

OPERATING-SYSTEM CONSTRUCTION

Introduction

<https://tud.de/inf/os/studium/vorlesungen/betriebssystembau>

TILL SMEJKAL (slides by HORST SCHIRMEIER)

Overview

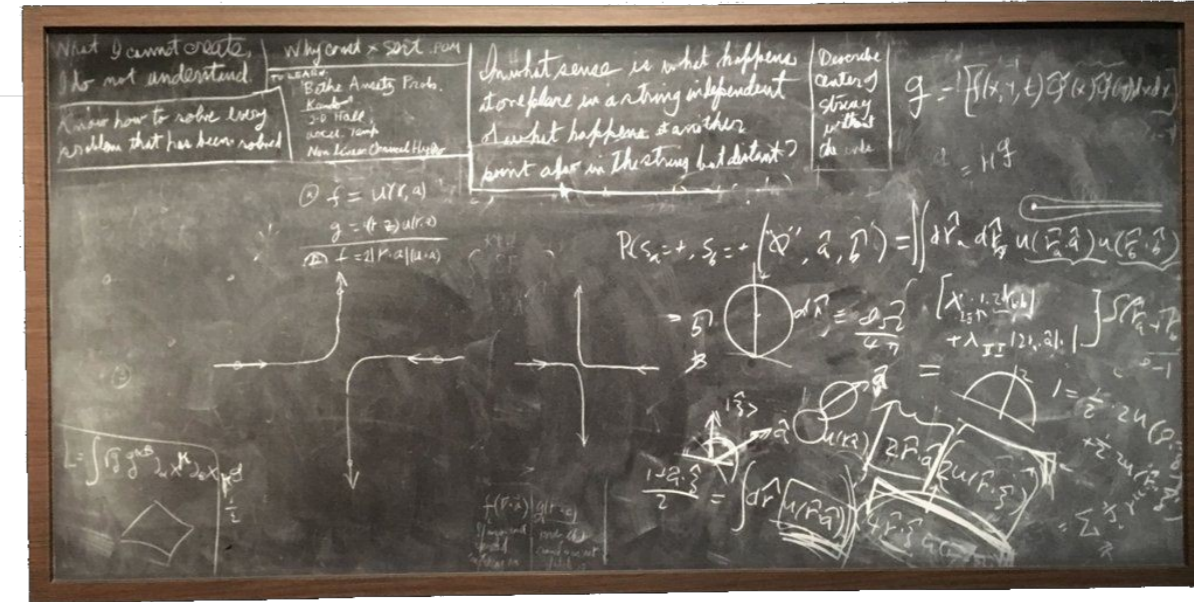
- Organization
- Lecture Contents
- Exercise Contents

Overview

- **Organization**
- Lecture Contents
- Exercise Contents

Learning Objective

- Deepen knowledge on operating systems
 - Functionality
 - Structure
 - Implementation
- Learning By Doing: OO-StuBS
 - Develop an OS from scratch
 - Understand HW/SW interface and PC technology



*What I cannot create,
I do not understand.*

- Richard Feynman
(on his last blackboard)

**Strong recommendation: Actively participate
in the lab exercises, hand in solutions!**

Prerequisites

- You ...
- ... have **basic knowledge on OSs** (e.g. from BuS)
- ... like **programming**
 - C/C++, Assembler (x86)

Don't panic!

- ... like programming **close to the hardware**
- ... like **concurrency problems**
- ... have a certain degree of **perseverance**



Organization

- **Lecture**

(1.5h weekly, Tue 09:20–10:50, APB/E023)

- **Exercise**

(1.5h weekly, Wed 11:10–12:40, APB/E042)

- In-depth interactive discussion of lecture topics, especially technical details
- Necessary technical background for practical exercises

- **Lab**

(0–3h weekly, Tue 14:50–16:20 and/or Wed 14:50–16:20, APB/E042)

- Work on exercise tasks in groups of 2–3 students with technical support
- Hand in + discuss your solutions
(goal: maintain a working code base that doesn't break later in the semester)

Exam

- **Oral, with Prof. Schirmeier**, after the semester
 - Don't miss the **exam registration deadline** with the examination office!
 - Contact os@mailbox.tu-dresden.de for an **exam appointment** early, include:
 - Matriculation number; degree program (Master CS, Bachelor/Master Informatik, DSE, IST, ...); Module name (e.g., INF-B-510); "OSC w/ Prof. Schirmeier"; date wishes + constraints (e.g., "between X and Y")
 - for INF-BAS4 / INF-VERT4: list other lectures + examiners
 - Expect delays due to second-examiner sync. (*Komplexprüfung*), vacation period, etc.
 - **Language:** German or English (or mixture)
 - **Withdrawal** until 14 days before the appointment – make sure to deregister with selma **AND** to notify os@mailbox.tu-dresden.de.
- **Topics:** Lecture AND exercise content
- INF-PM-ANW or INF-PM-FOR, anyone?

