NOVA FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

Course Administrivia

lecture 01 (2025-03-10)

Master in Computer Science and Engineering

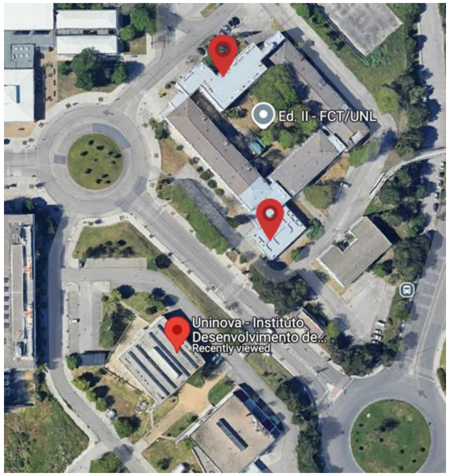
— Concurrency and Parallelism / 2024-25 —

João Lourenço <joao.lourenco@fct.unl.pt>

Basic Info

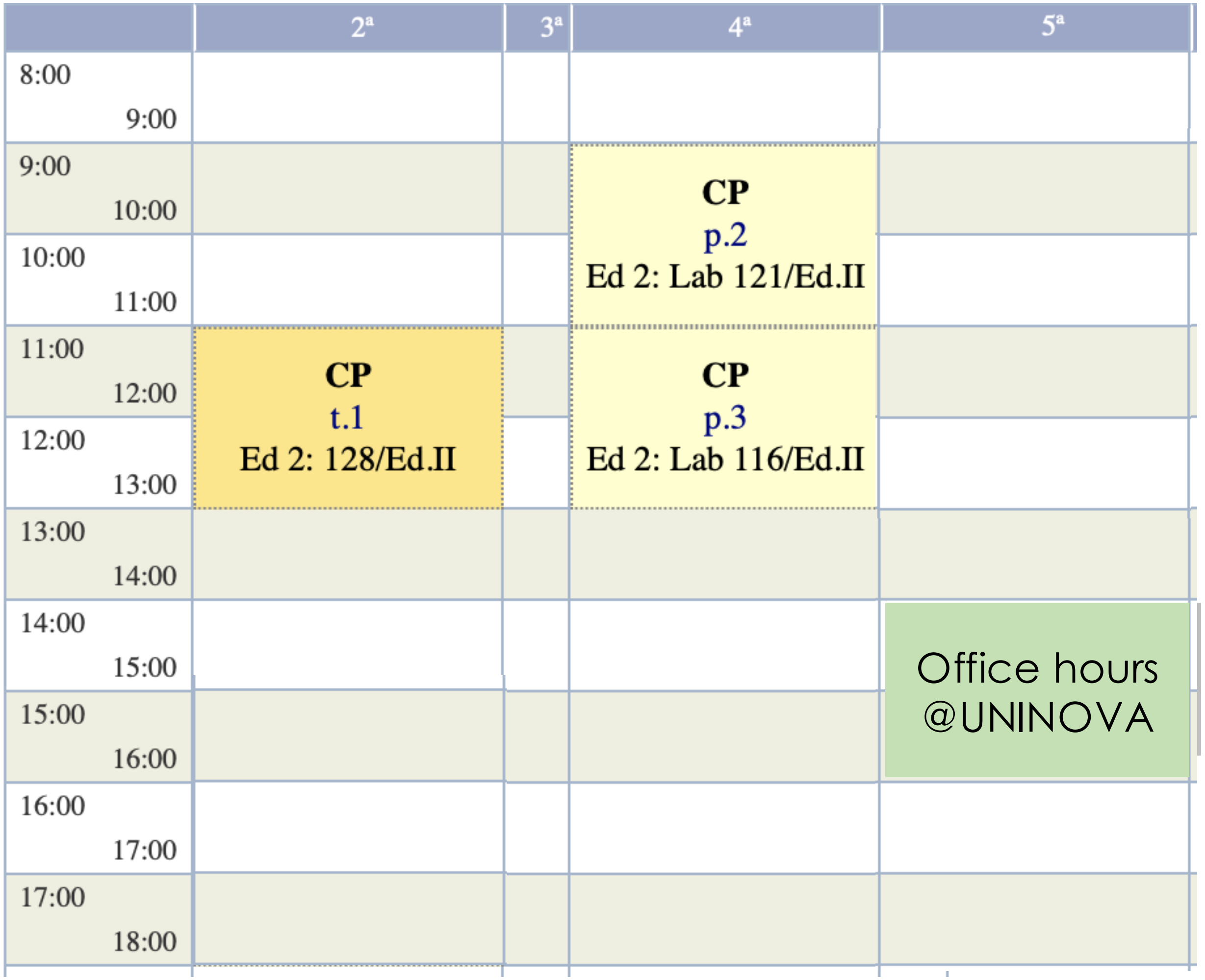
- Lectures and Labs
  - João Lourenço <joao.lourenco@fct.unl.pt>

