Last update: 05-07-2019

340384 - PACO-I5O01 - Parallelism and Concurrence

Coordinating unit: 340 - EPSEVG - Vilanova i la Geltrú School of Engineering

Teaching unit: 701 - AC - Department of Computer Architecture

Academic year: 2019

Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2018). (Teaching unit Compulsory)
BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Compulsory)

ECTS credits: 6 Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: Eva Marín Tordera

Others: Eva Marín Tordera

Prior skills

It is advisable to have studied Computer Architecture and Operating Systems.

Degree competences to which the subject contributes

Specific:

1. CEFB5. Knowledge of informatic systems, its structure, function and interconnection, as well as fundamentals of its programming.
2. CEFC2. Ability to plan, conceive, develop, manage informatic projects, services and systems in all areas, leading their implementation and continuous improvement assassing their economic and social repercussions.
3. CEFC8. Ability to analyze, to design, to construct and to maintain applications in a well built, secure and efficient way choosing the most adequated paradigms and languages.
4. CEFC9. Ability to know, understand and assess computer structure and architecture, as well as basic components forming them.
5. CEFC14. Knowledge and application of fundamentals principals and basic techniques of parallel, concurrent, distributed and real time programming.

Teaching methodology

Theory classes are conducted using the resources available in the classroom (whiteboards, multimedia equipment) and are based on oral exposure by teachers of content on the subject under study (expository method). In some cases, there will be lectures based on the participation and involvement of students through short-term activities in the classroom, such as direct questioning, student presentations on specific topics or resolution of problems related to the theoretical exposed. Also the teacher will solve classroom exercises and propose collection exercises for students to prepare them independently. These exercises will be solved in class by the students individually or in groups of two people.
Small group classes are:
- Laboratory classes: be performed on school computer classrooms. The student must take practice prepared (read and understand the statement of the practice from a script that was previously found in digital campus), and sometimes if indicated shall make a preliminary report. The practices will be individual.

Learning objectives of the subject

The main objectives of this course are:

Last update: 05-07-2019

340384 - PACO-I5O01 - Parallelism and Concurrence

· Paradigms concurrency, parallel and distributed systems: (client-server, load sharing, tasks, etc.)..
· Platforms parallel (shared memory architectures, distributed memory).
· Tools to aid the development of parallel programs
· Programming and evaluation of parallel programs (programming models for different parallel platforms).
· Memory Coherence and consistency. Communication sincronització, race conditions, mutex, critical section, monitors, deadlock.

Study load

Total learning time: 150h	Hours large group:	45h	30.00%
	Hours medium group:	0h	0.00%
	Hours small group:	15h	10.00%
	Guided activities:	0h	0.00%
	Self study:	90h	60.00%

Last update: 05-07-2019

340384 - PACO-I5O01 - Parallelism and Concurrence

Content

1. Introduction to parallelism

Learning time: 9h

Theory classes: 1h
Practical classes: 2h
Self study : 6h

Description:
Need of parallelism. Parallelism versus concurrence. Problems using concurrence: deadlock, lifelock, starvation, fairness, data races

Related activities:
Activity 1. Unit 1 problems
Activity 2. Lab 0: Experimental setup, tools and programming model

2. Analysis of parallel applications

Learning time: 11h

Theory classes: 1h
Practical classes: 2h
Laboratory classes: 2h
Self study : 6h

Description:
Can a computation be divided into different parts? It is divided based on the tasks to do or based on the input/output data.
Will there be dependence of data between the tasks? How will they be solved? A good decomposition determines the parallel degree achievable.

Related activities:
Activity 1. Problems: Analysis of parrallel applications
Activiyt 2: Lab 0: Experimental setup, tools and programming model

340384 - PACO-I5O01 - Parallelism and Concurrence

3. Basics of parallel programing: Tasks decomposition

Learning time: 22h

Theory classes: 2h
Practical classes: 4h
Laboratory classes: 2h
Guided activities: 2h
Self study : 12h

Description:
Identification of concurrence patterns. Tasks decomposition , granularity and analysis of dependences. Identification of parallelism patterns: task parallelism versus divide and conquer. Mechanisms to implement the task decomposition: thread creation and destruction, thread synchronization patterns, exclusion when accessing share data.

Related activities:
Activity 1. Task parallelism problems
Activity 2. Lab 1: Embarrassingly parallelism with OpenMP: Mandelbrot set

4. Introduction to (shared-memory) Parallel Architectures

Learning time: 20h

Theory classes: 2h
Practical classes: 4h
Laboratory classes: 2h
Self study : 12h

Description:
Parallelism inside a processor (IDLP, DLP, TLP), multiprocessors with share memory, multiprocessors with distributed memory.

Related activities:
Activity 1. Unit 3 problems
Activity 2. Lab 0: Experimental setup, tools and programming model
Activity 4. Knowledge test

340384 - PACO-I5O01 - Parallelism and Concurrence

5. Programing with share memory	Learning time: 31h Theory classes: 3h Practical classes: 6h Laboratory classes: 2h Guided activities: 2h Self study : 18h

Description:
Parallel regions, threads and tasks. Task threads, barriers, mutual exclusion locks. Work distributors: loops, sections.

Related activities:
Activity 1. Share memory problems
Activity 2. Lab 2: Divide and Conquer parallelism with OpenMP: Sorting
Activity 3. Directed work. Aditional practise

6. Basics of parallel programing: Data decomposition	Learning time: 31h Theory classes: 3h Practical classes: 6h Laboratory classes: 4h Self study : 18h

Description:
Data decomposition (geometric versus recursive structure), data flow organization (regular versus irregular). Mechanisms to implement the data decomposition: creation and destruction process, process synchronization (barrier) and communications patterns (point-to-point communication, synchronous and asynchronous communication)

Related activities:
Activity 1. Data decomposition problems
Activity 2. Lab 3: Geometric decomposition: solving the heat equation

Qualification system

1st partial knowledge test *0,2+ problems * 0.1 + 0.3 * Laboratory + complementary * 0.1 Working + 2nd partial knowledge test * 0.3> = 5
The second partial exam can be a 2nd partial exam with a weight of 30% or a final exam with a weight of 50% in order to recover the first partial (the student chooses what to do). In this second case the formula is:
Problems * 0.1 + Laboratory * 0.3 + complementary work * 0.1 + Final knowledge test * 0.5> = 5
The 1st and 2nd partial knoeledge tests are reevaluable (or final test)

340384 - PACO-I5O01 - Parallelism and Concurrence

Regulations for carrying out activities

Activities 1, 2 and 4 are in person.
Activity 3 is non-attendance, although there may be a short presentation in class.
In the activities that take place in group the mark will be the same for all group members

Bibliography

Basic:

Mattson, Timothy G.; Sanders, Beverly A.; Massingill, Berna. Patterns for parallel programming. Boston [etc.]: Addison-Wesley, 2005. ISBN 0321228111.

Others resources:

It is recommended to use a laptop with a terminal

Computer material

Software a Boada

Connection and software in boada.ac.upc.edu