Hybrid Teaching / Communication

- Website with lecture + exercise materials:
<https://tud.de/inf/os/studium/vorlesungen/betriebssystembau>
- Mailing list (subscribe!)
- Chat (also for you to freely use!): Matrix **#betriebssystembau:tu-dresden.de**
- **Lecture + exercise:** hybrid via BBB
 - Questions via BBB chat (presence audience: please relay!)
 - BBB-internal recordings (best effort, no guarantees)
- **Lab:** in presence; additionally online support via Matrix
- **Feedback:** in person (interrupt me!), or using above channels, or via our **Anonymous Mailbox**



Teaching Staff

- Till Smejkal
 - Lecture
 - Exercise
- Robin Thunig, Robert Walther, Tianhao Wang, Leonie Seelisch
 - Lab/Exercise
 - Technical support

Literature

- [1] A. Silberschatz, P. B. Galvin and G. Gagne. *Operating System Concepts (10th Ed.)*. Wiley, 2021.
- [2] R. Love. *Linux Kernel Development (3rd Ed.)*. Addison Wesley, 2010.
- [3] R. G. Herrtwich and G. Hommel. *Kooperation und Konkurrenz – Nebenläufige, verteilte und echtzeitabhängige Programmsysteme*. Springer-Verlag, 1989. ISBN 3-540-51701-4.
- [4] P. Yosifovich, A. Ionescu, M. E. Russinovich, D. A. Solomon. *Windows Internals: Part 1 (7th Ed.)*. Microsoft Press, 2017.
- [5] H.-P. Messmer, K. Dembowski. *PC-Hardwarebuch*. Addison-Wesley, 2003. ISBN 3-8273-2014-3.
- [6] Intel Corporation. *Intel Architecture Software Developer's Manual*. <http://www.intel.com/>

Overview

- Organization
- **Lecture Contents**
- Exercise Contents

Overview: Lectures

- L 1: Introduction
- L 2: Operating-System Development 101
- L 3: Interrupts – Hardware
- L 4: Interrupts – Software
- L 5: Interrupts – Synchronization
- L 6: Intel®64: The 32/64-Bit Intel Architecture
- L 7: Coroutines and Threads
- L 8: Scheduling
- L 9: Thread Synchronization
- L 10: Inter-process Communication
- L 11: Bus Systems
- L 12: Device Drivers

OS Development (Not Always Comfy)

- **First Steps** – *How to get your OS onto the target hardware?*
 - Compilation/Linking
 - Boot process
- **Testing and Debugging** – *What to do if your system doesn't respond?*
 - “printf debugging”
 - Emulators, virtual machines
 - Debuggers
 - Remote Debugging
 - Hardware support

Overview: Lectures

- L 1: Introduction
- L 2: Operating-System Development 101
- L 3: Interrupts – Hardware**
- L 4: Interrupts – Software
- L 5: Interrupts – Synchronization
- L 6: Intel®64: The 32/64-Bit Intel Architecture**
- L 7: Coroutines and Threads
- L 8: Scheduling
- L 9: Thread Synchronization
- L 10: Inter-process Communication
- L 11: Bus Systems**
- L 12: Device Drivers

1. An expedition through the architecture of the x86 PC

Interrupts

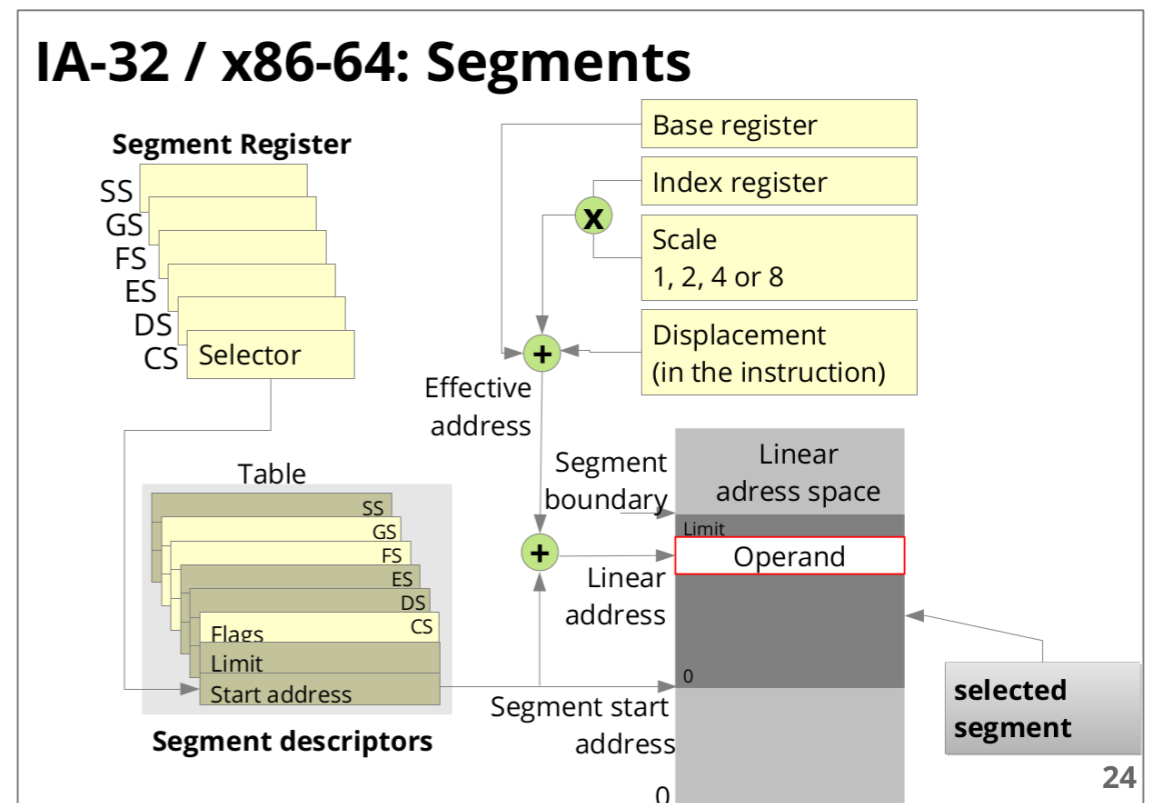
- ... in general
 - Vector tables
 - Spurious interrupts
 - Nested interrupts
- ... in the PC
 - PIC and APIC
 - Interrupts in multi-processor systems
 - IDT

