270069 - PCA - Architecture-Aware Programming

Coordinating unit: 270 - FIB - Barcelona School of Informatics
Teaching unit: 701 - AC - Department of Computer Architecture
Academic year: 2016
Degree: BACHELOR'S DEGREE IN INFORMATICS ENGINEERING (Syllabus 2010). (Teaching unit Optional)
ECTS credits: 6
Teaching languages: Catalan, Spanish

Teaching staff

Coordinator: - Daniel Jimenez Gonzalez (djimenez@ac.upc.edu)

Prior skills

The requirements provided for.

Requirements

- Prerequisite AC

Degree competences to which the subject contributes

Specific:
CEC2.2. To program taking into account the hardware architecture, using assembly language as well as high-level programming languages.
CEC2.3. To develop and analyse software for systems based on microprocessors and its interfaces with users and other devices.
CEC3.1. To analyse, evaluate and select the most adequate hardware and software platform to support embedded and real-time applications.
CEC3.2. To develop specific processors and embedded systems; to develop and optimize the software of these systems.

General:
G1. ENTREPRENEURSHIP AND INNOVATION: to know and understand the organization of a company and the sciences which govern its activity; capacity to understand the labour rules and the relation between planning, industrial and business strategies, quality and benefit. To develop creativity, entrepreneur spirit and innovation tendency.
Teaching methodology

Theory classes designed to present basic concepts of architecture and processor optimization tools and methodology; inserting small problems to solve in class.

For each subject, a series of theoretical and practical exercises will be proposed to the students. Those exercises have to be solved and hanged in the forums created by the teacher at Athena. The problems to be solved in the forums will be of medium difficulty and designed to facilitate the laboratory activity, i.e., which also serve as preparatory work. We create teams of about 5 students. The number of people should be equal to the number of exercises per subject and team and will be fixed for the course. In principle, allocated a minimum of 5 problems per team and topic.

The methodology to solve these problems consists of a personal work and teamwork. Basically consist of: (1) individually solve problems and upload the problem solved in the forum created by the teacher for each team, (2) discuss in the forum of the individual solutions of the team (the first feedback received by the students) and agree a solution to every problem and (3) post a final solution to every problem in a common forum for final solutions to all teams. For each team, an student will be responsible for a problem, and that student will post the final solution of that problem to the common forum before a certain date. The teacher will review those final solutions (second feedback) but can also check the forums for discussion of individual solutions.

The classes of problems and team activities will be discussed with all the teams (third feedback). To allow students to compare their solutions with those of other students of the subject, the forums will be open to everyone. In addition, the course offers a website where students can send the solution of certain exercises of the course. The website will run programs, check their correctness and order the programs submitted by students considering the execution time.

The laboratory classes will support the theory and the exercises done in class of problems. Students will have the documentation before each practice session. The implementation of these practices will be entirely in the assigned lab hours. Practices will be divided into two groups: limited experiments to illustrate certain applications and optimizations, and others that will apply all knowledge to optimize a complete application. Students also have to work a complex practice that will mean in group two.

At the end of the last laboratory session of each topic, students will have time to deliver some of the exercises done in the laboratory.

Learning objectives of the subject

1. Follow a methodology of optimization to reduce the execution time of an application in a specific processor; first identifying the parts more expensive, then evaluating whether or not (Amdahl's law ) its optimization is worth, and finally perform the optimization if the performance gain is reasonable.
2. Identify and recognize the optimizations the compiler can do, either by the basic knowledge that the student has of the gcc compiler, the analysis of assembler code generated, or because the student does profiling or instrumentation.
3. Choose the tools and their options most suitable to generate the binary code, analyze the binary code generated and the execution of that code (by profiling, instrumentation or tracing); in addition to choosing metrics for this analysis and the environment more suitable for evaluation.
4. Define latency, repetition rate and throughput of an instruction; and apply this knowledge in optimizing of the codes to choose the most appropriate operation in each case.
5. Apply, in the proper way, optimization techniques to reduce and / or avoid long latency operations, reducing the execution time of the application.
6. Reduce the time allocated for breaks in code, using techniques that reduce the number of jumps.
7. Characterize the memory hierarchy of the computer with either a manual or studying it empirically with codes and tools for profiling and tracing.
8. Apply, in the proper way, optimization techniques to make more efficient use of processor memory hierarchy, reducing
the execution time of the application.

