270068 - PAP - Parallel Programming and Architectures

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2018
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: Catalan

Teaching staff
Coordinator: - Eduard Ayguadé Parra (eduard@ac.upc.edu)
Others: - Lluc Álvarez Martí (lluca@ac.upc.edu)

Prior skills
Defined by the pre-requisites for the course

Requirements
- Prerequisite PAR

Degree competences to which the subject contributes

Specific:
CEC2.1. To analyse, evaluate, select and configure hardware platforms for the development and execution of computer applications and services.
CEC2.2. To program taking into account the hardware architecture, using assembly language as well as high-level programming languages.
CT8.7. To control project versions and configurations.

General:
G8. APPROPRIATE ATTITUDE TOWARDS WORK: to have motivation to be professional and to face new challenges, have a width vision of the possibilities of the career in the field of informatics engineering. To feel motivated for the quality and the continuous improvement, and behave rigorously in the professional development. Capacity to adapt oneself to organizational or technological changes. Capacity to work in situations with information shortage and/or time and/or resources restrictions.

Teaching methodology
The theory lessons introduce the knowledge, techniques, and concepts using examples of real code or pseudo-code. These lessons will be complemented with the realization of problems in the practical lessons. The laboratory sessions put into practice the theoretical contents, and evaluate the behavior and performance of the solutions proposed.

The course assumes that part of the theoretical contents, or laboratory statements, will have to be developed by the student independently.

The course is mainly focused on cluster architectures, using the C programming language, the Pthreads library and the OpenMP and MPI programming models.

Learning objectives of the subject
1. Students must be able to write and understand parallel programs that make use of the low-level Pthreads interface.
2. Students should be able to implement the basic functionalities in a library supporting the execution of parallel applications on a shared-memory architecture.
3. Students must be able to understand the main components used to build a multiprocessor architecture, and design on paper a system that fulfill certain design restrictions.
4. Students must be able to write simple applications using the MPI programming model, evaluate their performance and identify the critical parts that limit scalability.
5. Students must be able to assess the quality of a proposed solution to a specific problem.
6. Students must be capable of completing or expanding knowledge independently and to perform a specific job even though the statement is incomplete or information relevant to the implementation is missing.

### Study load

<table>
<thead>
<tr>
<th>Total learning time: 150h</th>
<th>Theory classes: 15h</th>
<th>10.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Practical classes: 15h</td>
<td></td>
<td>10.00%</td>
</tr>
<tr>
<td>Laboratory classes: 30h</td>
<td></td>
<td>20.00%</td>
</tr>
<tr>
<td>Guided activities: 6h</td>
<td></td>
<td>4.00%</td>
</tr>
<tr>
<td>Self study: 84h</td>
<td></td>
<td>56.00%</td>
</tr>
</tbody>
</table>
270068 - PAP - Parallel Programming and Architectures

Content

Parallel programming using Pthreads

Degree competences to which the content contributes:
Description:
Introduction to the basic functionalities that are offered by the Pthreads low-level support library

Implementation of a shared-memory programming model: threads and synchronization, work sharing and tasking model

Degree competences to which the content contributes:
Description:
This topic presents how to design and implement a library supporting the execution of parallel programs in OpenMP, in particular the aspects related with thread management and synchronization, work sharing in the OpenMP worksharing constructs and the tasking model.

Components and design of a cluster architecture

Degree competences to which the content contributes:
Description:
This topic will introduce the main components in a cluster architecture we the objective of doing a design with certain performance/power trade-offs and budget.

MPI: parallel programming for distributed-memory architectures

Degree competences to which the content contributes:
Description:
This topic will introduce how to program parallel applications using MPI, a programming model based on message passing for distributed-memory cluster architectures.
### Planning of activities

<table>
<thead>
<tr>
<th>Activity</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>POSIX threads (Pthreads)</strong></td>
<td>12h</td>
</tr>
<tr>
<td>Theory classes: 3h&lt;br&gt;Practical classes: 3h&lt;br&gt;Laboratory classes: 0h&lt;br&gt;Guided activities: 0h&lt;br&gt;Self study: 6h</td>
<td></td>
</tr>
<tr>
<td><strong>Implementation of a shared-memory programming model</strong></td>
<td>54h</td>
</tr>
<tr>
<td>Theory classes: 4h&lt;br&gt;Practical classes: 4h&lt;br&gt;Laboratory classes: 16h&lt;br&gt;Guided activities: 0h&lt;br&gt;Self study: 30h</td>
<td></td>
</tr>
<tr>
<td><strong>Components and design of a cluster architecture</strong></td>
<td>38h</td>
</tr>
<tr>
<td>Theory classes: 4h&lt;br&gt;Practical classes: 4h&lt;br&gt;Laboratory classes: 6h&lt;br&gt;Guided activities: 4h&lt;br&gt;Self study: 20h</td>
<td></td>
</tr>
<tr>
<td><strong>Other parallel programming models: MPI</strong></td>
<td>36h</td>
</tr>
<tr>
<td>Theory classes: 4h&lt;br&gt;Practical classes: 4h&lt;br&gt;Laboratory classes: 8h&lt;br&gt;Guided activities: 0h&lt;br&gt;Self study: 20h</td>
<td></td>
</tr>
<tr>
<td><strong>Final Exam</strong></td>
<td>10h</td>
</tr>
<tr>
<td>Guided activities: 2h&lt;br&gt;Self study: 8h</td>
<td></td>
</tr>
</tbody>
</table>

**Specific objectives:**

- POSIX threads (Pthreads): 1
- Implementation of a shared-memory programming model: 1, 2, 5, 6
- Components and design of a cluster architecture: 3, 6
- Other parallel programming models: MPI: 4, 5, 6
Specific objectives:

1, 2, 3, 4

Qualification system

The grade for the course is calculated from 3 grades:
- Theoretical contents grade
- Laboratory grade
- Autonomy and motivation grade

The theoretical contents grade (T) is obtained from the delivery of problems in class and presentations of assignments. If a student fails the theoretical part (T < 5), he/she will have to take a final exam; in this case, the new theoretical contents grade T will be obtained as the maximum between the original grade T and the mark obtained in the final exam.

The laboratory grade (L) is obtained from the grades of the laboratory deliverables and monitoring of the laboratory sessions by the professor.

The grade of autonomy and motivation (A) evaluates the ability of students to face situations of lack of information and their motivation to explore additional topics or go beyond what is initially assigned. It is obtained from the results of those laboratory experiments that require the exploration of additional material and/or perform optional/free parts.

The final grade is calculated $F = T \times 0.5 + L \times 0.3 + A \times 0.2$.

Bibliography

Basic:


Others resources:

Hyperlink

https://computing.llnl.gov/tutorials/pthreads/

https://www.openmp.org/specifications/

https://computing.llnl.gov/tutorials/mpi/