NOVA SCHOOL OF SCIENCE & TECHNOLOGY
COMPUTER SCIENCE DEPARTMENT

Course Administrivia

lecture 01 (2021-03-15)

Master in Computer Science and Engineering

— Concurrency and Parallelism / 2020-21 —

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

Administrivia — Basic Info

- Lectures
  - João Lourenço <joao.lourenco@fct.unl.pt>
- Labs
  - João Lourenço <joao.lourenco@fct.unl.pt>
- Office location
  - Dep. Informática · Building II · Room P2/9
  - Extension: 10740

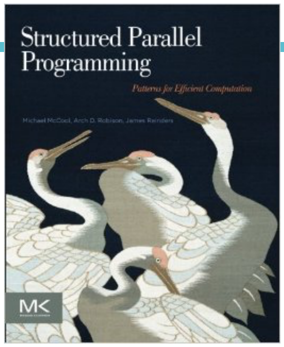
Administrivia — Schedule

	2ª	3ª	4ª	5ª	6ª
8:00 9:00					
9:00 10:00	**CP** to.1 não-presencial/Online		**CP** po.1 não-presencial/Online	**CP** po.2 não-presencial/Online	
10:00 11:00					
11:00 12:00	Office hours (*)		Office hours (*)	**CP** po.3 não-presencial/Online	
12:00 13:00					
13:00 14:00					
14:00 15:00					
15:00 16:00	Office hours (*)				
16:00 17:00				**CP** po.4 não-presencial/Online	
17:00 18:00					

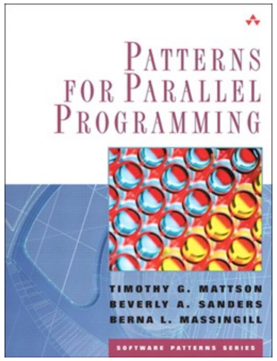
(*) Office hours by appoitment!

Administrivia — Main Bib.

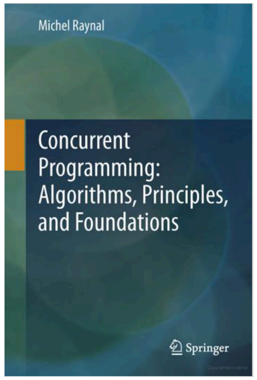
- McCool M., Arch M., Reinders J.; **Structured Parallel Programming: Patterns for Efficient Computation**; Morgan Kaufmann (2012); ISBN: 978-0-12-415993-8



- Mattson T., Sanders B., Massingill B.; **Patterns for Parallel Programming**; Addison-Wesley(2004); ISBN: 0-321-22811-1

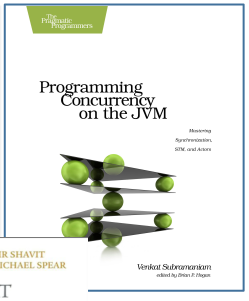


- Raynal M.; **Concurrent Programming: Algorithms, Principles, and Foundations**; Springer-Verlag Berlin Heidelberg (2013); ISBN: 978-3-642-32026-2

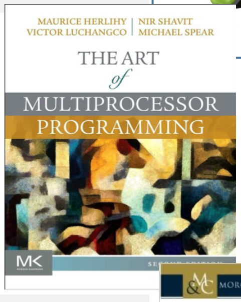


Administrivia — Additional Bib.

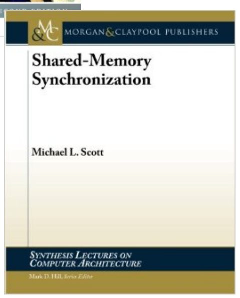
- Suhramaniam V.; **Programming Concurrency on the JVM: Mastering Synchronization, STM, and Actors**; The Pragmatic Bookshelf (2011); ISBN-13: 978-1-934356-76-0


The Pragmatic Programmers
Programming Concurrency on the JVM
Mastering Synchronization, STM, and Actors
Venkat Subramaniam
edited by Brian P. Hogan

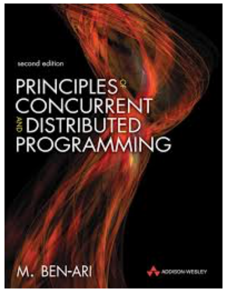
- Herlihy M., Shavit N., Luchangco V., Spear M.; **The Art of Multiprocessor Programming** (2nd ed); Morgan Kauffman (2021). ISBN: 978-0-12-415950-1