1. An expedition through the architecture of the x86 PC

The Intel CPU Programming Model

- x86: History and developments
- Relics
 - 8086 Real Mode, A20 Gate
- Protected mode, protection rings
- Task model

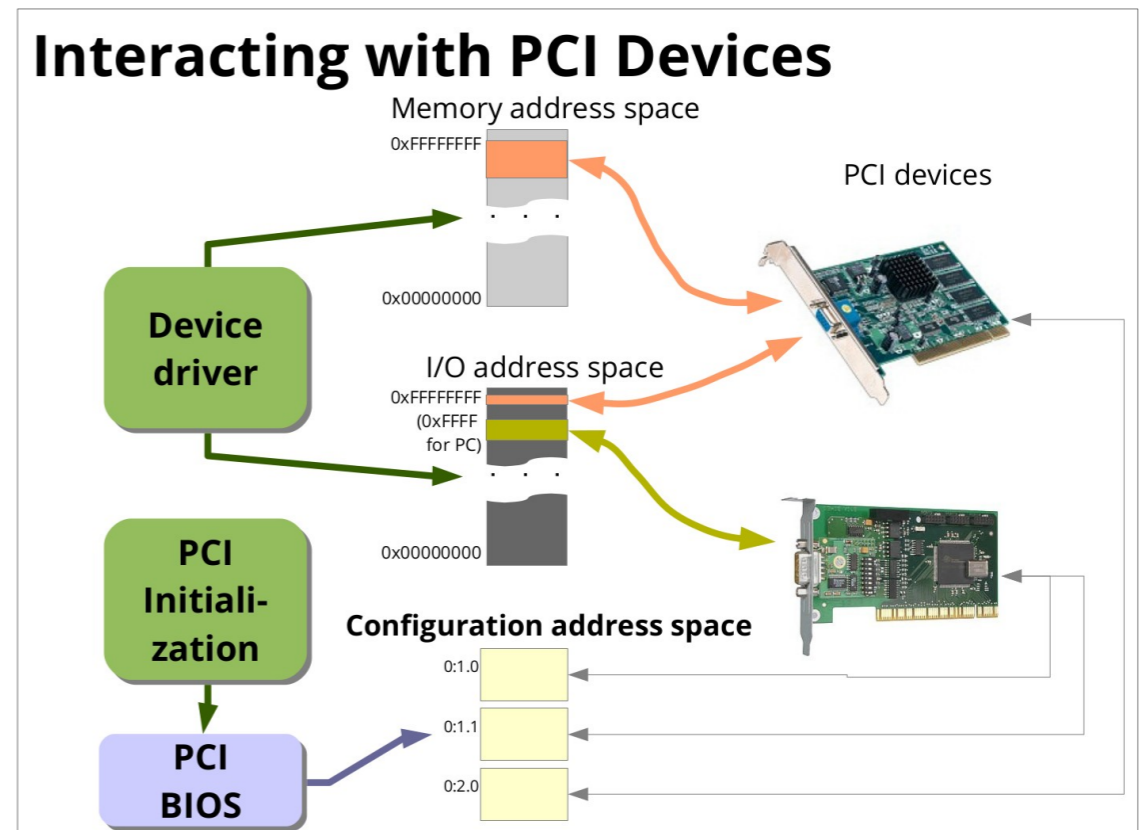
1. An expedition through the architecture of the x86 PC



PC Bus Systems

- Architecture and programming
- Local buses
 - PCI and PCI Express
 - AGP
 - AMD HyperTransport
 - Intel QPI

1. An expedition through the architecture of the x86 PC



Overview: Lectures

- L 1: Introduction
- L 2: Operating-System Development 101
- L 3: Interrupts – Hardware
- L 4: Interrupts – Software**
- L 5: Interrupts – Synchronization**
- L 6: Intel®64: The 32/64-Bit Intel Architecture
- L 7: Coroutines and Threads**
- L 8: Scheduling
- L 9: Thread Synchronization**
- L 10: Inter-process Communication**
- L 11: Bus Systems
- L 12: Device Drivers

