

Computer Architecture

Code: 102775
ECTS Credits: 6

Degree	Type	Year	Semester
2502441 Computer Engineering	OB	2	2

The proposed teaching and assessment methodology that appear in the guide may be subject to changes as a result of the restrictions to face-to-face class attendance imposed by the health authorities.

Contact

Name: Miquel Àngel Senar Rosell
Email: MiquelAngel.Senar@uab.cat

Use of Languages

Principal working language: catalan (cat)
Some groups entirely in English: No
Some groups entirely in Catalan: Yes
Some groups entirely in Spanish: Yes

Teachers

Juan Carlos Moure Lopez
Xavier Cano de Castro
Xiaoyuan Yang
Santiago Marco Sola
Gemma Sanjuán Figuerola
Javier Panadero Martinez

Prerequisites

There are no prerequisites. However, students should have a good knowledge of the C programming language and be familiar with the basic concepts of computer structure, including the organization of the memory hierarchy and assembly language.

Objectives and Contextualisation

1. To understand architectural techniques based on parallelism to improve computer performance.
2. To understand the principle of data access locality and the architectural solutions applied to the memory hierarchy.
3. To describe techniques for performance evaluation, the metrics used, and methods for result visualization.
4. To evaluate the performance of application fragments in multi-core and GPU computing systems, selecting the appropriate evaluation techniques.
5. To analyze performance bottlenecks in the execution of a program fragment: execution limits due to capacity of computer resources, execution limits due to data dependencies and latencies of operations, and limits due to operation misses in the memory hierarchy.

6. To use performance analysis techniques to select the computer system appropriate to an application and / or apply optimizations of the code that improve their parallelism (at the level of instruction and threads) and the locality of data accesses.

Competences

- Capacity to design, develop, select and evaluate computer applications and systems, ensuring reliability, security and quality, in accordance with ethical principles, and applicable standards and legislation.
- Have the capacity to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications.
- Have the right personal attitude.
- Know and apply the basic and main techniques of parallel, concurrent, distributed and real time programming.
- Know the characteristics, functionalities and structure of operating systems and design and implement applications based on their services.
- Know, understand and evaluate the structure and architecture of computers, as well as the basic components that they are composed of.

Learning Outcomes

1. Analyse and understand the translation that a computer makes of the original source code to generate the executable binary code.
2. Analyse the different design options for a computer in accordance with technical (features) and economic (costs) parameters.
3. Create efficient code to be run in multi-core and multiprocessor architectures.
4. Design, develop, select and evaluate computer systems, ensuring their reliability, security and quality.
5. Generate proposals that are innovative and competitive.
6. Identify performance anomalies and inefficiencies when running a uniprocessor computer program both in terms of computing and memory access, and propose modifications to the program to improve performance.
7. Identify the architecture and organisation of current microprocessors, from uniprocessor to multi-core systems.
8. Know and apply the fundamental principles and basic techniques of parallel and real time programming.
9. Know, administer and maintain computer systems in terms of hardware.
10. Manage information by critically incorporating the innovations of ones professional field, and analysing future trends.
11. Measure the run time of a program in a processor and the number of executed instructions.
12. Understand the assembly language to debug errors in source code and to detect performance problems.

Content

1. Fundamentals of Computer Design and Evaluation

- Latency, Parallelism and Locality
- Cost, Performance, Energy Consumption and Reliability
- Metrics and Performance Evaluation Techniques
- Results Visualization Methods

2. Memory Hierarchy: Functionality and Miss Rate

- Principles of the Memory Hierarchy: Cache, Memory and Disk

- Block and Cache line. Placement and replacement algorithms
- Miss Rate and Performance of sequential execution
- Memory Access Patterns
- Code optimizations to improve the locality of data accesses

3. Parallelism at the processor core and the memory hierarchy

- Pipeline execution: Latency and execution capacity
- Multiple execution of instructions and branch prediction
- Dependency analysis and processor capacity limits
- Explicit instructions to take advantage of Data Parallelism (SIMD)
- Parallelism in access to memory hierarchy: Latency and Bandwidth
- Code optimizations that take advantage of the internal parallelism of the processor
- Introduction to parallel systems based on multiprocessors and multicomputers.
- Introduction to distributed systems.

Methodology

Theory classes: main concepts of the subject will be explained. The basic concepts will be described and examples and small problems of how to use them in practice will be indicated. The most important learning problems will be highlighted and will show how to complete and deepen the contents. Practical cases will be discussed and the teacher will detect the most common comprehension and reasoning problems and solve them for all students.

