DEGREE CURRICULUM

CONCURRENT AND PARALLEL SYSTEMS

Coordination: VILADEGUT ABERT, MARC

Academic year 2021-22

CONCURRENT AND PARALLEL SYSTEMS 2021-22

Subject's general information

<table>
<tr><td>Subject name</td><td>CONCURRENT AND PARALLEL SYSTEMS</td></tr>
<tr><td>Code</td><td>102022</td></tr>
<tr><td>Semester</td><td>1st Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>3</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Double bachelor's degree: Degree in Computer Engineering and Degree in Business Administration and Management</td><td>4</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
<tr><td>Master's Degree in Informatics Engineering</td><td></td><td>COMPLEMENTARY TRAINING</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>6</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><th>PRALAB</th><th>TEORIA</th></tr>
<tr><td>Number of credits</td><td>3</td><td>3</td></tr>
<tr><td>Number of groups</td><td>2</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>VILADEGUT ABERT, MARC</td></tr>
<tr><td>Department</td><td>COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>6 ECTS = 25x6 = 150 hours
40% -> 60 classroom hours
60% -> 90 hours of autonomous student work</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Preferably in Spanish, in English if there are a foreign student.</td></tr>
</table>

CONCURRENT AND PARALLEL SYSTEMS 2021-22

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
ONRUBIA PALACIOS, JORDI RICARD	jordi.onrubia@udl.cat	6	
VILADEGUT ABERT, MARC	marc.viladegut@udl.cat	3	

Subject's extra information

The course is eminently practical, so we give more weight to the practical and programming. Basically we work with two languages, C programming for concurrent execution threads and Java to introduce high-level APIs for synchronization and concurrent patterns. None of these languages will be explained from scratch, because both C and Java have already seen in previous subjects of the degree. To continue the course is essential for students to have good fundamentals in C and Java programming. In the course it is assumed that students are able to design, develop and debug sequential applications of medium difficulty without much trouble.

Learning objectives

- To understand the importance of concurrent programming in actual applications.
- To identify the main characteristics of the different types of concurrent systems.
- To know and understand the problems derived from the development of concurrent programs that do not appear in sequential programming.
- To understand the concepts of synchronization and mutual exclusion between processes.
- To identify the security properties and vivacity that a concurrent system must meet and be able to reason if these properties are met.
- To gain experience and knowledge in communication and synchronization mechanisms that are used today to develop concurrent programs for both shared memory systems to distributed systems.
- To understand the operation of semaphores and monitors as synchronization mechanisms for shared memory; and understand how we can solve problems of concurrent programming using monitors.
- To apply methodologies of software engineering in the development of concurrent and parallel applications.

Competences

Cross-disciplinary Competences:

- EPS7. Capacity to work in situations with a lack of information and/or under pressure.

Specific Competences:

- GII-CRI11. Knowledge and application of the characteristics, functionalities and structure of the Distributed Systems, the Networks of Computers and Internet and design and implement applications based in them.
- GII-CRI14. Knowledge and application of the basic principles and basic techniques of the parallel, concurrent, distributed and of real time programming.
- GII-CRI16. Knowledge and application of the principles, methodologies and life cycle of the software engineering.

Subject contents

1. **Introduction to the concurrency**
    1. Concurrency definition

2. Need and benefits of concurrent programming
3. Concurrent hardware architectures
4. Features of concurrent systems
5. Concurrent programs Specification
    1. Conditions of Bernstein
    2. Concurrency Table
    3. Precedence Graphs
6. How to express concurrency
    1. Cobegin/Coend
    2. Fork/Join
    3. Unix + C
    4. Examples
7. Case study: Linux and Java threads

2. **Design concurrent and parallel applications**
    1. Model of concurrent/parallel programming
    2. Efficiency of concurrent/parallel programs
        1. Scope of parallelism
        2. Granularity
        3. Location
    3. Design of concurrent programs
        1. Stages of Design
        2. Task decomposition techniques
        3. Parallel design patterns
        4. Distribution and communication of tasks
    4. Case studies

3. **Synchronization of concurrent processes**
    1. Synchronization Introduction
        1. Race conditions
    2. The problem of mutual exclusión
        1. Software Solutions
        2. Hardware Solutions
    3. Conditional synchronization
        1. Classical problems of synchronization
    4. Case Study:
        1. Pthread mutex and condition variables
        2. Java Synchronization

4. **High-level APIs for concurrency and synchronization**
    1. Monitors
    2. Patterns concurrent
    3. Case Study:
        1. Boost Library
        2. Concurrent Java API

Methodology

Big-size Group: Theory Sessions (3 credits)

- Lecture: classes based on notes and transparencies where the concepts of the subject will be presented.
- Problems: The concepts of the subject will work through a series of exercises to be resolved collaboratively and help assimilate key concepts.
- Use Cases: It will apply the techniques seen in class to real examples and their impact on application

performance will be analyzed.

Mid-size group: Problems /Laboratory Sessions (3 credits)

- Tutorials and personalized monitoring by groups of practices.
- Laboratory: technologies and APIs concurrent programming will be presented and worked through tutorials and examples.
- Problems: Making and correcting exercises related to both the theoretical and practical part of the course.