- Office location @ CS Dept
  - **Office:** Dep. Informática • Building II • Room P2/9
  - **Extension:** 10740



- Office location @UNINOVA
  - **Office:** Building 1 • Room 1.1.1
  - **Office hours:** Thursdays 14:00 – 16:00

Schedule

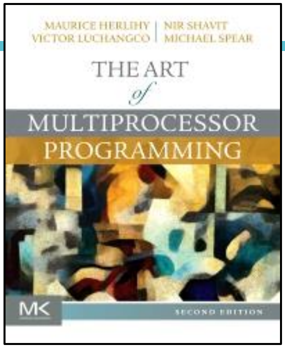


	2ª	3ª	4ª	5ª
8:00 9:00				
9:00 10:00			**CP** p.2 Ed 2: Lab 121/Ed.II	
10:00 11:00				
11:00 12:00	**CP** t.1 Ed 2: 128/Ed.II		**CP** p.3 Ed 2: Lab 116/Ed.II	
12:00 13:00				
13:00 14:00				
14:00 15:00				Office hours @UNINOVA
15:00 16:00				
16:00 17:00				
17:00 18:00				

(*) Remote office hours by appoitment!

Main Bibliography

- Herlihy M., Shavit N., Luchangco V., Spear M.;
  The Art of Multiprocessor Programming;
  Morgan Kaufmann (2020); ISBN: 978-0-12-415950-1

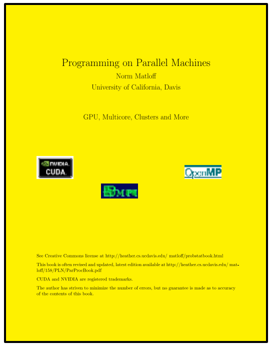


MAURICE HERLIHY | NIR SHAVIT
VICTOR LUCHANGCO | MICHAEL SPEAR

THE ART of MULTIPROCESSOR PROGRAMMING

MK SECOND EDITION

- Matloff N.;
  Programming on Parallel Machines;
  http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf



Programming on Parallel Machines
Norm Matloff
University of California, Davis

GPU, Multicore, Clusters and More

NVIDIA CUDA

OpenMP

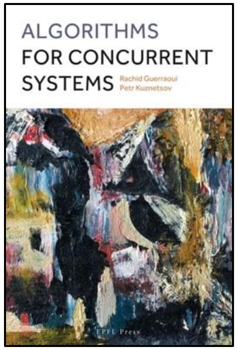
See Creative Commons license at http://heather.cs.ucdavis.edu/ matloff/probstatbook.html

This book is often revised and updated, latest edition available at http://heather.cs.ucdavis.edu/ matloff/158/PLN/ParProcBook.pdf

CUDA and NVIDIA are registered trademarks.

The author has striven to minimize the number of errors, but no guarantee is made as to accuracy of the contents of this book.

- Guerraoui R., Kuznetsov P.;
  Algorithms for Concurrent Systems;
  EPFL Press English Imprint(2013); ISBN: 978-2-88-915283-4