Problem Classes: Cooperative problem solving activities will be carried out. Based on the previous individual work of the students, they will make a group sharing of solutions and they will discuss any doubts that may have arisen. The teacher will detect the most common comprehension and reasoning problems and will solve them in small groups or to all the students. These classes will provide the applied knowledge needed by students to complement theoretical concepts. They will serve as a bridge between theory classes and practical work. These classes are evaluable, and in certain pre-established sessions the students will solve problems delivered at the beginning of the session which should be delivered at the end of the session as an evaluable evidence.

Laboratory Classes: they will support the theory classes. The students will have the practical information in advance before each session, and they will have to prepare the preliminary part indicated in the report so that the teacher, at the beginning of the session, can review it. During the session the students should inform the teacher about their progress and the problems that may be encountered, and at the end of the session they will deliver a document with the results of the exercise and a summary of the problems encountered.

Transversal skills: They will be worked on and evaluated at various times throughout the course. Specifically:

T06.03 - Generate innovative and competitive proposals in the professional activity: throughout the students will work with examples to illustrate how different solutions can be proposed to the same problem, evaluating its relevance from a point of view of the performance obtained. In the laboratory classes, this competence will be evaluated through exercises that students must solve by providing solutions that have original and innovative aspects.

T06.04 - Manage information by critically incorporating the innovations of the professional field itself, and analyze the trends of the future: throughout the course students will be presented with those elements that are most innovative from the point of view of the architecture of new processors. In laboratory classes, specific

questions about these technological innovations will be included, contrasting them with the equipment used in the resolution of the practical exercises.

Annotation: Within the schedule set by the centre or degree programme, 15 minutes of one class will be reserved for students to evaluate their lecturers and their courses or modules through questionnaires.

Activities

Title	Hours	ECTS	Learning Outcomes
Type: Directed			
Exercised-based classes	12	0.48	1, 2, 3, 8, 9, 4, 12, 6
Lab sessions	12	0.48	1, 3, 9, 4, 12, 5, 10, 6, 11
Theoretical classes	26	1.04	1, 2, 3, 8, 9, 4, 5, 10, 6, 7
Type: Autonomous			
Preparing exercises	30	1.2	1, 3, 8, 12, 6, 7, 11
Preparing partial and final tests	30	1.2	1, 2, 3, 8, 12, 5, 10, 6, 7, 11
Preparing practical assignments	32	1.28	1, 3, 8, 12, 5, 10, 6, 7, 11

Assessment

Individual Test with exercises, including theory and laboratory questions 25% 2 0.08 CRI01.11, CRI05.02, CRI09.02, CRI09.03, CRI14.01, CRI14.02, CRI14.03, E04.01, T06.04

Individual Test with exercises, including theory and laboratory questions 25% 2 0.08 CRI01.11, CRI05.02, CRI09.02, CRI09.03, CRI14.01, CRI14.02, CRI14.03, E04.01, T06.04

Exercise resolution in groups 20% 2 0.08 CRI01.12, CRI01.11, CRI09.02, CRI09.03, CRI14.03, E04.01, T06.04

Laboratory exercises in groups 30% 2 0.08 CRI01.12, CRI01.11, CRI05.01, CRI09.02, CRI09.03, CRI14.02, CRI14.03, T06.04, T06.03

Dates for continuous assessment and report submissions will be published on UAB's Moodle platform and may be subject to changes due to possible incidents during the course; Moodle is the usual communication platform between lecturers and students. Students who take the subject again must do the same activities as the rest and there is no compensation mechanism for activities carried out in previous years. Subject assessment will be carried out as follows:

Two individual partial tests (I1, I2, 25% and 25% on the final grade). They are two tests based on exercise resolution, theoretical questions and questions related to laboratory exercises. The evaluation of the student will be based on his test answers.

Exercise resolution in groups and participation in exercise resolution classes (PRB, 20% on the final grade). A series of exercises will be proposed to be solved either individually or in groups at exercise resolution classes. There will be 4 evaluated exercises, at least, throughout the course each will have the same relative weight in the final assessment of this part. Each exercise will be evaluated separately and there will be no re-assessment mechanism for this part.

Laboratory exercises to be solved in groups (Lab, 30% on the final grade), with discussion and help from the lecturer. These exercises will be done over one or two lab sessions and a final report with results, answers and

conclusions must be submitted at the end of the last session (between 2 and 4 lab exercises will be done throughout the course). Student assessment will be done taking into account his active involvement in lab sessions and the submitted reports. There will be no re-assessment mechanism for this part.

A grade equal to or higher than 5 out of 10 is required in the LAB to pass the subject. Otherwise, the subject will be graded as Fail with a numerical value equal to the Lab / 2 mark.

