



Parallel Processing

Winter Term 2024/25

Roland Wismüller
Universität Siegen
roland.wismueller@uni-siegen.de
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: October 10, 2024



Parallel Processing

Winter Term 2024/25

0 Organisation

- ➔ Studies in Computer Science, Techn. Univ. Munich
 - ➔ Ph.D. in 1994, state doctorate in 2001
- ➔ Since 2004 Prof. for Operating Systems and Distributed Systems
- ➔ **Research:** Secure component based systems; Using generative AI for teaching; Parallel and distributed systems
- ➔ Head of Examination Board
- ➔ **E-mail:** roland.wismueller@uni-siegen.de
- ➔ **Tel.:** 0271/740-4050
- ➔ **Room:** H-B 8404
- ➔ **Office Hour:** Mo., 14:15-15:15

About the Chair “Operating Systems / Distrib. Sys.”



Andreas Hoffmann

andreas.hoffmann@uni-...
0271/740-4047
H-B 8405

- ➔ E-assessment and e-labs
- ➔ IT security
- ➔ Web technologies
- ➔ Mobile applications



Felix Breitweiser

felix.breitweiser@uni-...
0271/740-4719
H-B 8406

- ➔ Operating systems
- ➔ Programming languages
- ➔ Virtual machines



Sven Jacobs

sven.jacobs@uni-...
0271/740-2533
H-B 8407

- ➔ E-assessment and e-labs
- ➔ Generative artificial intelligence
- ➔ Web technologies

Lectures/Labs

- ➔ Rechnernetze I, 6 CP (Bachelor, summer term)
- ➔ Rechnernetze Praktikum, 6 CP (Bachelor, winter term)
- ➔ Rechnernetze II, 6 CP (Master, summer term)

- ➔ Betriebssysteme und nebenläufige Programmierung, 6 CP (Bachelor, summer term)
- ➔ Parallel processing, 6 CP (Master, winter term)
- ➔ Distributed systems, 6 CP (Bachelor, winter term)

Teaching ...

Project Groups

- ➔ e.g., secure cooperation of software components
- ➔ e.g., concepts for secure management of Linux-based thin clients

Theses (Bachelor, Master)

- ➔ Topic areas: secure virtual machine, parallel computing, pattern recognition in sensor data, e-assessment, ...

Seminars

- ➔ Topic areas: IT security, programming languages, pattern recognition in sensor data, ...
- ➔ Procedure: block seminar (30 min. talk, 5000 words paper)
- ➔ Master: attend the lecture “Scientific Working” beforehand!
 - ➔ block course end of Feb. / beginning of March

Notes for slide 6:

A note on external Master theses: The right to give you a topic for a Master thesis lies with the University only!

This means, if you want to do a thesis at an external company or research institute, you **first** have to find a professor who will supervise you, and then, if she or he is interested, the professor may define a topic together with the company.

Please have a look at our [handout on conducting external theses](https://www.eti.uni-siegen.de/dekanat/pruefungsamt/dokumente/studien-ganguebergreifend/externe-abschlussarbeiten-eti-en.pdf)!^a

^a<https://www.eti.uni-siegen.de/dekanat/pruefungsamt/dokumente/studien-ganguebergreifend/externe-abschlussarbeiten-eti-en.pdf>

6-1

About the Lecture



Lecture

- ➡ Mon., 12:15-13:45, AR-B 2104/05
- ➡ on 08.10., 15.10., 22.10., and 29.10. also in the lab slot!
 - ➡ Tue., 10:15-11:45, H-C 6321

Practical labs

- ➡ Preferably at home
 - ➡ if necessary, you can also use the PC lab room H-A 4111
- ➡ Tutor: Felix Breitweiser (felix.breitweiser@uni-siegen.de)
- ➡ Questions and help: via Discord server
 - ➡ <https://discord.gg/UZTv8yptqj>
- ➡ Discussion of solutions: Tue., 10:15-11:45, H-C 6321
 - ➡ only on the due date of an assignment!