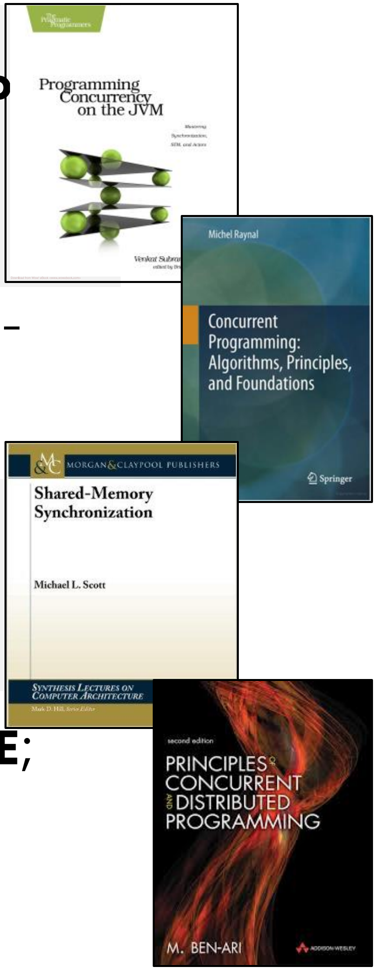


ALGORITHMS FOR CONCURRENT SYSTEMS
Rachid Guerraoui
Petr Kuznetsov

EPFL Press

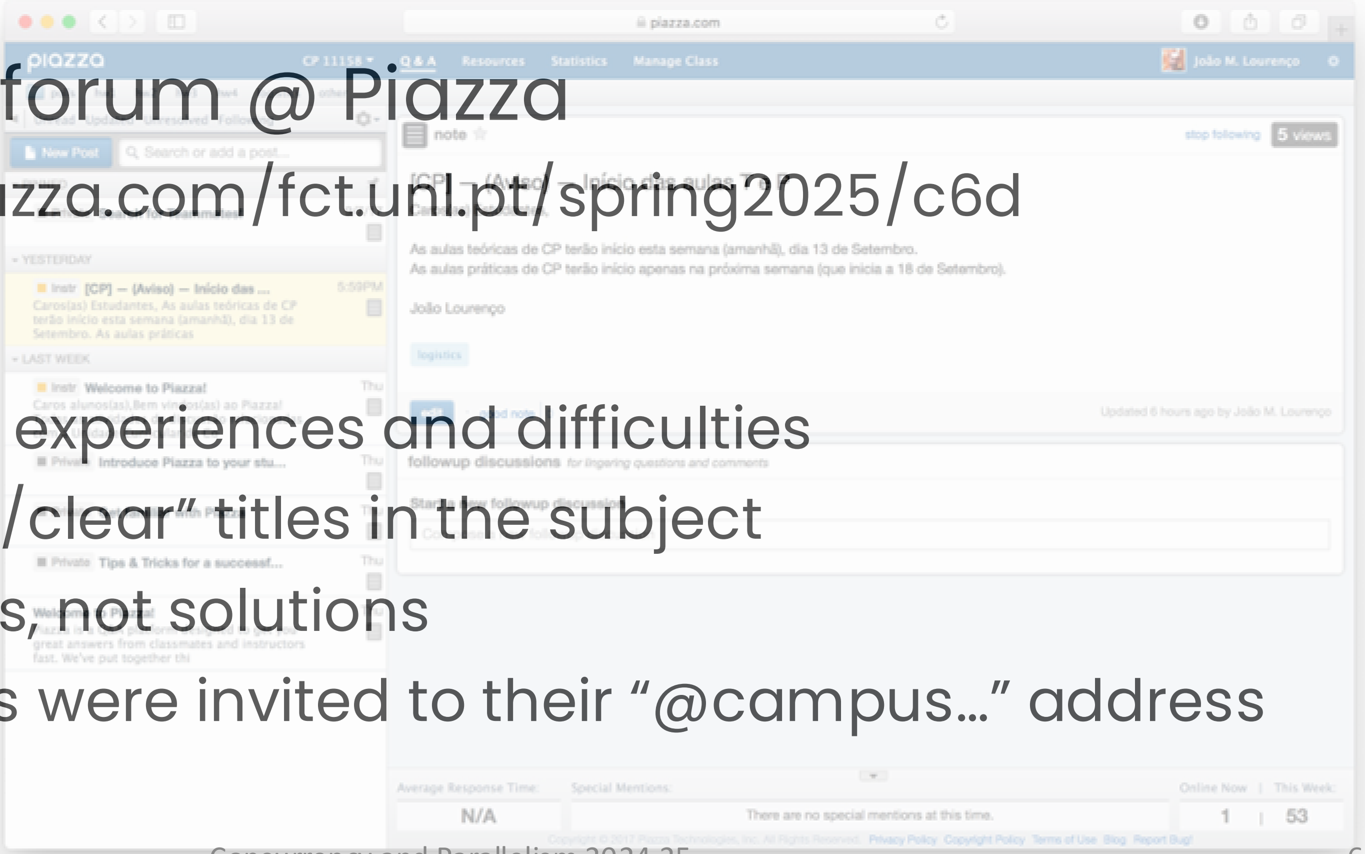
Additional Bibliography

- Suhramaniam V.;
  Programming Concurrency on the JVM: Mastering Synchronizatio STM, and Actors;
  The Pragmatic Bookshelf (2011); ISBN-13: 978-1-934356-76-0
- Raynal M.;
  Concurrent Programming: Algorithms, Principles, and Foundations; Springer-Verlag Berlin Heidelberg (2013); ISBN: 978-3-642-32026-2
- Michael L. S.;
  Shared-Memory Synchronization;
  Morgan & Claypool (2013); ISBN: 978-1-608-45956-8
- Ben-Ari M.;
  Principles of Concurrent and Distributed Programming, 2/E;
  Pearson (2006); ISBN: 978-0-321-31283-9