Students don't need to go to the final exam if the scores of their individual test (I1 and I2) meet the following condition

$$((\text{Max (I1, I2)} + 2 * \text{min (I1, I2)}) / 3) \geq 5 \text{ (qualifications I1 and I2 on 10-point scale)}$$

In this case, the final grade of the subject is computed as:

$$\text{FINAL Grade} = (I1 * 2,5 + I2 * 2,5 + \text{PRB} * 0,2 + \text{LAB} * 3) / 10$$

FINAL Grade has to be equal to or greater than 5 to pass the subject.

If the above condition is not met, the student must necessarily take the final exam.

Final exam (F), held at the date specified by the coordination responsible. Student assessment will be based on his answers to the exam.

$$\text{FINAL Grade} = (F * 5 + \text{PRB} * 2 + \text{LAB} * 3) / 10$$

The subject is passed if the Final exam (F) has a mark equal to or higher than 5 and the Final Grade is also equal or greater than 5. If the grade of the Final exam is lower than 5, the subject will have a grade of fail with a numerical value equal to the grade of the Final exam.

In order to pass the course with honors, the final grade must be a 9.0 or higher. Because the number of students with this distinction cannot exceed 5% of the number of students enrolled in the course, this distinction will be awarded to whoever has a more prominent participation throughout the course at the discretion of course lecturers.

A student will be graded as Non Assessable if he has not carried out any partial or final exam (I1, I2, F), or in exceptional cases that will be analyzed and decided by the subject responsible (severe illnesses, accidents,...).

For each individual assessment activities, a place, date and time of review will be indicated allowing students to review the activity with the lecturer. In this context, students may discuss the activity grade awarded by the lecturers responsible for the subject. If students do not take part in this review, no further opportunity will be made available.

NOTES:

a) Apart from the evaluation activities described above, other optional activities may be proposed during the course that may contribute to the final grade of the subject.

b) Notwithstanding other disciplinary measures deemed appropriate, and in accordance with the academic regulations in force, assessment activities will receive a zero whenever a student commits academic irregularities that may alter such assessment. Assessment activities graded in this way and by this procedure will not be re-assessable. If passing the assessment activity or activities in question is required to pass the subject, the awarding of a zero for disciplinary measures will also entail a direct fail for the subject, with no opportunity to re-assess this in the same academic year. Irregularities contemplated in this procedure include, among others:

- the total or partial copying of a practical exercise, report, or any other evaluation activity;

- allowing others to copy;
- presenting group work that has not been done entirely by the members of the group;
- presenting any materials prepared by a third party as one's own work, even if these materials are translations or adaptations, including work that is not original or exclusively that of the student;
- having communication devices (such as mobile phones, smart watches, etc.) accessible during theoretical-practical assessment tests (individual exams).
- talking with peers during individual partial or final exams;
- use or attempt to use any document related to the subject during the individual assessment exams, if these documents have not been explicitly allowed.

The final mark of each part will be the minimum value between 3.0 and the weighted average of the marks of the other activities of such part if the student has committed irregularities in an evaluation act (and, therefore, the passed by compensation will not be possible). In future editions of this subject, the student who has committed irregularities in an evaluation act will have to be re-assessed again.

In summary: copying, letting copy or plagiarizing (or attempting to) in any of the evaluation activities is equivalent to a FAIL, not compensable and without re-assessments.

To consult the academic regulations approved by the Governing Council of the UAB, please follow this link: https://www.uab.cat/doc/TR_Normativa_Academica_Plans_Nous

Assessment Activities

Title	Weighting	Hours	ECTS	Learning Outcomes
Exercise resolution in groups	20%	2	0.08	1, 2, 8, 4, 12, 10, 6
Individual Test with exercises, including theory and laboratory questions	25%	2	0.08	1, 2, 3, 8, 9, 12, 10, 6, 7
Individual Test with exercises, including theory and laboratory questions	25%	2	0.08	1, 2, 3, 8, 9, 12, 10, 6, 7
Laboratory exercises in groups	30%	2	0.08	1, 3, 8, 4, 12, 5, 10, 6, 11

Bibliography

- HENNESSY, John L. and PATTERSON, David, Computer Architecture: A Quantitative Approach. Morgan Kaufmann (Elsevier), 2018 (Chap. 1, 2 and 3)
- BRYANT, Randal and O'HALLARON David, Computer Systems: A Programmer's Perspective. Prentice Hall, 2011 (Chap. 5 and 6)
- BAKHVALOV, Denis, Performance Analysis and Tuning on Modern CPUs, easyperf.net, 2020
- PATTERSON, David and HENNESSY, John, Computer Organization and Design: The Hardware/Software Interface. Morgan Kauffman (Elsevier), 2009 (Chap. 4 and 5)

Software

Linux (operating system) + basic tools (editors, web browsers,...)

gcc/icc (compilers)

perf (application profiling tool)