Information, slides, and announcements

- ➔ See the WWW page for this course
- ➔ <http://www.bs.informatik.uni-siegen.de/lehre/pv/>
- ➔ Annotated slides (PDF) available; maybe slightly modified

Moodle course

- ➔ <https://moodle.uni-siegen.de/course/view.php?id=23366>
- ➔ Recorded screen casts of the lecture (from winter term 2020/21)
- ➔ Submission of lab assignments



Discord invite link





Link to course's homepage



Learning targets

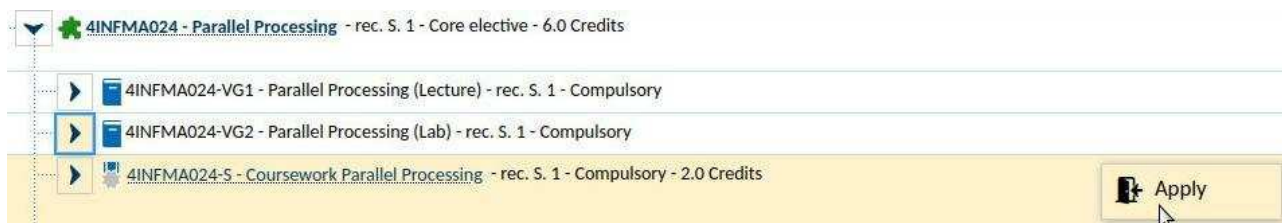
- ➔ Knowing the basics, techniques, methods, and tools of parallel programming
- ➔ Basic knowledge about parallel computer architectures
- ➔ Practical experiences with parallel programming
- ➔ Knowing and being able to use the most important programming models
- ➔ Knowing about the possibilities, difficulties and limits of parallel processing
- ➔ Being able to identify and select promising strategies for parallelization
- ➔ Focus: high performance computing

Methodology

- ➔ Lecture: Basics
 - theoretical knowledge about parallel processing
 - practical introduction to programming environments
 - “hands-on” tutorials
- ➔ Lab: practical use
 - **independent programming work**
 - practical skills and experiences
 - in addition: raising questions
 - different parallelizations of two representative problems
 - iterative, numerical method (Jacobi, Gauss/Seidel)
 - combinatoral search (Sokoban)

Registration for “Course Achievement” (Studienleistung)

- ➔ Passing the course requires successful completion of the lab:
 - i.e., qualified attempt for all mandatory exercises
 - Exam Regulations 2012: prerequisite for the exam!
- ➔ You must register for the
 - “Coursework Parallel Processing” 4INFMA024-S, or
 - “Prüfungsvorleistung” 822120-Sin unisono **before you can submit a solution!** (do it right now!)
 - independent of the registration to the course and the lab!
 - if you cannot complete the course: **deregister** again!



The screenshot shows a web-based course selection interface. At the top, a green star icon is next to the text "4INFMA024 - Parallel Processing - rec. S. 1 - Core elective - 6.0 Credits". Below this, there are three rows of course options, each with a blue square icon and a right-pointing arrow. The first row is "4INFMA024-VG1 - Parallel Processing (Lecture) - rec. S. 1 - Compulsory". The second row is "4INFMA024-VG2 - Parallel Processing (Lab) - rec. S. 1 - Compulsory". The third row is "4INFMA024-S - Coursework Parallel Processing - rec. S. 1 - Compulsory - 2.0 Credits". At the bottom right of the interface, there is a button labeled "Apply" with a mouse cursor pointing at it.

Notes for slide 13:

If you are not registered for the course achievement, you will not be able to submit any solutions in the Moodle platform (the corresponding section will not be available in Moodle).

Since data is transferred between unisono and Moodle only about once a week, you should register way in advance!

