Principles of Operating Systems and Concurrent Programming

Welcome!

Thomas Ropars

thomas.ropars@univ-grenoble-alpes.fr

2024

Important Pointers

Class Web page

- https://m1-mosig-os.gitlab.io
- Lecture slides, agenda, and more

Moodle page

- https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=228
- Labs, news, and discussion forum

Important Pointers

Reference book

- https://pages.cs.wisc.edu/~remzi/OSTEP/
- **Remzi Arpaci-Dusseau and Andrea Arpaci-Dusseau. Operating Systems: Three Easy Pieces.**
- Electronic book, freely available on-line
- More references? https://m1-mosig-os.gitlab.io/resources/#reference-books

Contacts

Teaching staff

- Thomas Ropars (thomas.ropars@univ-grenoble-alpes.fr): Lectures (and Labs)
- Nicolas Homberg (nicolas.homberg@univ-grenoble-alpes.fr): Labs
- Albin Petit (albin.petit@inria.fr): Labs

Contacting us

- Email is the fastest and most convenient way to reach us
  - Add [M1 Mosig - OS] to the subject of your emails
- For questions of general interest: A discussion forum is open on Moodle

Administrivia

- **Schedule**
  - Lectures: 1 slot (3 hours) per week
    - Usually on Tuesday afternoons (1:30-4:45 pm)
  - Practical work / lab sessions: 1 slot (3 hours) per week
    - Usually on Thursday mornings (8:00-11:15 am)
  - No OS class/lab on week 44 (fall vacation)
- **Warning**: Some modifications may occur during the semester. Please check regularly both:
  - The web page of the class
  - ADE

Administrivia

- **Exams**
  - Midterm exam(s):
    - At least two (details to be confirmed)
    - Will be announced/confirmed 1 week in advance (in class)
  - Final exam: week 49 or 50, 3 hours (to be confirmed)
  - **Warning**: No documents allowed, except 1 handwritten paper sheet (A4 format)

Grading

- 20% from practical projects
- 20% from midterms
- 60% from the final exam

Style

- You must turn in a design document along with your code
  - Remember that figures are often very useful to convey/summarize design ideas
  - Please respect the document format(s) imposed by the teaching staff (e.g., PDF)

- Instructors will manually inspect your code for correctness
  - Does the code respect the specification?
  - Does it actually implement the described design?
  - Does it handle corner cases (e.g., handle malloc failure)?

- Instructors will deduct points for error-prone code without errors
  - Do not use global variables if local ones suffice
  - Do not use obscure/misleading names for variables/functions

Style (continued)

- **Your code must be easy to read**
  - Keep lines (and when possible) functions short
  - Use a uniform coding style (try to match existing code)
  - **Indent** your code in a clean and consistent way
  - **Put comments** on structure members, global variables, functions
  - Do not leave lots of commented-out garbage code

Assignment requirements

- **Project deadlines are firm**
- **If you run into trouble, contact instructors in advance** to ask for an extension
- **Do not look at other people's solutions to projects**
- You can read but must not copy the code of existing (open source) systems/applications available on the Internet (e.g., Linux).
- **Cite any code that inspired your code**
  - As long as you cite what you used, it is not cheating
  - In the worst case, we will deduct points if this undermines the assignment

Additional details regarding collaborations with other students & plagiarism (1/2)

You are encouraged to discuss ideas and problems related to the programming assignments with the other students. You can also look for additional resources on the Internet.

However, **we consider plagiarism and cheating very seriously**. Hence, if any part of your final submission reflects influences from external sources, you must cite these external sources in your report and clearly indicate what you have taken from them. Also, **any part of your designs, your implementations, and your reports should come only from you and not from other students**.

We will run tests to detect similarities between source codes. Additionally, we will allow ourselves to question you about your submission if part of it looks suspicious, which means that you should be able to explain each line (of text and code) that you have submitted.

Additional details regarding collaborations with other students & plagiarism (2/2)

- **Wrap up and general guidelines:**
  - Do not try to read/reuse solutions provided by other students (or past students, or teachers from other/previous classes, even outside UGA).
  - Precisely cite/identify all the material from external sources that you have used to complete your assignment.
  - Do not let other students access your own solutions. In particular:
    - Make sure to protect your files on the University servers/printers.
    - Do not publish your work on the Internet (GitHub, blog, …)
  - However, you are free (and encouraged) to help your classmates. For example, it is OK to help them understand a bug in their programs or to re-explain the goal of an exercise.
  - **If you are in doubt about any rule/detail, contact the teaching staff.**

Rules regarding AI assistants

- Nowadays, AI assistants have become very popular tools:
  - Examples: ChatGPT, Github Copilot, etc.

- **Our policy for this course:**
  - **For lab sessions that are not graded:** AI assistants are allowed but not recommended
  - **For lab sessions that are graded:** AI assistants are forbidden (both for code and reports)
  - If you have any specific question/request on this matter, please contact us in advance.