

Exercise Sheet 0 for Lecture Parallel Programming

Deadline: No submission necessary

Prof. Dr. Michael Kuhn (michael.kuhn@ovgu.de)

Michael Bleisel (michael.bleisel@ovgu.de)

Parallel Computing and I/O • Institute for Intelligent Cooperating Systems

Faculty of Computer Science • Otto von Guericke University Magdeburg

<https://parcio.ovgu.de>

1. Organization

The exercises are intended to take a deeper look at the topics presented in the lecture using practical assignments. The deadlines for submission are mentioned on each exercise sheet. The time required for the exercises should be around 6 hours per week.

If you have any questions, please use our Mattermost team:

https://mm.cs.ovgu.de/signup_user_complete/?id=c8erx4z9ytnpdqknuhauuyzrua&md=link&sbr=su

Experience has shown that many groups lose a lot of time with seemingly easy problems. It is important to us that you do not waste your time with irrelevant tasks but can focus on the important ones.

The exercises will be graded.

1.1. Exercise Structure

Each exercise sheet consists of two parts: Practical exercises and feedback. The exercise sheets grant variable amounts of points, where the working time in minutes should roughly be equivalent to the amount of points.

1.1.1. Feedback (10–15 Bonus Points)

The purpose of the feedback is for you to give us your opinions on the exercise sheets, the lecture and the exercises. Based on how detailed the feedback is, we will grant 10–15 bonus points. Please embrace this opportunity so that we can improve the later exercise sheets and the lecture for the coming years. You can state your honest opinions about everything. Negative feedback will of course never influence how we grade your submissions.

The feedback can for example contain the following topics (you can also include anything else that you find relevant):

- Were the tasks and questions clear and understandable? If not, how could we improve them?
- Did you have all relevant information to work on the tasks? If not, what was missing?
- Were there problems while working on the tasks? If so, please elaborate.

- How was the last exercise? What could have been better? What was unnecessary? What did you like?
- How did you like the last lecture? What could have been better? What was unnecessary? What did you like?
- Do you have any additional comments about the exercise sheet?

The feedback template contained in the exercise materials (`feedback.txt`) is mainly concerned with the exercise sheets and should always be submitted. We are however happy about any additional feedback text about the lecture and exercises included in the file.

1.2. Group Work

Exercise submissions have to happen in groups of three persons. Cooperation with other groups is of course allowed, but it is not allowed for multiple groups to hand in identical solutions. Obvious copies of solutions will not be accepted by us and will be graded with zero points.

2. Submissions

For the exercises, each group will be assigned a Git repository in the university GitLab. Your last commit on the `main` branch before the deadline of an exercise sheet will count as your submission for the exercise sheet. If there are reasonable grounds for a delayed submission, the exercise instructor has to be informed **in advance**. The following general points should be followed for the exercise submissions:

- **All** required materials have to be included in your submission. If the file names were specified (e. g. `answers.md`) they have to be adhered to. The exact requirements for each submission are stated on the exercise sheets (please double check before submitting).
 - Files need to be in a plaintext (UTF-8) or PDF format. Hint: You can use the *Acrobat PDF-Writer for Windows*.
 - No binary or object files should be submitted. These just take up unnecessary space in the submission.
 - Always provide a `Makefile` so that the program can be built with `make`.
 - No file or directory names shall contain spaces or other special characters (“-” and “_” are of course allowed).
- Additional information about the submissions will be given on the exercise sheets.

To give you early feedback about the correctness of a submission the Git repositories use Continuous Integration (CI). For each commit, a number of tests is run automatically:

- The existence and the correct paths for the required files.
- Whether your program compiles and is executable.

- In some cases, the programs are also checked for the correctness of outputs for some input parameters.

It is important here that a correctly running CI pipeline does not necessarily mean that your submission is completely correct. We can not test for all possible errors automatically and it is still your responsibility to make sure that your submission fulfills all requirements of the exercise tasks. However, the CI pipeline should give you feedback about obvious errors and thereby hopefully support your work on the tasks.

2.1. Grading Criteria

When grading your solutions, we consider the following points:

- Submission of your solutions on time
- Well-structured, documented and executable source code
- Completeness of the required materials
- Correctness of your solutions
- Correctness of the file names and paths

3. Presenting your Solutions

Every student has to present a solution at least once in the exercises. Refusal to present a solution will lead to the withdrawal of all points that have already been granted for the task. Even if a group member has not actively participated in the solution of a specific task, they need to understand the submitted solution well enough to be able to present it in the exercise. Unsatisfactory presentations can be grounds for a point deduction.

4. Passing the Exercises

Attending the exercises is mandatory. During the semester, you can be absent without valid excuse for at most two exercise dates. To pass the exercises, you need at least 50 % of the overall total points (without bonus points).

On at least 5 out of the 7 exercise sheets you need to reach at least 25 % of the total points. Moreover, submitting the exercise sheet about Jacobi parallelization with MPI is mandatory.