TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Systemarchitektur, Professur für Betriebssysteme

OPERATING-SYSTEM CONSTRUCTION

Material based on slides by Olaf Spinczyk, Universität Osnabrück

Introduction

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

HORST SCHIRMEIER

https://pingo.coactum.de/816711

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

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**

https://pingo.coactum.de/816711

Organization

- **Lecture**
  (1.5h weekly, Tue 11:10–12:40, APB/E005)
- **Exercise**
  (1.5h weekly, Wed 11:10–12:40, APB/E040)
  - In-depth interactive discussion of lecture topics, especially technical details
  - Necessary technical background for practical exercises
- **Lab**
  (0–3h weekly, Mon 09:20–10:50 and/or Tue 14:50–16:20, APB/E040)
  - 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, after the semester
- Lecture AND exercise content
- INF-PM-ANW or INF-PM-FOR, anyone?

https://pingo.coactum.de/816711

Hybrid Teaching / Communication

- Mailing list (subscribe!)
- Chat (also for you to freely use!): **#betriebssystembau:tu-dresden.de**
- **Lecture + exercise:** hybrid via BBB
  - Questions via BBB chat (presence audience: please relay!)
  - 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**

Feedback	via email, Matrix, or our anonymous mailbox

Teaching Staff

- Horst Schirmeier
  - Lecture
  - Exercise
- Max Kurze
  - Lab
  - Technical support

Literature

[1] A. Silberschatz and P. B. Galvin. *Operating System Concepts*. Addison-Wesley, 1994. ISBN 0-201-59292-4.

[2] R. Love. *Linux Kernel Development (2nd Ed.)*. Novell Press, 2005.

[3] R. G. Herrtwich and G. Hommel. *Kooperation und Konkurrenz - Nebenläufige, verteilte und echtzeitabhängige Programmsysteme*. Springer-Verlag, 1989. ISBN 3-540-51701-4.

[4] M. E. Russinovich and D. A. Solomon. *Microsoft Windows Internals (4th Ed.)*. Microsoft Press, 2005.

[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: Operating-System Architectures
L 10: Thread Synchronization
L 11: Inter-process Communication
L 12: Bus Systems
L 13: 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

TECHNISCHE UNIVERSITÄT DRESDEN

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: Operating-System Architectures

L 10: Thread Synchronization

L 11: Inter-process Communication

L 12: Bus Systems

L 13: Device Drivers

1. An expedition through the architecture of the x86 PC

Interrupts

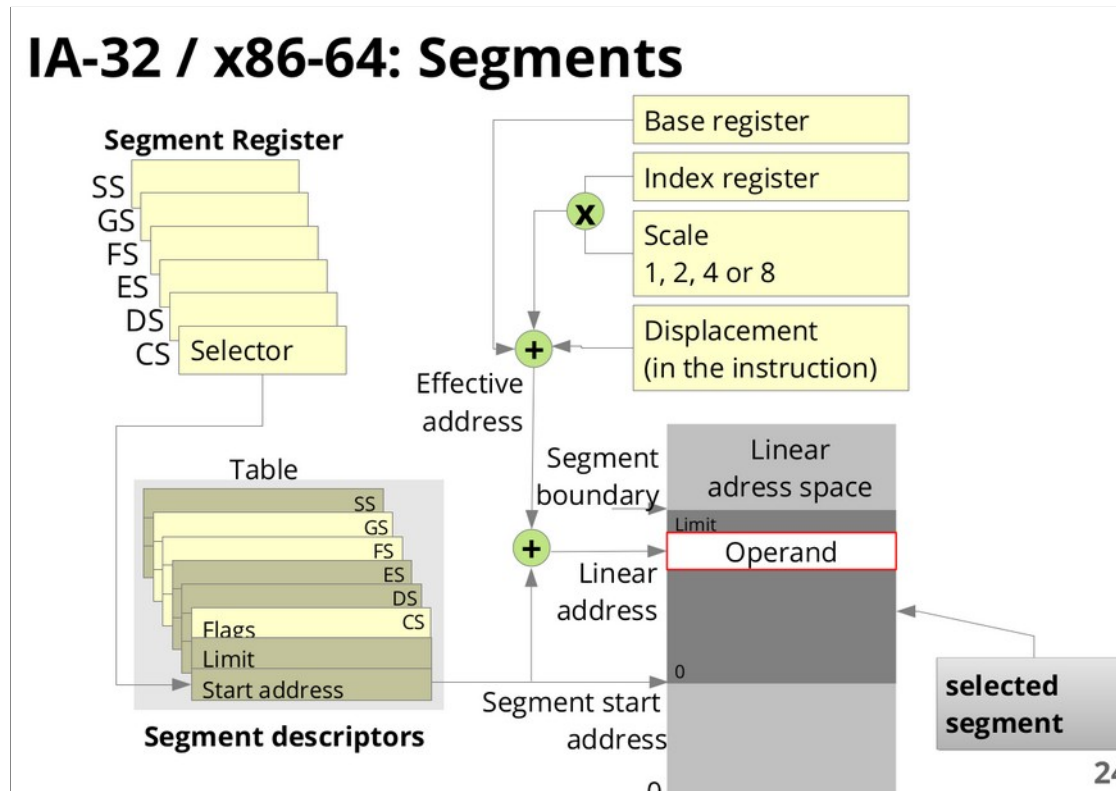
- … in general
  - Vector tables
  - Spurious interrupts
  - Nested interrupts
- … in the PC
  - PIC and APIC
  - Interrupts in multi-processor systems
  - IDT