13-1

Examination



- ➡ Written examination (60 minutes)
 - ➡ electronic exam, computers provided by university
 - ➡ subject matter: lecture and labs!
 - ➡ examination also covers the practical exercises
- ➡ Application via unisono
 - ➡ **at least two weeks before the exam date (hard deadline!)**
 - ➡ exam date is published via unisono and course web page
 - ➡ if you study Computer Science with Exam Regulations 2012, you first must have your mentor's approval
 - ➡ **be sure to meet the deadline!**

Organisational Issues regarding the Labs



- ➔ Assignments should be done at home, if possible
- ➔ Programming is done in C/C++
- ➔ Ideally, you need a Linux-PC with the GNU-compilers (gcc/g++)
 - ➔ Windows with MSVC will also work, except for one exercise sheet
- ➔ In addition, you need to install MPI, preferable MPICH
 - ➔ see <https://www.mpich.org/downloads>
- ➔ Four exercise sheets
 - ➔ code must be submitted via Moodle in due time
 - ➔ different requirements depending on 5 CP vs. 6 CP

Contents of the Lecture



- ➔ Repetition / Foundations
 - ➔ C/C++ for Java programmers
 - ➔ Threads and synchronisation
 - ➔ C++ threads
- ➔ Basics of Parallel Processing
 - ➔ Motivation, Parallelism
 - ➔ Parallelization and Data Dependences
 - ➔ Parallel Computers
 - ➔ Programming Models
 - ➔ Organisation Forms for Parallel Programs
 - ➔ Performance Considerations
 - ➔ Design Process



- ➔ Parallel Programming with Shared Memory
 - ➔ Basics
 - ➔ OpenMP
- ➔ Parallel Programming with Message Passing
 - ➔ Approach
 - ➔ MPI
- ➔ Optimization Techniques
 - ➔ Cache Optimization
 - ➔ Optimization of Communication

Time Table of Lecture and Labs



- ➔ Until October, 29th: only lectures (Mon. + Tue.), no lab
- ➔ Then: lectures (Mon.) and lab (home work + Tue.)
- ➔ Last two weeks: only lab
- ➔ Prospective due dates for the assignments:
 - ➔ 05.11.: Exercise sheet 1
 - ➔ ... (see [web page](#))
 - ➔ 28.01.: Exercise sheet 8
 - ➔ On due date: presentation and discussion of assignments in H-C 6321



- ➔ Currently no recommendation for a all-embracing text book
- ➔ Barry Wilkinson, Michael Allen: *Parallel Programming*. internat. ed, 2. ed., Pearson Education international, 2005.
 - ➔ covers most parts of the lecture, many examples
 - ➔ short references for MPI, PThreads, OpenMP
- ➔ A. Grama, A. Gupta, G. Karypis, V. Kumar: *Introduction to Parallel Computing*, 2nd Edition, Pearson, 2003.
 - ➔ much about design, communication, parallel algorithms
- ➔ Thomas Rauber, Gudula Rünger: *Parallele Programmierung*. 2. Auflage, Springer, 2007.
 - ➔ architecture, programming, run-time analysis, algorithms



- ➔ Theo Ungerer: *Parallelrechner und parallele Programmierung*, Spektrum, Akad. Verl., 1997.
 - ➔ much about parallel hardware and operating systems
 - ➔ also basics of programming (MPI) and compiler techniques
- ➔ Ian Foster: *Designing and Building Parallel Programs*, Addison-Wesley, 1995.
 - ➔ design of parallel programs, case studies, MPI
- ➔ Seyed Roosta: *Parallel Processing and Parallel Algorithms*, Springer, 2000.
 - ➔ mostly algorithms (design, examples)
 - ➔ also many other approaches to parallel programming

- ➡ S. Hoffmann, R. Lienhart: *OpenMP*, Springer, 2008.
 - ➡ handy pocketbook on OpenMP
- ➡ W. Gropp, E. Lusk, A. Skjellum: *Using MPI*, MIT Press, 1994.
 - ➡ the definitive book on MPI
- ➡ D.E. Culler, J.P. Singh: *Parallel Computer Architecture - A Hardware / Software Approach*. Morgan Kaufmann, 1999.
 - ➡ UMA/NUMA systems, cache coherency, memory consistency
- ➡ Michael Wolfe: *Optimizing Supercompilers for Supercomputers*, MIT Press, 1989.
 - ➡ details on parallelizing compilers