SOFTWARE ENGINEERING

Laurent Poirrier

2023-09-06

PRACTICAL ORGANIZATION

Lectures

- Room: Sarfatti 25, Aula 5
- Wednesdays 10:15 – 12:45 / Fridays 8:30 – 10:00 **

** with many exceptions

Today after class

- Welcome Event for the MSc in AI.
- 18:15, room Manfredini
- “The Benefits of Complexity” – by Luca Trevian (Bocconi)
- Talk on computer security – by Salil Vadhan (Harvard)
- Cocktail 🍸

Material

- No reference book
- Links will be shared for some of the lectures
- Slides will always be available

Evaluation

- 20% assignments (individual)
- 20%+ project (individual or groups of 2)
- written exam
  - multiple-choice questions
  - open-ended questions
  - coding exercises (open-book, open-laptop, no internet)

COURSE OVERVIEW

From low-level electronics to high-level project development.

PART I: HARDWARE AND SOFTWARE

- Hardware architectures
  - Boolean logic
  - Integer arithmetic
  - Hardware structure
  - Instructions
  - Memory
- Programming languages
  - Compilers and interpreters
  - Assembly
  - Higher-level languages
  - Types
  - Memory management
  - Other language features

PART II: SOFTWARE DEVELOPMENT

- Version control systems
- Deployment
- Dependencies
- Portability

PART III: CORRECTNESS

- Machine and language specifications
  - Unspecified behavior, implementation-defined behavior, undefined behavior
  - Memory ordering and barriers
  - Floating-point arithmetic
- Software engineering practices
  - Documentation
  - Testing
  - Static analysis and refactoring
  - Dynamic analysis
  - Assertions
  - Fuzzing

PART IV: PERFORMANCE

- Algorithmic performance
- Code optimization
    - Out-of-order execution, CPU pipelines
    - Cache, memory, storage and network
    - Benchmarking and static instrumentation
    - Stochastic instrumentation
    - SIMD instructions
    - Thread-level concurrency
    - Distributed computing
    - Hardware acceleration

CHOICE OF PROJECT TOPIC

- Submit your own topic
- Subject to my approval
- There will be a deadline for topic submission (but changes are possible)
- I will make suggestions

EXAMPLE TOPICS

- add **features** to an open source project (ideally useful to you, look at e.g. F-Droid apps)
- improve **performance** of an open source project
  - aim for low-hanging fruit
  - performance is not just speed: memory, network data, power
- **find bugs** in an open source project
  - aim for low-hanging fruit
- **fix bugs** in an open source project
  - look at bugzilla, github/gitlab issues
- develop your own project (ideally useful to you)

Project organization

- Individual or groups of two
- I will help you in class and after class

Policy for participation in open source projects

- no extra marks for getting “upstreamed”
- you “must” get my approval before contacting project developers (email, pull requests, etc.)

Project grading

- Overall weight 20% of final grade at least
- More than 20% for outstanding projects
- You will write a half-page (1-page max) report

Evaluation criteria

- Correctness
- Technical difficulty
- Originality
- Impact and presentation