2. Control flows and their interactions

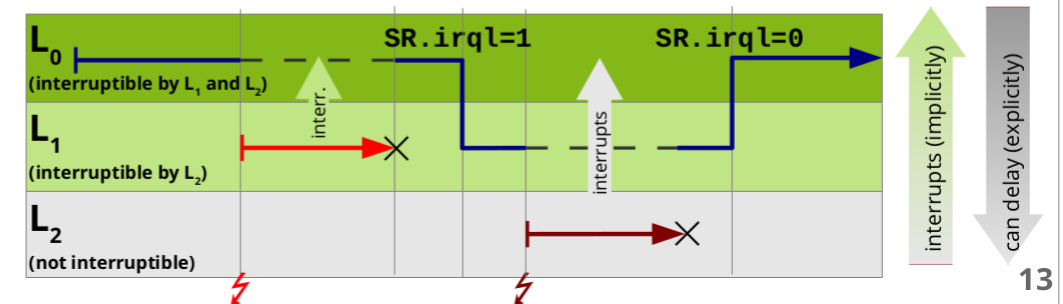
Interrupt Synchronization

- Interplay between interrupt handling and “normal” control flow
- Hardware mechanisms
 - “Hard synchronization”
- Software mechanisms
 - “Nonblocking synchronization”
 - Pro-/epilogue model
 - Interrupt transparency

2. Control flows and their interactions

Control-Flow Level Model

- Generalization to multiple interrupt levels:
 - Control flows on L_f are
 - **interrupted anytime** by control flows on L_g (for $f < g$)
 - **never interrupted** by control flows on L_e (for $e \leq f$)
 - **sequentialized** with other control flows on L_f
 - Control flows can switch levels
 - by special operations (here: modifying the status register)



Threads

- Implementing threads on x86
 - Implementing context switches
 - Basis: Coroutines
 - Preemptive scheduling
- Thread models
 - heavyweight
 - lightweight
 - featherweight

2. Control flows and their interactions

Control-Flow Level Model: **new**

- Control flows on L_f are
 - **interrupted anytime** by control flows on L_g (for $f < g$)
 - **never interrupted** by control flows on L_e (for $e \leq f$)
 - **sequentialized** with other control flows on L_f (for $f > 0$)
 - **preempted** by other control flows on L_f (**for $f = 0$**)

L_0 → Thread level

(interruptible, **preemptible**)

L_1 → Epilogue level

(interruptible, **not preemptible**)

L_2 → Interrupt level

(not interruptible, **not preemptible**)

Control flows on level L_0 (thread level) are **preemptible**.

To maintain consistency on this level, we need additional mechanisms for **thread synchronization**.

Thread Synchronization

- Blocking vs. non-blocking
- Multiprocessor thread synchronization
- Semaphore – the ultimate synchronization primitive?
- Specific problems
 - Interrelationship between synchronization and scheduling
 - Deadlocks revisited

2. Control flows and their interactions

Inter-process communication (IPC)

- Abstractions beyond semaphor and message
- Relationship between IPC and synchronization
 - real-world examples
- Duality of message-oriented and procedure-oriented systems
 - Lauer & Needham

2. Control flows and their interactions

Overview: Lectures

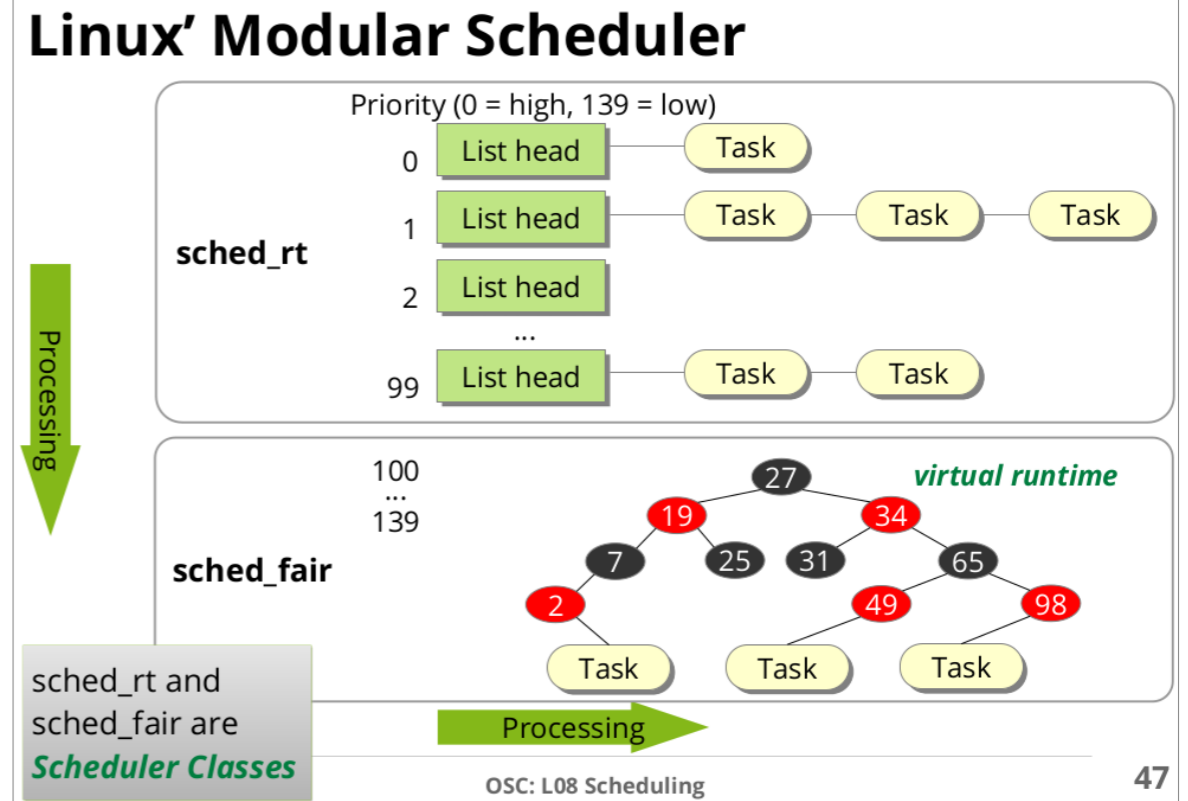
- L 1: Introduction
- L 2: Operating-System Development 101
- L 3: Interrupts – Hardware
- L 4: Interrupts – Software
- L 5: Interrupts – Synchronization
- L 6: Intel®64: The 32/64-Bit Intel Architecture
- L 7: Coroutines and Threads
- L 8: Scheduling**
- L 9: Thread Synchronization
- L 10: Inter-process Communication
- L 11: Bus Systems
- L 12: Device Drivers**