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

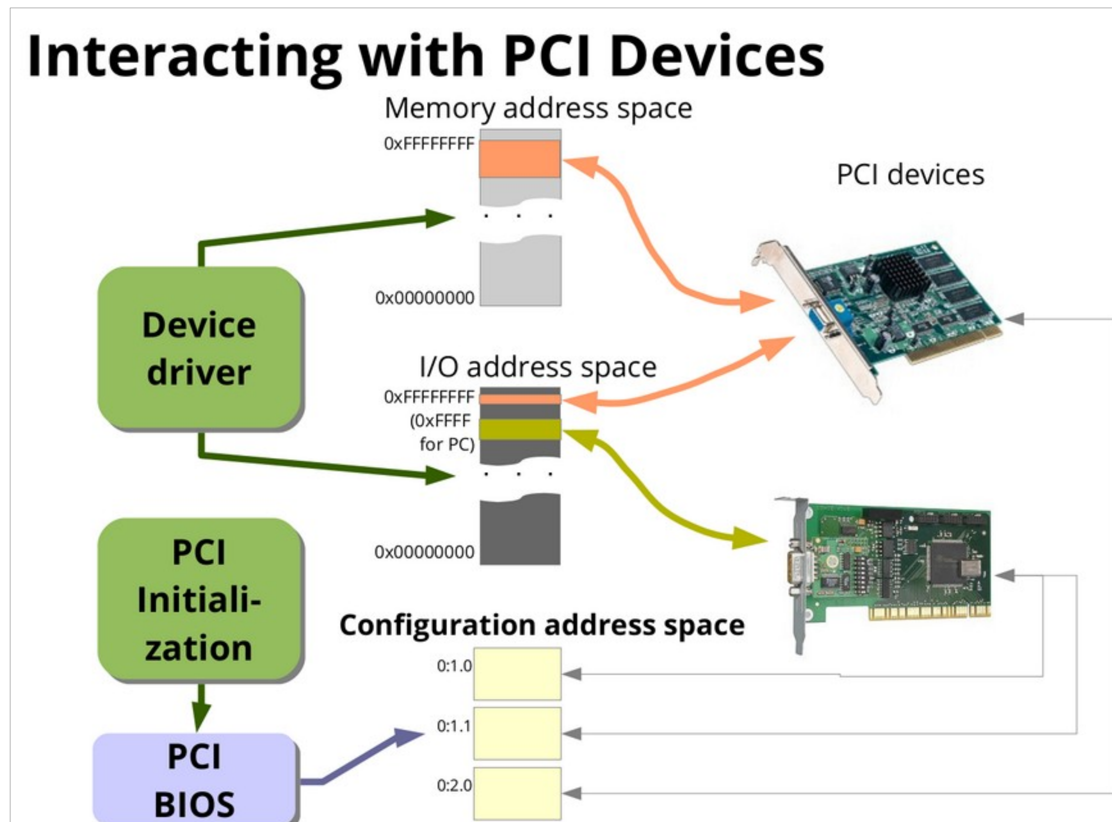
1. An expedition through the architecture of the x86 PC



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: Operating-System Architectures