9. Identify if a code is vectorizable and vectorized it if it means reducing the execution time of the application.

10. Analyze and optimize a medium sized application, applying the optimization techniques and methodology followed in the course, in coordination with the rest of their team members.

11. Lead a group of students to obtain the best solution to an optimization problem or exercise of analysis given, considering the solutions of the rest of his team members and after a discussion among all team members.

### Study load

| Total learning time: 150h | Theory classes: 15h 10.00% | Practical classes: 15h 10.00% | Laboratory classes: 30h 20.00% | Guided activities: 6h 4.00% | Self study: 84h 56.00% |
# Introduction: Architecture-Aware Programming

**Degree competences to which the content contributes:**

**Description:**
What we mean by Architecture-Aware Programming and in what cases it is worth it. Optimization strategies. Performance evaluation.

## Programming and Optimization Tools

**Degree competences to which the content contributes:**

**Description:**
Performance measurement programs. Compilation options. Study of dynamic behavior of programs.

## Long latency operations

**Degree competences to which the content contributes:**

**Description:**
Alternatives to run some high cost operations and/or long latency. Replacement of complex operations for simple operations. Memoization of results. Detection of trivial cases. Use of operating system resources.

## Control flow optimization

**Degree competences to which the content contributes:**

**Description:**
Detection and elimination of critical breaks. Inlining. Loop Unrolling, Collapsing, Fusion, etc.

## Memory Conscious Optimizations

**Degree competences to which the content contributes:**

**Description:**
Size and alignment of data. Division of data into blocks of size proportional to the cache (blocking). Memory disambiguation.

## Vector Extensions and use of custom hardware.

**Degree competences to which the content contributes:**
Description:
SIMD extensions. Identification of codes that can be vectorized. Inserting source code vector in C. Use of tools of High Level Synthesis to accelerate (optimize) in hardware the most time consuming parts of the program, on ARM-based SoC.
### Planning of activities

<table>
<thead>
<tr>
<th>Introduction to Programming conscious architecture</th>
<th>Hours: 3h 30m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 1h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 1h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 0h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 1h 30m</td>
</tr>
</tbody>
</table>

**Description:**
Actively participate in a session-class exhibition participatory 2 hours of theory and problems. (2 hours). Studying at home and do the exercises assigned topic to be proposed as an individual homework.

**Specific objectives:**
1

<table>
<thead>
<tr>
<th>Scheduling and Optimization Tools</th>
<th>Hours: 16h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 1h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 4h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 9h</td>
</tr>
</tbody>
</table>

**Description:**
Actively participate in a session-class exhibition participatory 2 hours of theory and problems. (2 hours). Studying at home and do the exercises assigned topic to be proposed as an independent work.

**Specific objectives:**
2, 3

<table>
<thead>
<tr>
<th>Long latency operations</th>
<th>Hours: 16h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory classes: 2h</td>
</tr>
<tr>
<td></td>
<td>Practical classes: 2h</td>
</tr>
<tr>
<td></td>
<td>Laboratory classes: 3h</td>
</tr>
<tr>
<td></td>
<td>Guided activities: 0h</td>
</tr>
<tr>
<td></td>
<td>Self study: 9h</td>
</tr>
</tbody>
</table>

**Description:**
Participar activament en una sessió de classe expositiva-participativa de 2 hores de teoria i problemes. (2 hores). Estudiar a casa el tema assignat i realitzar els exercicis que s’hagin proposat com a treball autònom.

**Specific objectives:**
1, 2, 4, 5, 11
## Optimization flow control

**Hours:** 24h  
Theory classes: 2h  
Practical classes: 2h  
Laboratory classes: 6h  
Guided activities: 0h  
Self study: 14h

**Description:**  
Actively participate in a session-class exhibition participatory 2 hours of theory and problems. (2 hours). Studying at home and do the exercises assigned topic to be proposed as an independent work.

**Specific objectives:**  
1, 2, 3, 6, 11

## Control 1

**Hours:** 3h  
Guided activities: 1h  
Self study: 2h

**Description:**  
Issues 1-4.

**Specific objectives:**  
1, 2, 3, 4, 5, 7

## Optimizations conscious memory

**Hours:** 21h 30m  
Theory classes: 2h  
Practical classes: 3h  
Laboratory classes: 4h  
Guided activities: 0h  
Self study: 12h 30m

**Description:**  
Actively participate in a session-class exhibition participatory 2 hours of theory and problems. (2 hours). Studying at home and do the exercises assigned topic to be proposed as an independent work.

