



---

# Parallel Processing

WS 2020/21

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

Stand: October 22, 2020



---

# Parallel Processing

WS 2020/21

## 0 Organisation

## About Myself



- ➔ 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; Pattern recognition in network data; Parallel and distributed systems
- ➔ **Mentor** for Bachelor Studies in Computer Science with secondary field Mathematics
  
- ➔ **E-mail:** roland.wismueller@uni-siegen.de
- ➔ **Tel.:** 0271/740-4050
- ➔ **Room:** H-B 8404
- ➔ **Office Hour:** Mo., 14:15-15:15 Uhr

## 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



**Damian Ludwig**  
damian.ludwig@uni-...  
0271/740-2533  
H-B 8402

- ➔ Capability systems
- ➔ Compilers
- ➔ Programming languages



**Hawzhin Hozhabr Pour**  
hawzhin.hozhabrpour@uni-...  
0271/740-4038  
H-B 8411

- ➔ Machine learning
- ➔ Pattern recognition in car sensor data
- ➔ Anomaly detection



## Lectures/Labs

- ➔ Rechnernetze I, 5 LP (Bachelor, every summer term)
- ➔ Rechnernetze Praktikum, 5 LP (Bachelor, every winter term)
- ➔ Rechnernetze II, 5 LP (Master, every summer term)
  
- ➔ Betriebssysteme I, 5 LP (Bachelor, every winter term)
- ➔ Parallel processing, 5 LP (Master, every winter term)
- ➔ Distributed systems, 5 LP (Master, every summer term)

# Teaching ...



## Project Groups

- ➔ e.g., recording and analyzing car sensor data
- ➔ e.g., outlier detection in car sensor data

## 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 word seminar paper



- ➔ **Lecture + practical Labs:** 2+2 SWS, 5 LP
  - ➔ Tutor: Felix Breitweiser (felix.breitweiser@uni-siegen.de)
- ➔ **Date and Time:**
  - ➔ **No presence teaching in this semester!**
  - ➔ The lecture will be provided digitally via moodle
  - ➔ The lab will also be organized via moodle
- ➔ **Information, slides, and announcements:**
  - ➔ In the WWW
    - ➔ <http://www.bs.informatik.uni-siegen.de/lehre/pv/>
    - ➔ annotated slides (PDF) available; maybe slightly modified
  - ➔ Screen casts available in moodle



### 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
- ➔ Lab: practical use
  - ➔ practical introduction to programming environments
  - ➔ “hands-on” tutorials
  - ➔ **independent programming work**
  - ➔ practical skills and experiences
  - ➔ in addition: raising questions
  - ➔ different parallelizations of two representative problems
    - ➔ iterative, numerical method
    - ➔ combinatoral search

### Examination



- ➔ Oral examination (about 30-40 min.)
  - ➔ subject matter: lecture and labs!
  - ➔ examination also covers the practical exercises
- ➔ Application:
  - ➔ first, apply online via unisono
    - ➔ **at least** one week (better: a few weeks) in advance
    - ➔ note that you first must have your mentor’s approval (**deadline**: see [web page of the examination office!](#))
  - ➔ then, fix a date with my secretary Mrs. Syska
    - ➔ via email ([regina.syska@uni-siegen.de](mailto:regina.syska@uni-siegen.de))
    - ➔ or personally (H-B 8403, in the morning)

## Registration for “Course Achievement” (Studienleistung)

- ➔ Prerequisite for admission to the exam:
  - active attendance to the labs
  - i.e., qualified attempt for all (i.e., 4) main exercises
- ➔ You have to register in unisono for the “Course Achievement” (Studienleistung) 822120-SL until **end of January!**
  - independent from the registration to the course and the lab!
  - the best is to **do it right now!**



Course ID	Course Name	Credits	Registration Button
37	Parallel Processing - core elective	5.0	Apply wt 2019
	Parallelverarbeitung (Vorlesung) - compulsory		
43BV50501V	Parallel Processing - Event		Apply wt 2019
	Parallelverarbeitung (Übung) - compulsory		
43BV50502V	Parallelverarbeitung - Exercise		Apply wt 2019
822120-SL	Parallelverarbeitung - compulsory	0.0	Apply
822120	Parallel Processing - core elective	5.0	Apply

## Organisational Issues regarding the Labs

- ➔ In this semester, you will have to do the assignments at home
- ➔ Programming is done in C/C++
- ➔ Ideally, you need a Linux-PC with the GNU-compilers (gcc/g++)
  - however, Windows with MSVC will also work
- ➔ In addition, you need to install MPI, preferable MPICH
  - see <https://www.mpich.org/downloads>
- ➔ You also get an account on the high performance computer HoRUS, accessible via SSH ([hpc.zimt.uni-siegen.de](http://hpc.zimt.uni-siegen.de))
  - 136 nodes, each with two Intel XEON processors (2 \* 6 cores), 2.66 GHz, 48 GB RAM
- ➔ Four major assignments
  - code must be submitted via **moodle** in due time



- ➔ Basics
  - ➔ Motivation, Parallelism
  - ➔ Parallel Computers
  - ➔ Parallelization and Data Dependences
  - ➔ Programming Models
  - ➔ Design Process
  - ➔ Organisation Forms for Parallel Programs
  - ➔ Performance Considerations
- ➔ Parallel Programming with Shared Memory
  - ➔ Basics
  - ➔ POSIX Threads
  - ➔ OpenMP



- ➔ Parallel Programming with Message Passing
  - ➔ Approach
  - ➔ MPI
- ➔ Optimization Techniques
  - ➔ Cache Optimization
  - ➔ Optimization of Communication

## Time Table of Lecture and Labs



- ➔ First three weeks: only lectures, no lab
- ➔ Then: lectures and lab
- ➔ Last two weeks: only lab
  
- ➔ Due dates for the major assignments:
  - 03.12.: Quicksort using PThreads
  - 07.01.: Jacobi using OpenMP
  - 28.01.: Sokoban using OpenMP
  - 11.02.: Jacobi using MPI

## Consultation hours



- ➔ Online meetings using the `zoom` video conference system
  - see [the documentation in the WWW](#)
  - only for questions on the lecture / the lab
- ➔ For the lecture: **Mon., 12:00 - 12:30** (or longer, if necessary)
- ➔ For the lab: **Thu., 14:00 - 15:00** (or longer, if necessary)
- ➔ Links and access information for the zoom meetings are published in [moodle](#)!





- ➔ 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