L 10: Thread Synchronization

L 11: Inter-process Communication

L 12: Bus Systems

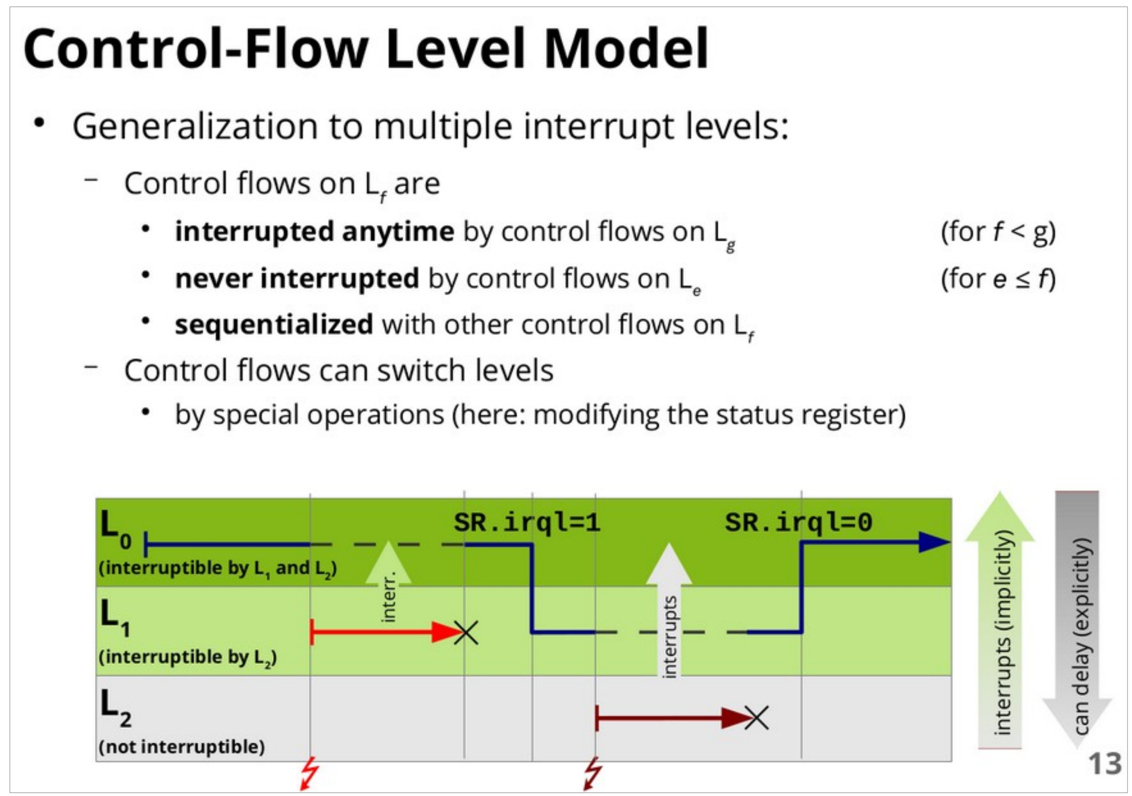
L 13: Device Drivers

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



Threads

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

2. Control flows and their interactions

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

Inter-process communication (IPC)

2. Control flows and their interactions

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

TECHNISCHE UNIVERSITÄT DRESDEN

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: Operating-System Architectures

