these tests covers all course contents taught at that time. If T is below 4, NA1 will be 4 or less.

The mark will be formed in the case of follow the continuous assessment as the sum of the previous parts as follows:

NA1=0.10*P+0.25*L+0.65*T

In the second call, students must submit to a final exam (FE) and the final grade is computed as:

NA2=max{0.10*P+0.25*L+0.65*EF, 0.05*P+0.20*L+0.75*EF}

It will be necessary to obtain a minimum mark of 4 in EF to pass the module. If EF is below 4, NA2 will be 4 or less.

In order request the advanced call, students must have attended the subject in a previous year, and the same assessment norms as in the second call will apply.

In any case, the evaluation of this subject will be done in compliance with the University Regulations in this regard, approved by the Governing Council on 30th May 2017 (ACGUV 108/2017).

REFERENCES

Basic

- Apuntes de la asignaturas, transparencias de las presentaciones de clase y vídeos
- Arquitectura de Computadores, Julio Ortega, Mancia Anguita, Alberto Prieto, Thomson-Paraninfo, 2005.
- J.L. Hennessy, D.A. Patterson Computer Architecture: A Quantitative Approach, 5ª edición Morgan Kaufmann Publishers, 2012.
- J. Duato, S. Yalamanchili and L.M. Ni, Interconnection networks: An engineering approach, 1ª y 2ª edición IEEE Computer Society Press 1997 y Morgan Kaufmann 2002
- Barry Willkinson y Michael Allen. Parallel Programming. Pearson (2005)
- Gerassimos Barlas: Multicore and GPU Programming, 1ª edición, Morgan Kaufmann, 2015.

Additional

- Parallel Computer Architecture, David E. Culler, Jaswinder P. Singh, with Anoop Gupta, Morgan Kaufmann, 1998.
- Introduction to Parallel Computing, second edition, A. Grama, A. Gupta et al., Addison-Wesley, 2003.

- Arquitectura de computadoras y procesamiento en paralelo, Kai Hwang, Fayé A. Briggs, Mc-Graw Hill, 1990.
- D. Sima, T. Fountain, P. Kacsuk Advanced computer architectures: A design space approach Addison Wesley, 1997
- OpenMP Tutorial, Blaise Barney, Lawrence Livermore National Laboratory, disponible en https://computing.llnl.gov/tutorials/openMP/