**3. OS concepts in general
and in Linux/Windows**

Scheduling

- Recapitulation, deepening
 - Basic principles and classification
- Scheduling and interrupt synchronization
- Scheduling in multiprocessor systems
- Case studies:
Linux and Windows

3. OS concepts in general and in Linux/Windows

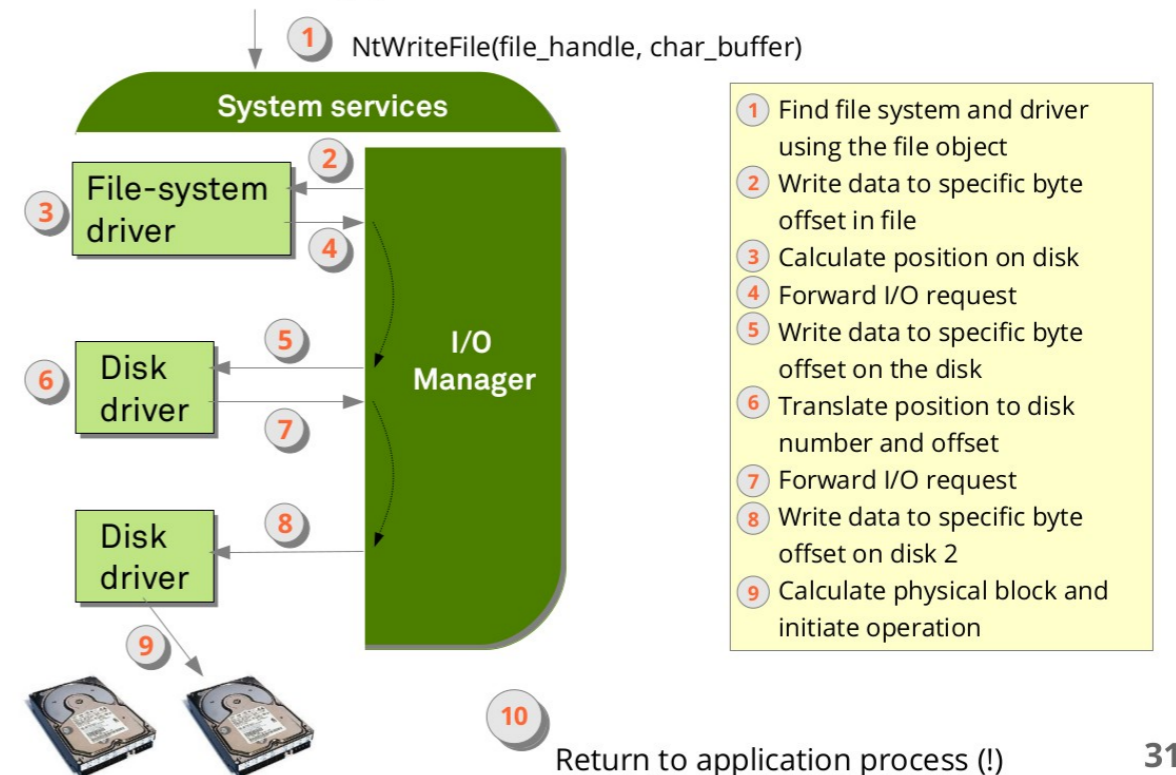


Device Programming

- Variety of typical PC devices and problems
 - Mouse, hard disk, hardware-accelerated graphics cards
- Driver models
- Real-world I/O systems
 - Windows, Linux