MAURICE HERLIHY | NIR SHAVIT
VICTOR LUCHANGCO | MICHAEL SPEAR
THE ART of MULTIPROCESSOR PROGRAMMING
MK

- Michael L. S.; **Shared-Memory Synchronization**; Morgan & Claypool (2013); ISBN: 978-1-608-45956-8

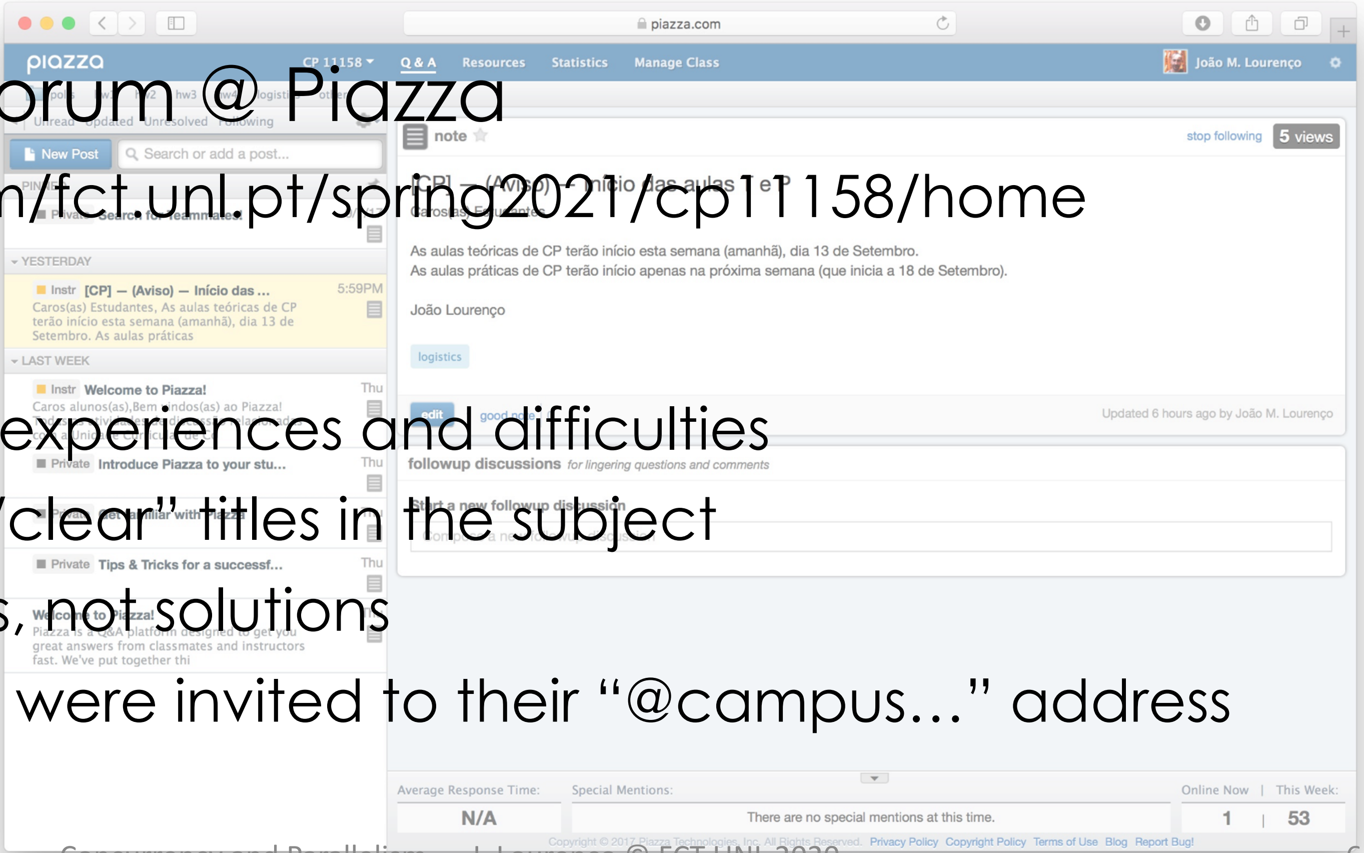

MORGAN & CLAYPOOL PUBLISHERS
Shared-Memory Synchronization
Michael L. Scott
Synthesis Lectures on Computer Architecture
Mark D. Hill, Series Editor

- Ben-Ari M.; **Principles of Concurrent and Distributed Programming, 2/E**; Pearson (2006); ISBN: 978-0-321-31283-9



Administrivia — Additional Bib.

