DEGREE CURRICULUM

CONCURRENT AND PARALLEL SYSTEMS

Coordination: CORES PRADO, FERNANDO

Academic year 2023-24

CONCURRENT AND PARALLEL SYSTEMS 2023-24

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><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>3</td></tr>
<tr><th>Number of groups</th><td>2</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>CORES PRADO, FERNANDO</td></tr>
<tr><td>Department</td><td>COMPUTER ENGINEERING AND DIGITAL DESIGN</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>
<tr><td>Distribution of credits</td><td>Fernando Cores 9</td></tr>
</table>

CONCURRENT AND PARALLEL SYSTEMS 2023-24

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
CORES PRADO, FERNANDO	fernando.cores@udl.cat	9	To be arranged by email

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: 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: Java Concurrent API
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
    4. Classical problems of synchronization
    5. Case Study:
        1. Monitors Concept
        2. Synchronization in Java.
4. **Concurrency and synchronization in C**
    1. Threads of Execution in Linux (Pthreads)
    2. Pthread mutex and condition variables

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 (Thursday)	Classroom Activity GM Tuesday (Lab 1 & Lab2)	Autonomous work activity
1	Presentation Concurrency Introduction	T1: Concurrency Introduction	Subject presentation **L1: Java Threads**	**Study literature and the program**
2	Concurrency Introduction	T1: Concurrency Introduction	**L1: Java Threads**	T1: Concurrency Introduction
3	Concurrency Introduction	**Thursday Holiday (Theory)**	**L1: Java Threads** Problems: Concurrent Programs	T1: Concurrency Introduction
4	Concurrency Introduction	T1: Concurrency Introduction	Practice 1: Presentation **L2: Java Concurrent API**	Practice 1 T1: Concurrency Introduction
5	Concurrency Introduction	**Thursday Holiday (Theory)**	**L2: Java Concurrent API**	Practice 1 T1: Concurrency Introduction
6	Design of concurrent applications	T2: Design of concurrent applications	Problems: Concurrency introduction P1 Cont. Assessment Problem	Practice 1 & 2 Problems: Concurrency
7	Design of concurrent applications	T2: Design of concurrent applications	**P1 Cont. Asses. Correction** Practice 2: Presentation	Practice 2
8	Design of concurrent applications	T2: Design of concurrent applications	Problems: Design	Practice 2 Problems: Design
9		**1er Partial**		Study
10	Synchronization	T3: Synchronizing concurrent processes	**L3: Java synchronization**	T3: Synchronization
11	Synchronization	T3: Synchronizing concurrent processes	**L3: Java synchronization** Practice 3: Presentation	Practice 3 T3: Synchronization
12	Synchronization	T3: Synchronizing concurrent processes	**L4: Pthreads Concurrency and Synchronization**	Practice 3 T3: Synchronization
13	Synchronization	**Thursday Holiday (Theory)**	**L4: Pthreads Concurrency and Synchronization**	Practice 3 T3: Synchronization
14	Synchronization Pthreads	T3: Synchronizing concurrent processes	Problems: Synchronization P2 Cont. Assessment Problem Practice 4: Presentation	Practice 4 T3: Synchronization Problems: Synchronization
15	Synchronization Pthreads	Problems: Synchronization	P2 Cont. Assessment Correction Problems: Synchronization	Practice 4 T3: Synchronization Problems: Synchronization
16		**2nd Partial**		Study
17		**2nd Partial**		Study
18		TUTORIAS		
19		**Recovery**		Study

Evaluation

ASSESSMENT

The course is passed with a final mark of 5 or more, obtaining an average mark of 4 in the exams and having completed the laboratory practices correctly.

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

Table. Assessment Activities

Block	Acr	Assessment activity	Weighting	Minimum Grade	In Group	Mandatory	Recoverable
Theory	*P1*	*1st Partial Exam*	20%	4	NO	YES	YES
	P2	*2nd Partial Exam*	20%	4	NO	YES	YES
PRA1	*PRA1*	*Practice 1*	10%	NO	YES (<=2)	YES	YES/NO
PRA2	*PRA2*	*Practice 2*	8%	NO	YES (<=2)	YES	YES/NO
PRA3	*PRA3*	*Practice 3*	10%	NO	YES (<=2)	YES	YES/NO
PRA4	*PRA4*	*Practice 4*	12%	NO	YES (<=2)	YES	YES/NO
Continuous Assessment	*EC*	*Cont. Ass. problems*	10%	NO	YES (<=2)	NO	NO
	EP	*Practical exercises*	10%	NO	NO	NO	NO
	PA	*Class Participation*	5%	NO	NO	NO	NO
There is a minimum mark of 4 in the average of the written exams to be able to pass the subject							
Final Grade = 0,20*(P1+P2) + 0,10*(PRA1+ PRA2+ PRA3+PRA4) + 0,10*(EC+EP) + 0,05*PA							

The subject has two partial exams, each of them with a weight of 20% in the final grade. These exams are compulsory and eliminate material. There is a minimum mark (4) for the theory grade (average of the two partial exams). If the weighting of the different marks is greater than or equal to 5, but the minimum grade of the theory exam has not been reached, then the subject will be considered failed with a 4.9.

The completion of laboratory practices is mandatory to pass the subject. The practices will be evaluated with a grade that will represent 40% of the final grade for the subject. Of the four practices of the subject, only two can be recovered in the second call. The practices recovered will have a 20% penalty in the mark obtained.

The copy of any of the practices of the subject will imply failing all practices and with it the subject.

ALTERNATIVE ASSESSMENT

In the case of requesting the alternative evaluation (by family or work conciliation), this will consist of carrying out a final exam, plus the delivery of the 4 practices of the subject. The requirements for these activities are the same as those applied in the normal assessment.

Table. Alternative Assessment Activities

Block	Acr	Assessment activity	Weighting	Minimum Grade	In Group	Mandatory	Recoverable
Theory	*EX*	*Final Exam*	50%	4	NO	YES	YES
PRA1	*PRA1*	*Practice 1*	12,5%	NO	NO	YES	YES/NO
PRA2	*PRA2*	*Practice 2*	12,5%	NO	NO	YES	YES/NO
PRA3	*PRA3*	*Practice 3*	12,5%	NO	NO	YES	YES/NO
PRA4	*PRA4*	*Practice 4*	12,5%	NO	NO	YES	YES/NO
There is a minimum mark of 4 in the average of the written tests to be able to pass the subject.							
Final Grade = 0,5*EX + 0,125*(PRA1+ PRA2+ PRA3+PRA4)							

The course is passed with a final grade greater than or equal to 5 and obtaining at least a 4 in the final exam and having completed the laboratory practices correctly.

The final grade for the subject is obtained from the weighted sum of the final exam grade and the practical marks.

There is a minimum mark (4) for the theory mark. If the weighting of the different marks is greater than or equal to 5, but the minimum grade of the theory exam has not been reached, then the subject will be considered failed with a 4.9.

The practices will be evaluated with a mark that will represent 50% of the final grade for the course. The practices will be delivered all together after taking the exam. Of the four practices of the subject, only two can be recovered in the second call. The practices recovered will have a 20% penalty in the mark obtained.

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.