3. OS concepts in general and in Linux/Windows

Windows – Typical I/O Procedure

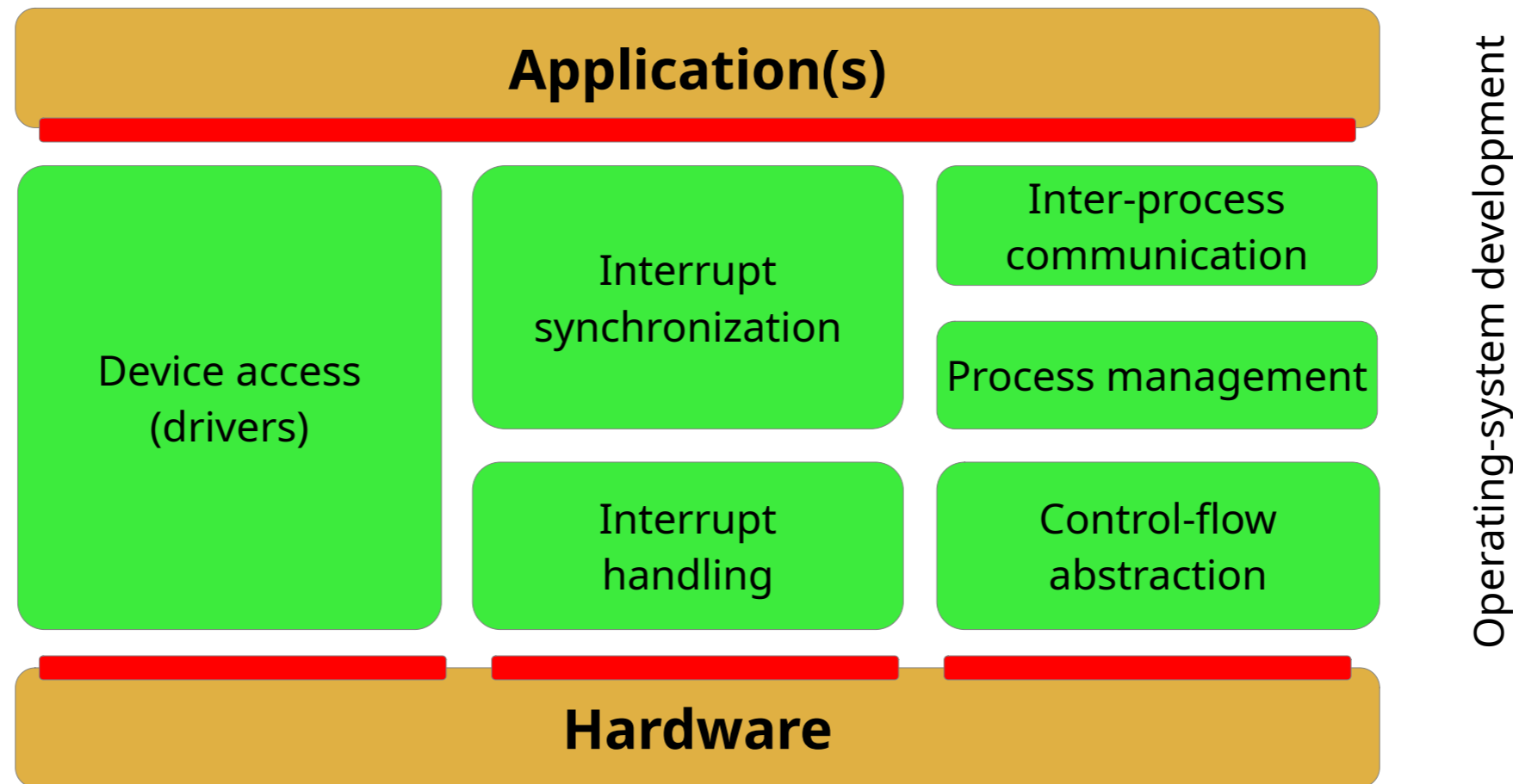


Overview

- Organization
- Lecture Contents
- **Exercise Contents**

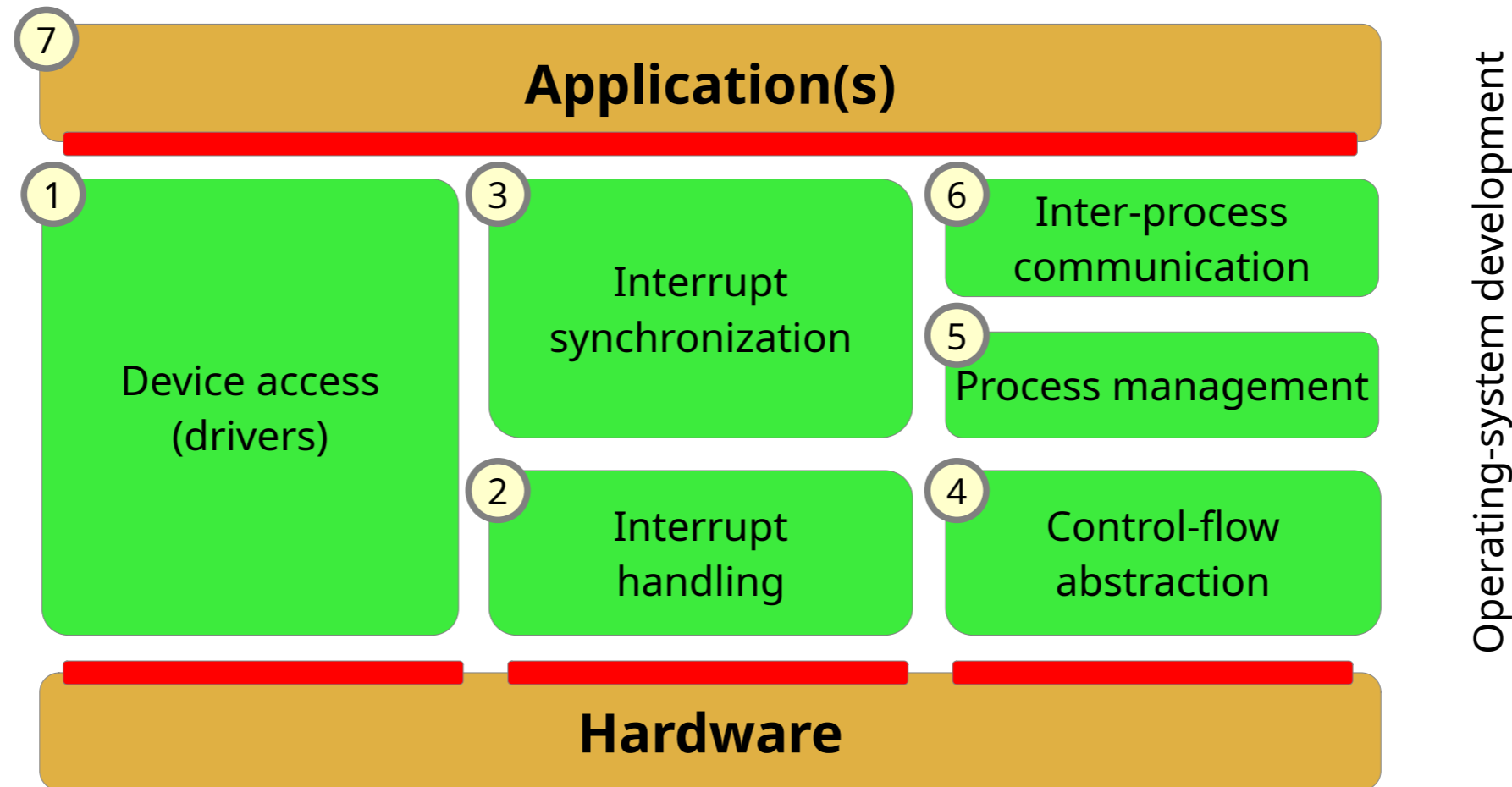
Overview: Exercise and Lab

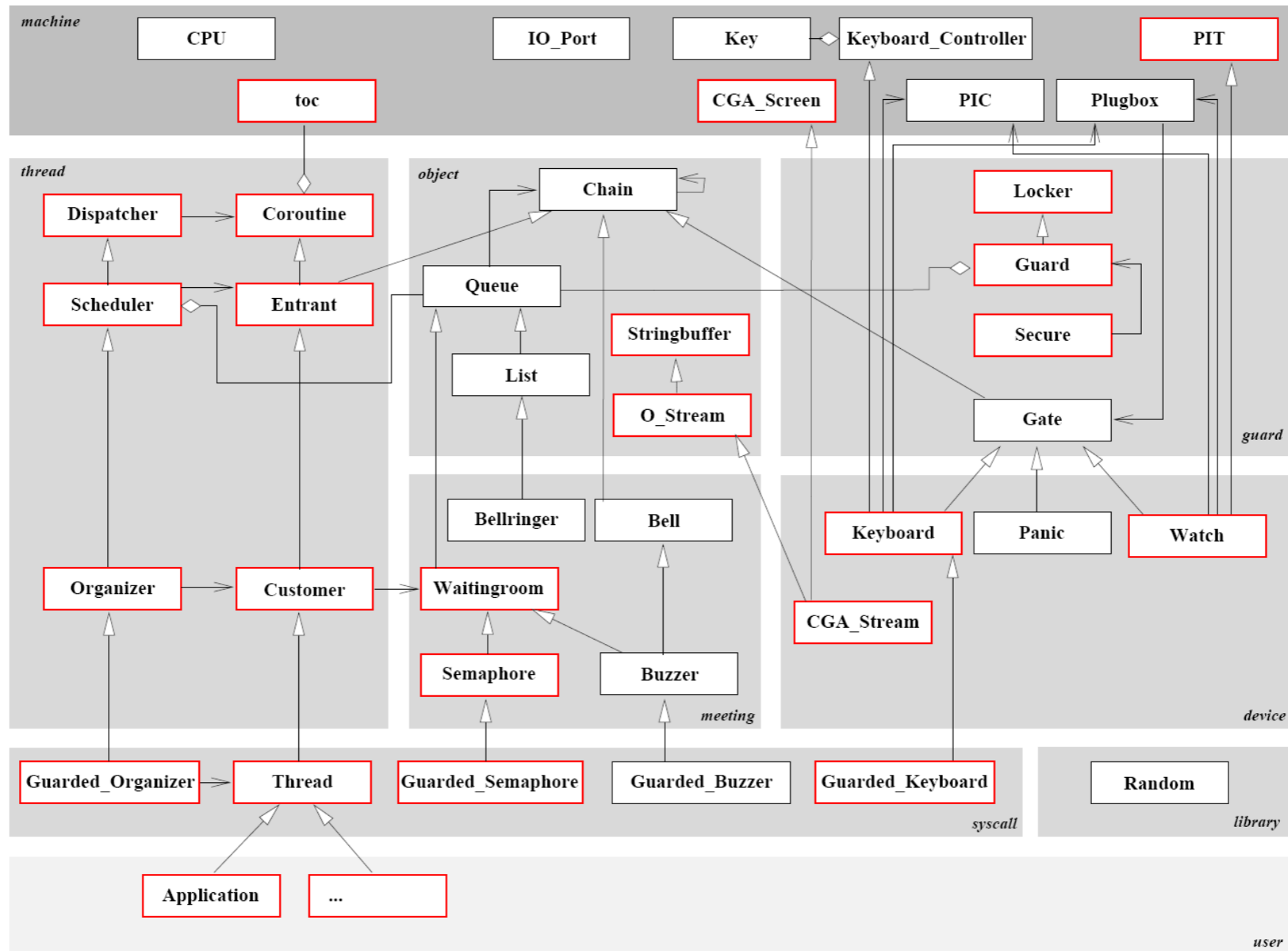
Structure of the "OO-StuBS" operating system:



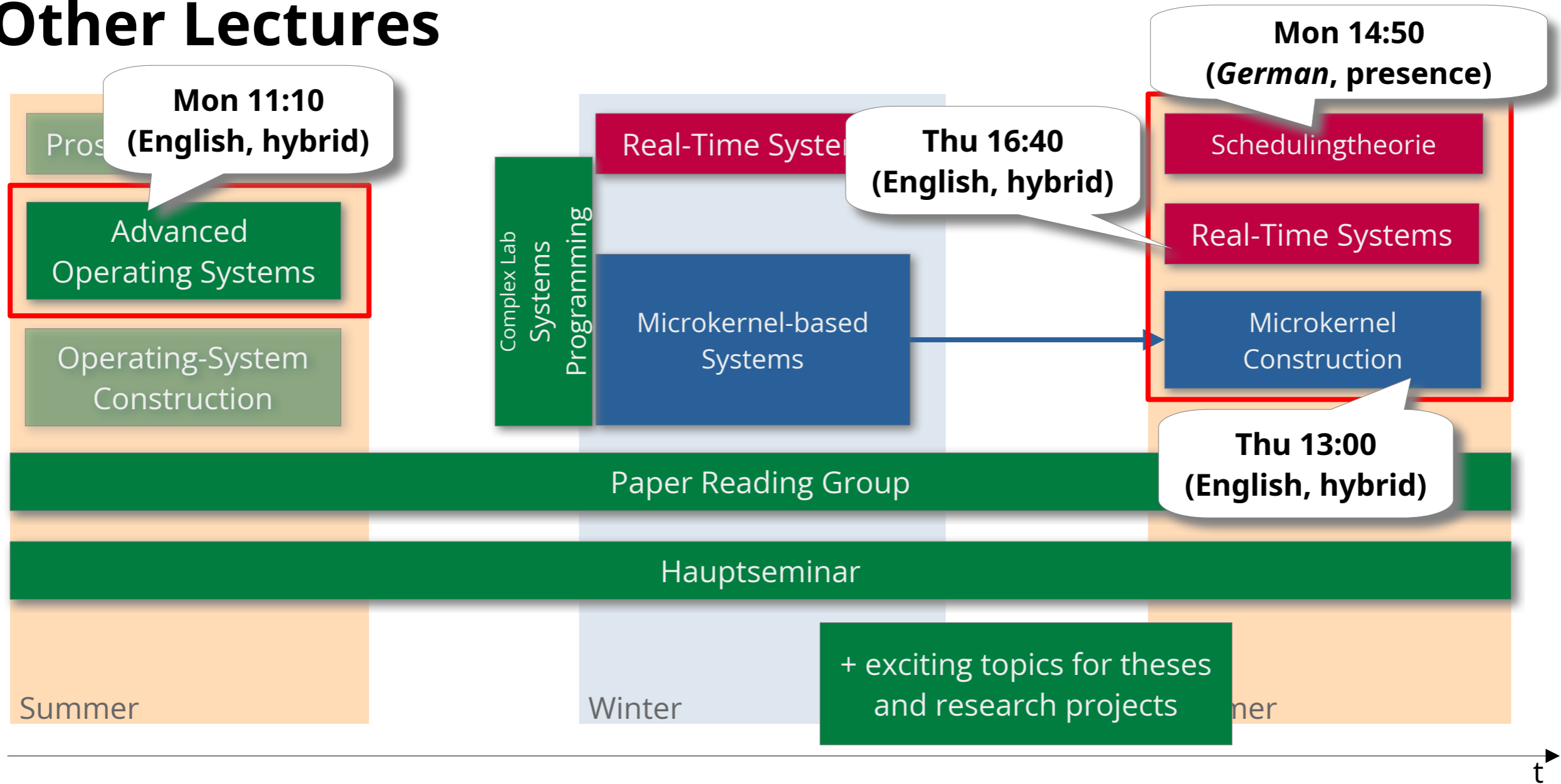
Overview: Exercise and Lab

Programming tasks:





Other Lectures



Operating-System Construction

**See you tomorrow in the first exercise
(and/or next week, here)!**