Autonomous work:

- The homework exercises and practices will be completed outside of class time.

Development plan

W	Description	Classroom Activity GG	Classroom Activity GM Tuesday (Lab 1 & Lab2)	Autonomous work activity
1	Presentation Concurrency Introduction	Subject presentation T1: Concurrency Introduction	**L1: Posix Threads**	**Study literature and the program**
2	Concurrency Introduction	T1: Concurrency Introduction	**L1: Posix Threads** Practice 1: Presentation	T1: Concurrency Introduction
3	Concurrency Introduction	T1: Concurrency Introduction	**Tuesday Holiday (Lab1 & Lab 2)**	Practice 1 Problems: Concurrency
4	Design of concurrent applications	T1: Concurrency Introduction T2: Design of concurrent applications	**L2: Java Threads** Practice 2: Presentation	Practice 1 & 2 Problems: Concurrency
5	Design of concurrent applications	T2: Design of concurrent applications	**Tuesday Holiday (Lab1 & Lab 2)**	Practice 1 & 2 T2: Design
6	Design of concurrent applications	T2: Design of concurrent applications	Problems: Concurrency introduction P1 Cont. Assessment Problem	Practice 1 & 2 Problems: Design
7	Design of concurrent applications	T2: Design of concurrent applications	**P1 Cont. Asses. Correction** Problems: Design	Practice 1 & 2
8	Synchronization	T3: Synchronizing concurrent processes	Problems: Design Practice 1 & 2: Delivery	T3: Synchronization
9		**1er Partial**		Study
10	Synchronization	T3: Synchronizing concurrent processes	**L3: Linux synchronization** Practice 3: Presentation	T3: Synchronization
11	Synchronization	T3: Synchronizing concurrent processes	**L4: Java synchronization** Practice 4: Presentation	Practice 3 Problems: Synchronization
12	Synchronization	T3: Synchronizing concurrent processes	**L4: Java synchronization** P2 Cont. Assessment Problem	Practice 3 & 4 Problems: Synchronization
13	Synchronization	T3: Synchronizing concurrent processes	**Tuesday Holiday (Lab1 & Lab 2)**	Practice 3 & 4 Problems: Synchronization

14	Concurrency & synchronization APIs	T4: concurrency and synchronization High-level APIs	P2 Cont. Assessment Correction Practice 5: Presentation	Practice 3 & 4 T4: High-level APIs
15	Concurrency & synchronization APIs	**Thursday Holiday (Christmas)**	Practice 3 & 4: Delivery	Practice 5 T4: High-level APIs
16		**2nd Partial**		Study
17		**2nd Partial**		Study
18		TUTORIAS		
19		**Recovery**		Study

Evaluation

Table. Assessment Activities

Acr.	Assessment activity	Weighting	Minimum Grade	In Group	Mandatory	Recoverable
P1	*1st Partial Exam*	25%	4	NO	YES	YES
P2	*2nd Partial Exam*	25%	4	NO	YES	YES
PRA	*Practices*	40%	4	YES (<=2)	YES	1 PRA
PRB	*Problems*	10%	NO	YES (<=2)	NO	NO
There are minimum mark of 4 in the average of the written tests in order to have final note of the subject. It must be approved all practices individually. A practice is considered suspended if fails to reach 4.						
FinalGrade = 0,25*P1 + 0,25*P2 + 0,4*PRA + 0,1*PRB						

The course will be approved with a final grade at least of 5 and having completed all laboratory practices (each with at least 4 grade)

The final grade for the course is derived from the weighted sum of the grades of the 2 tests and the practical marks, plus the class participation and continuous assessment.

The course has two sets, each with a weight of 25% of the final grade. These tests are mandatory and the approved ones will not be revalue in the recovery exams. There are a minimum mark of 4 in the middle of the written tests to obtain the final grade for the course.

The completion and improvement of laboratory practices is mandatory to pass the course. The practices will be evaluated with a note that represents 40% of the final grade for the course. The copy of any of the practices of the course will involve suspend all practice and the subject. There is also a high note (4) to the labs. A practice is considered suspended if fails to arrive to mark of 4.During the recovery, only 1 of the 2 practices of the subject can be delivered and recovered.

There minimum mark of 4 in the middle of the written tests to obtain the final grade for the course.

Bibliography

Basic Bibliography:

- José Tomás Palma Méndez, M. C. Garrido Carrera, F. Sanchez Figueroa, A. Quesada Arencibia, "Programación Concurrente ", Thomson, 2003.
- Maurice Herlihy, Nir Shavit, "The Art of Multiprocessor Programming", Morgan Kaufmann, 2008.

- Douglas Lea, "Concurrent Programming in Java: Design Principles and Patterns", Addison-Wesley Professional, 2000.

Extended Bibliography:

- Gadi Taubenfeld, "Synchronization Algorithms and Concurrent Programming", Pearson / Prentice Hall, 2006
- M. Ben-Ari, "Principles of Concurrent and Distributed Programming", Addison-Wesley, 2nd Edition, 2006
- KayA. Robbins, Steven Robbins, "UNIX Programación Práctica. Guía para la Concurrencia, la Comunicación y los Multihilos", Edt.Prentice-Hall, 1997.