**Specific objectives:**  
1, 2, 3, 7, 8, 11

## Vector Extensions and custom hardware usage

**Hours:** 31h  
Theory classes: 3h  
Practical classes: 4h  
Laboratory classes: 6h  
Guided activities: 0h  
Self study: 18h
Description:
Actively participate in a session-class exhibition participatory 2 hours of theory and problems. (2 hours). Studying at home and do the exercises assigned topic to be proposed as an independent work.

Specific objectives:
1, 2, 3, 9, 11

Revision Theory & Problems

Description:
Review of theoretical concepts and solve problems.

Specific objectives:
4, 5, 6, 7, 8, 9

Final Project

Description:
The group of students must present the optimization performed for final project.

Specific objectives:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Resolution of questions about practice and Oral and Public Presentation of Final Results of the final project

Description:
The group of students must present the optimization performed for final project.

Specific objectives:
1, 2, 3, 5, 6, 8, 9, 10
### Control 2

**Description:**
All subjects. More centered on 5-6, but any previous concept may be evaluated.

**Specific objectives:**
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

**Hours:**
- Guided activities: 1h
- Self study: 2h

### Final Exam

**Description:**
Final Exam. Outside of class time.

**Specific objectives:**
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

**Hours:**
- Guided activities: 3h
- Self study: 0h
Qualification system

In the evaluation of the student involved five components:

(1) Factor Monitoring:

Over the course, students must solve a series of exercises (exercises) from the total number of laboratory exercises and the collection of problems. There are two types of exercises: laboratory exercises (individual work) and class exercises (cooperative work).

- Laboratory exercises:
  The solution of lab exercises will be a brief report (between 1 and 3 pages) that must be delivered to the teacher via the Raco to assess whether the report is satisfactory.

- Exercises class of problems:
  These exercises are requested by the teacher in class of problems and must work in teams and presented to Athena Forum or via Raco.

The successful completion of at least 75% of the exercises of the lab and required problems is a required condition to pass the course without having to go to the final exam. These exercises help to trace a factor (fs) to the exams mark.

If the teacher evaluates the student has worked all the exercises and laboratory exercises the factor monitoring will be:

\[ fs = \max (0.8 \times \left( \frac{\text{exercises\_proposed\_and\_satisfactorily\_evaluated}}{\text{exercises\_proposed}} \right) - 0.75, 0) + 1.0 \]

Otherwise, be 1.0 fs.

(2) Partial control mark:

During the course students must answer two controls: C1, C2. Its objective is to assess whether the student has achieved assimilate code optimizations explained so far.

\[ C = 0.35 \times C1 + 0.65 \times C2 \]

(3) Practice Final mark:

En: Presentation of a report on the practice final. The practice will end a code that the students have applied to optimize the techniques learned during the course.

(4) Final Exam Mark:

Final: Final examination will include theoretical and practical cases.

(5) Extra Challenge Point:

Challenge: We propose an optimization exercise delivered via a web portal or via Raco that will be awarded function of the final ranking position. The student with the best solution's final grade will be increased by 1 point (faster solution), 0.9 points (second quickest solution), ..., and 0.1 points (tenth quickest solution) respectively. These extra points will be only applied to those student that has AC>=5.0 and En>=5.0.
The course can be approved in two ways:

The grade exam (NEX) can be achieved in two ways, either by continuous assessment or final examination:

If ((C >= 5) and (exercises_proposed_and_satisfactorily_evaluated >= 75%))
   NEX = MAX (C, Final)
otherwise:
   NEX = MAX(0.25*C+0.75*Final, Final)

The final grade is:

Final Mark = MIN (0.60 * NEX * fs + 0.40*En+ Challenge 10)

The competition evaluates each student considering cross their teamwork, leadership in finding effective solutions, alternative, creative and innovative to the problems to be solved which the student is responsible. That is, it will be evaluated their ability to find a solution to be a successful combination of the solutions found by a problem or the solution. Also part of this assessment to participate in finding creative solutions to problems that can not be responsible.

Bibliography

Basic: