Institute of Computer Science
Department of Mathematics and Computer Science

Freie Universität Berlin

Algorithms and Programming IV

Course Concept and Organization

Summer Term 2024 | 15.04.2024

Claudia Müller-Birn, Barry Linnert

Target Group of this Course

- You are in the fourth semester of your bachelor's degree in computer science or computer science for a teaching profession.
- You have attended the modules "Functional Programming", "Concepts of imperative and object-oriented programming" and the course "Computer Engineering II: Computer Architecture", "Operating and Communication Systems" in the module "Computer Architecture, Operating and Communication Systems".
- You have no previous knowledge in the area of parallel and distributed programming.

Freie Universität Berlin

Integration of the Course into the BSc. Programme

Semester	Algorithmen und Programmierung	Technische Informatik	Theoretische Informatik und Praktische Informatik	Mathematik für Informatik	Wissenschaft	Anwendungs-bereich	ABV	SWS
1. FS 28 LP	Funktionale Programmierung (9 LP)	Rechnerarchitektur, Betriebs- und Kommunikations-systeme (10 LP)		Logik und Diskrete Mathematik (9 LP)			ABV (5 LP)	18
2. FS 30 LP	Objekt-Orientierte Programmierung (8 LP)		Grundlagen der Theoretischen Informatik (7 LP)	Lineare Algebra für Informatik (10 LP)				20
3. FS 29 LP	Algorithmen, Datenstrukturen und Datenabstraktion (9 LP)		Auswirkungen der Informatik (5 LP)	Analysis für Informatik (10 LP)			ABV (5 LP)	18
4. FS 31 LP	Nichtsequentielle und verteilte Programmierung (9 LP)		Datenbank-systeme (7 LP)		Wissenschaft-liches Arbeiten in der Informatik (5 LP)			18
			Software-technik (10 LP)					

Learning Objectives of this Course

- Students should be able to differentiate relevant terms in this field.
- Students should be able to know and apply relevant interaction paradigms for concurrent, parallel, and distributed programming to a given problem.
- Students should be able to evaluate relevant interaction paradigms for concurrent, parallel, and distributed programming and understand their advantages and disadvantages.

Mainly Used Text Books

- Rauber, Thomas, and Gudula Rünger: Parallele und verteilte Programmierung, Springer-Verlag, 2013
- Andrews, Gregory R.: Foundations of multithreaded, parallel, and distributed programming, Vol. 11, Reading: Addison-Wesley, 2000.
- Quinn, Michael J.: Parallel Programming in C with MPI and OpenMP, McGrawHill, 2003

- Coulouris, Dollimore, Kindberg, Blair: Distributed Systems: Concepts and Design, Addison-Wesley, May 2011
- Andrew S. Tanenbaum, Maarten van Steen: Distributed Systems: Principles and Paradigm, Pearson Education Limited, 2013

Results from Evaluation last Semester

- We received good or average results in our course evaluation.
  - Exception: References to other courses and further studies, application-orientation
- Positive:
  - Interaction
  - Band Shirt Contest
- Improvable:
  - Slides
  - C-Programming and practice sheet
  - Lecture is too slow, lecture is too fast
- Miscellaneous:
  - Interaction with (some) students

Concepts of Non-Sequential, Parallel and Distributed Programming

ORGANISATION

Course Structure

Lecture
Mon 2 to 4 PM / Wed 6 to 6 PM

Exercise / Tutorial
Mon to Fri as selected in Whiteboard

Examination	Wed 17.07.2024, 4 to 6 PM
Retake	Wed 09.10.2024, 10 AM to 12 PM

Time Required for Attending this Course

Lecture	Time of attendance 60 h Pre- and postprocessing 30 h
Exercise / Tutorial	Time of attendance 30 h Pre- and postprocessing 120 h
Examination	Exam preparation and examination 30 h

Organisational Setting

- You should be registered in Whiteboard (WB): https://mycampus.imp.fu-berlin.de
  - In the WB, you will find all the lecture and exercise information. You also use the WB to communicate if you have questions about the lecture or exercise. Please do not ask questions by e-mail; use the forum on Whiteboard.

- You should be registered in Campus Management for the current semester: https://www.ecampus.fu-berlin.de/
  - With your registration in Campus Management, you are bindingly registered for the module and, thus, for the examination.

Concepts of Non-Sequential, Parallel and Distributed Programming

APL IV TEAM

Lecturer

- **Prof. Dr. Claudia Müller-Birn**
  - Head of the Research Group Human-Centered Computing
  - Human-Computer Interaction (BSc), Data Visualization (BSc), Advanced Concepts of Data Visualization (MSc), Interactive Technologies (MSc), Human-Centered Data Science (MSc)
  - Software projects and BSc/MSc theses in the area of Human-Computer Interaction

- Barry Linnert
  - Member of Research Group Empirical Software Engineering
  - Betriebssysteme (MSc) and Cluster Computing (MSc)
  - Software projects and BSc/MSc theses in the field of operating systems and HPC

Exercise Group Leader – Tutorials

- Jonah Brüchert

Thu 2 PM – 4 PM	Fri 2 PM – 4 PM

- Mehmed Günes

Mo 10 AM – 12 PM	Mo 12 PM – 2 PM

- Markus Mielimonka

Thu 8 AM – 10 AM	Fri 10 AM – 12 PM

- Mert Yaylaci

Mo 4 PM – 6 PM	Fri 4 PM – 6 PM

Concepts of Non-Sequential, Parallel and Distributed Programming

COURSE CONCEPT

Concept of Lecture

- We do not teach one programming language in the lecture, but you should understand the underlying concepts of programming languages.

- You will not learn programming during the lecture but only when applying the insights within practical software projects. For example, we recommend participating in Open Source Software (OSS) projects.

Concept of the Exercise

- You will not learn programming in the exercise but will deepen concepts from the lecture using selected programming languages (C, Python, Javascript).

- You will work in small groups in the exercise. If you are working with an already experienced student, do not rest on these skills, but work together with the other students and support them (peer learning!).

- Use the competence of the exercise group leaders to improve your understanding.

Concepts of Non-Sequential, Parallel and Distributed Programming

ASSIGNMENTS

Criteria for the Confirmation of Regular and Active Participation

- **Regular participation** in the exercises/tutorials is essential.

- Each student must work through **(n-1)** exercise sheets and provide **reasonable answers** to each task.

- There are 12 exercise sheets.

- Each student has to **present** her/his **solutions at least twice**. Students should register with the tutor before the tutorial to be able to present. Without registration, the tutor decides randomly.

Delivery of the Exercise Sheets

- The publication of the exercise sheets takes place on Mondays at 8 AM.
- The results of the exercise sheet should be available via our FU git: https://git.imp.fu-berlin.de/.
  - One group member creates a private project containing the names of all group members in the project description, adds the remaining members and the tutor, and gives them sufficient access rights.
  - Each exercise sheet and the code gets its own branch.
  - The solution to an exercise sheet consists of the source code, a Makefile or index.js with a package.json file and a PDF. The PDF is to be created with the LaTeX containing all the group members' names, the documentation, and further solutions. We made a template: https://www.overleaf.com/read/dbctyypwypqj#bc34c5

Delivery of the Exercise Sheets II

- The source code should be adequately commented on, and, if necessary, the approach/idea of programming tasks should be described in the PDF.
- The source code has to be compilable and runnable on andorra.imp.fu-berlin.de.
- The link to the commit, which represents the most recent version of the submission, is submitted on the Whiteboard by one group member.

• The delivery of the exercise results is required by Fridays until 7:55 PM on the Whiteboard under Assignment. Exercise sheets submitted after this date will not be accepted.