Programming Concurrency on the JVM
Michel Raynal
Concurrent Programming: Algorithms, Principles, and Foundations
Springer
MORGAN & CLAYPOOL PUBLISHERS
Shared-Memory Synchronization
Michael L. Scott
SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE
second edition
PRINCIPLES OF CONCURRENT AND DISTRIBUTED PROGRAMMING
M. BEN-ARI

Additional Information

- Class web page @ CLIP
  - All assignments, handouts, [lecture notes]
- Discussion forum @ Piazza
  - https://piazza.com/fct.unl.pt/spring2025/c6d
- Rules
  - Share your experiences and difficulties
  - Use “smart/clear” titles in the subject
  - Share ideas, not solutions
  - All students were invited to their “@campus…” address



Private Search for Teammates!

Instr [CP] — (Aviso) — Início das ... 5:59PM
Caros(as) Estudantes, As aulas teóricas de CP terão início esta semana (amanhã), dia 13 de Setembro. As aulas práticas

Instr Welcome to Piazza! Thu
Caros alunos(as),Bem vindos(as) ao Piazza!

Private Introduce Piazza to your stu... Thu

Private Get familiar with Piazza Thu

Private Tips & Tricks for a successf... Thu

Welcome to Piazza! Thu
Piazza is a Q&A platform designed to get you great answers from classmates and instructors fast. We've put together thi

[CP] — (Aviso) — Início das aulas T e P

Caros(as) Estudantes,

As aulas teóricas de CP terão início esta semana (amanhã), dia 13 de Setembro.
As aulas práticas de CP terão início apenas na próxima semana (que inicia a 18 de Setembro).

João Lourenço

Updated 6 hours ago by João M. Lourenço

followup discussions for lingering questions and comments

Start a new followup discussion

Compose a new followup discussion

Average Response Time: N/A

Special Mentions: There are no special mentions at this time.

Online Now | This Week: 1 | 53

Copyright © 2017 Piazza Technologies, Inc. All Rights Reserved. Privacy Policy Copyright Policy Terms of Use Blog Report Bug!

Course Goals: Knowledge

- To understand the concepts of **concurrency** and **parallelism**, and how they can be explored when designing software;
- To identify the **models** used for problem solving in multiprocessor systems;
- To know the **paradigms** used to develop algorithms on multiprocessor systems;
- To know the **languages, libraries and tools** used in the development of concurrent programs;
- To understand the **correctness properties** of concurrent systems;
- Be able to **evaluate the use of synchronization primitives** used in concurrent data structures;
- Be familiar with **common concurrency problems**, and **how to mitigate or avoid them**.

Course Goals: Application

- Be able to **identify and exploit opportunities for concurrency** within a software system;
- Be able to **partition a problem** into multiple tasks to be executed in a concurrent system;
- Be able to **reason about the behavior** of concurrent programs;
- Be able to **build correct and efficient** concurrent algorithms;
- Be able to **use the Java/C-like programming languages and libraries** to develop concurrent software systems;
- Be able to **use programming tools** in the development of concurrent applications, including the design, implementation, debugging and deployment stages;
- Be able to **predict, measure, and evaluate** the performance characteristics of a parallel system.

Syllabus

1. **Introduction to concurrency**
   and its challenges.
2. **Mutual exclusion:**
   Time; Critical Sections; Locks; Fairness.
3. **Concurrent objects:**
   Correctness; Progress; Sequential Objects; Quiescent and sequential Consistency; Linearizability.
4. **Foundations of shared memory:**
   Registers; Register Constructions; Atomic Snapshots.
5. **Primitive synchronization operations:**
   Monitors and Conditions; Spin-Locks, Readers–Writers Locks; Semaphores.
6. **Universality of consensus:**
   Universality; Lock-Free Universal Constructions.
7. **Spin locks and contention:**
   Test-And-Set Locks Spin Locks; Exponential Backoff; Queue and hierarchical Locks.
8. **Lock-free data structures:**
   Lists; Queues; the ABA Problem.
9. **Transaction memory:**
   Transactions and Atomicity; Software TM; Hardware TM.
10. **Work management:**
    Parallelism; Multiprocessor scheduling; Work distribution; Futures; Work-stealing dequeues.
11. **Concurrency without shared data:**
    Message passing, Actors, and Active objects.