- Class web page @ CLIP
  - All assignments, handouts, [lecture notes]
- Discussion forum @ Piazza
  - piazza.com/fct.unl.pt/spring2021/cp11158/home
- 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!

YESTERDAY

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

LAST WEEK

Instr Welcome to Piazza! Thu
Caros alunos(as),Bem vindos(as) ao Piazza! Todas as atividades de discussão relacionadas com a Unidade Curricular de Co

Private Introduce Piazza to your stu… Thu

Private Get familiar with Piazza Thu

Private Tips & Tricks for a successf… Thu

Welcome to Piazza!
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

Average Response Time: N/A

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

Administrivia — 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 multiprocessors and highly-parallel systems;
- To know the paradigms used in the development of algorithms for multiprocessors and highly-parallel systems;
- To know the languages, libraries and tools used in the development of concurrent and parallel programs;
- Be familiar with common concurrency problems, and how to mitigate or avoid them.

Administrivia — Course Goals: Application

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

Syllabus: Concurrency

1. **Parallel architectures**
   Flynn's taxonomy; performance theory (including Amdahl's and Gustafson's laws).
2. **Parallel programming**
   The spectrum of high-demanding computational problems; regular and irregular problems; strategies for problem decomposition and their mapping to programming patterns; the transactional and map-reduce models.
3. **Concurrency control and synchronization**
   Competition and collaboration; atomicity; linearization; monitors; locks; semaphores; barriers; producer-consumer; multi-reader single-writer locks; futures; concurrency in practice in Java and C.
4. **Safety and liveness**
   Safety vs. liveness; progress; deadlock; deadlock prevention, avoidance, detection, and recovery; livelock; livelock avoidance; priority inversion; priority inheritance. Lock-free algorithms.
5. **The transactional model**
   Composite operations; transactions (serializability), optimistic concurrency control (OCC) and transactional memory.
6. **Concurrency without shared data**
   Active objects; message passing; actors.

Lab classes

- In the class
  - Design and implement parallel and concurrent programs
- One Homework / Project
  - Addressing parallelism and/or concurrency
- 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

Administrivia — Evaluation

- [60%] two tests (individual, online)
  [average ≥ 8.5 points]
- [40%] one HW/project (groups of 3 students)
  [grade ≥ 8.5 points]
- [3%] participation in class' life cycle
  (includes lectures, labs, piazza, etc)
  (please note that "participation ≠ being there")

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

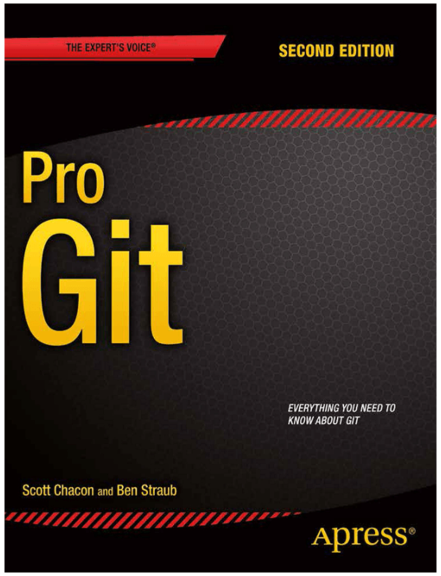
Administrivia — "Frequency"

- "Frequency":
  - Project ≥ 8.5 points => FREQUENCY ACQUIRED
- Frequency from 2019-20?
  - Your lab grade from 2019-20 will be considered.
  - Please note that tests will include questions about:
    - the lab exercises — ***you MUST answer these questions!***
    - this year's project — ***you MUST NOT answer these questions!***
- Frequency from 2018-19 or before?
  - You must acquire frequency again this year!

Administrivia — Project devel.

- We will use GIT extensively
- One **private** git repository per group.
  - Rep name: cp_2020_21_Gnn (where "nn" is the group number)
  - Add me as an "observer" [Read-Only]
- **Each group member** will **commit regularly** his/her individual contributions to the group repository
  - Commit logs/messages must clearly state the contributions
- Individual grade will consider individual contributions committed the GIT repository
  - No meaningful commits/contributions
    => no frequency (failing the course)
- *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®

Administrivia — 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 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

Administrivia — Project method.

- 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

Remember…

- Clip is the official source of information for the course.
- Confirm @Clip all the administrivia related topics.
  - In case of contradiction, is the information in Clip that 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