Lab classes

- In the class
  - Design and implement concurrent and parallel (multiprocessor) programs
- One Homework / Project
  - Addressing concurrency and parallelism
- Rules for grouping
  - Group members may be enrolled in different lab classes
  - Groups of 3 students
    - ****All exceptions**** require explicit authorization
    - Non-authorized individual projects ****will not**** be graded
  - Group registration until March 21 at https://docs.google.com/spreadsheets/d/1W_iMDxqgge5ewv8tQuQp14IOLe63sVFtb7tA-_2ZqEQ/edit?usp=sharing

Evaluation

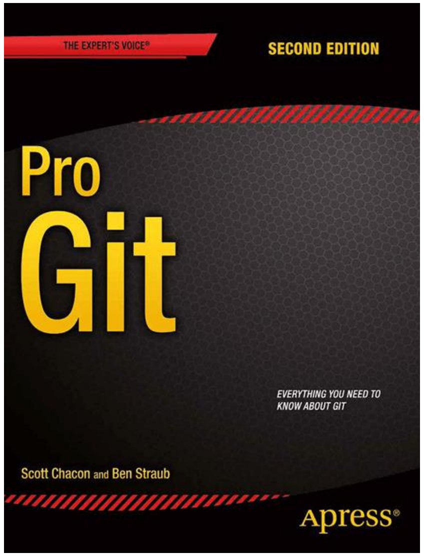
- [80%] two tests (individual, online)
  [average ≥ 8.5 points]
- [20%] one HW/project (groups of 3 students)
  [grade ≥ 8.5 points]
- [2.5%] participation in class' life cycle
  (includes lectures, labs, piazza, etc)
  (please notice that "**participation ≠ being there**")

The tests and exam will contain questions about the lab exercises and home project

Project development

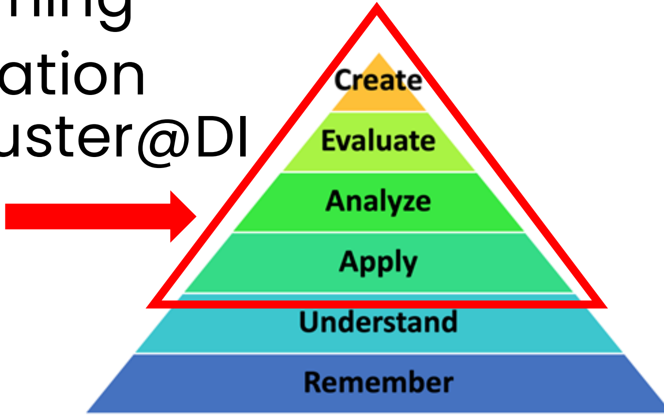
- We will use GIT extensively
- One **private** git repository per group.
  - Rep name: cp_2024_25_Gnn (where “nn” is the group number)
- **Each group member** will **commit regularly** his/her individual contributions to the group repository
  - Commit logs/messages must clearly state the contributions
- Individual project grade will consider individual contributions to the GIT repository
- *Project submission is just a Commit ID*
- ***Learn GIT now!!!!***

https://git-scm.com/book/en/v2


THE EXPERT'S VOICE®
SECOND EDITION
Pro
Git
EVERYTHING YOU NEED TO KNOW ABOUT GIT
Scott Chacon and Ben Straub
Apress®

Project schedule

- Assignment will be published by mid April
- It will comprise four phases
  - Shared memory multicore programming
  - Performance assessment and evaluation w/ multicore computers from the Cluster@DI
  - Writing a (small but relevant) report
  - Peer review of the submitted reports



- Final submission by the mid/end of May
- Discussions (random) in early June

Project report

- I don’t care who does what in the project, as long as everybody does technically relevant / meaningful work for the project
- **Work division** (what and percentage) **must be reported** in the project report
  - Must be supported by the individual commit logs

Any attempt of fraud => all groups' members will fail the course immediately

Project methodology

- Feel free to ask questions in/out classes
  - Teacher, colleagues, Piazza
    - *Please make use of Piazza!*
- Fell free to answer questions from colleagues
  - Helping finding a solution ≠ giving the solution for free

- Cite any source that inspired your work
  - If you cite what/who you used, then it is not cheating
  - Worst case I will deduce some points if it undermines the assignment

Project evaluat.

- Project report
  - If possible/feasible
    - will be graded using peer review
  - Otherwise
    - will be graded by me

- Project's code
  - Will be graded by me

What about the use of AI-based tools?

- Permitted for code that is not directly related with the subject under evaluation
  - Specifically, cannot be used in the domains of code synchronization and parallelization
- Any use must be explicitly reported in the project's report

Remember…

- Clip is the official source of information for the course.
- Confirm @Clip all the administrivia related topics.
  - **In case of contradiction, the information in Clip prevails**
- If yours is a special case where the rules are unclear or do not apply, please let me know (*so that we can handle it appropriately*)!

The END