L 10: Thread Synchronization

L 11: Inter-process Communication

L 12: Bus Systems

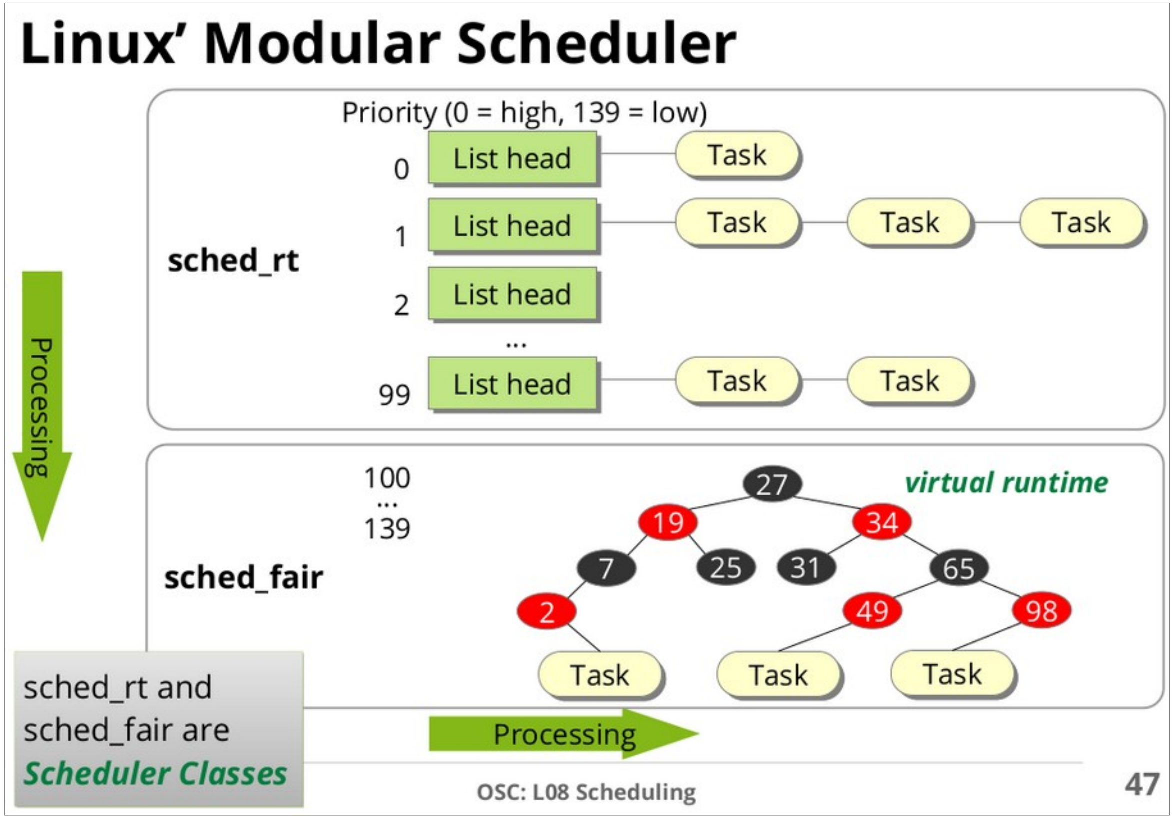
L 13: Device Drivers

3. OS concepts in general and in Linux/Windows

Scheduling

3. OS concepts in general and in Linux/Windows

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



Operating-System Architecture

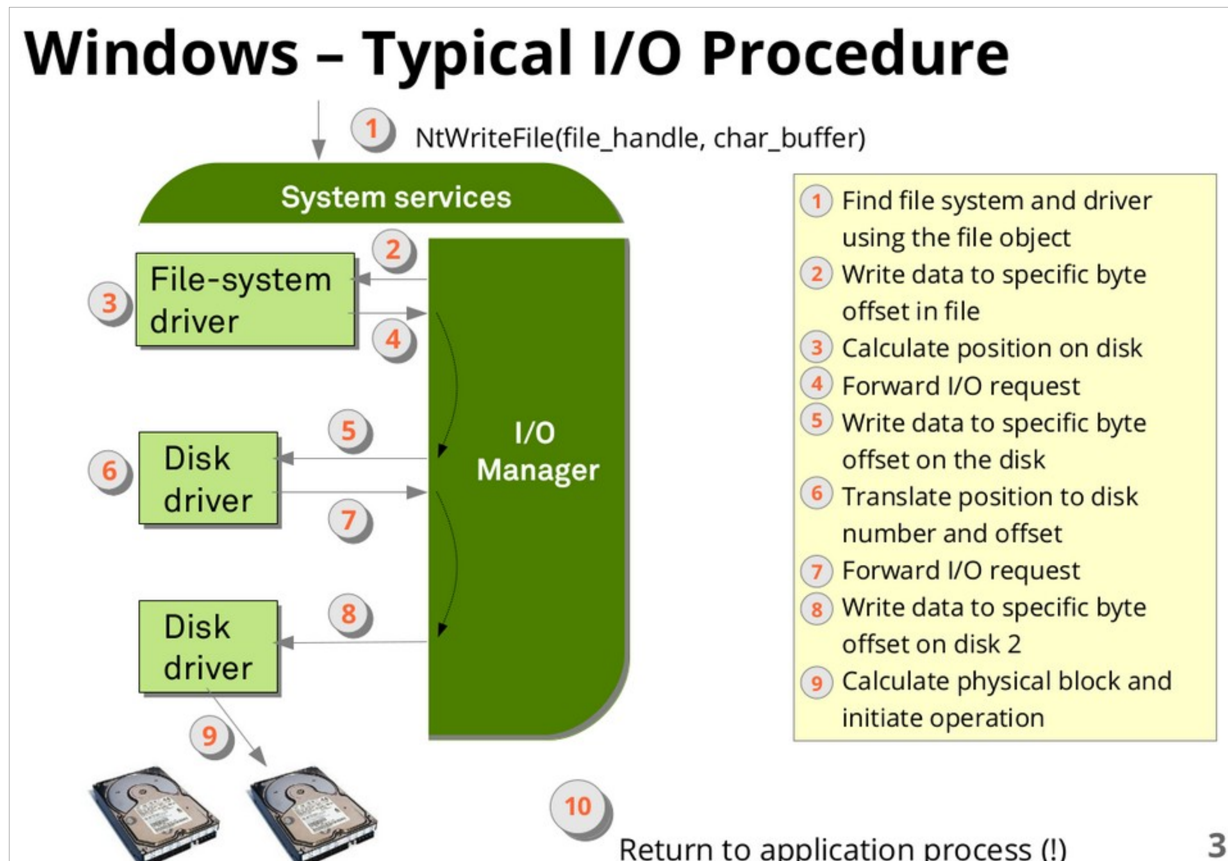
3. OS concepts in general and in Linux/Windows

- Different compositions of OS mechanisms yield different system classes.
- Microkernels, monoliths, exokernels, …
  - L4, Solaris, Linux, Windows
- Hypervisors
  - Xen, VMware

Device Programming

3. OS concepts in general and in Linux/Windows

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



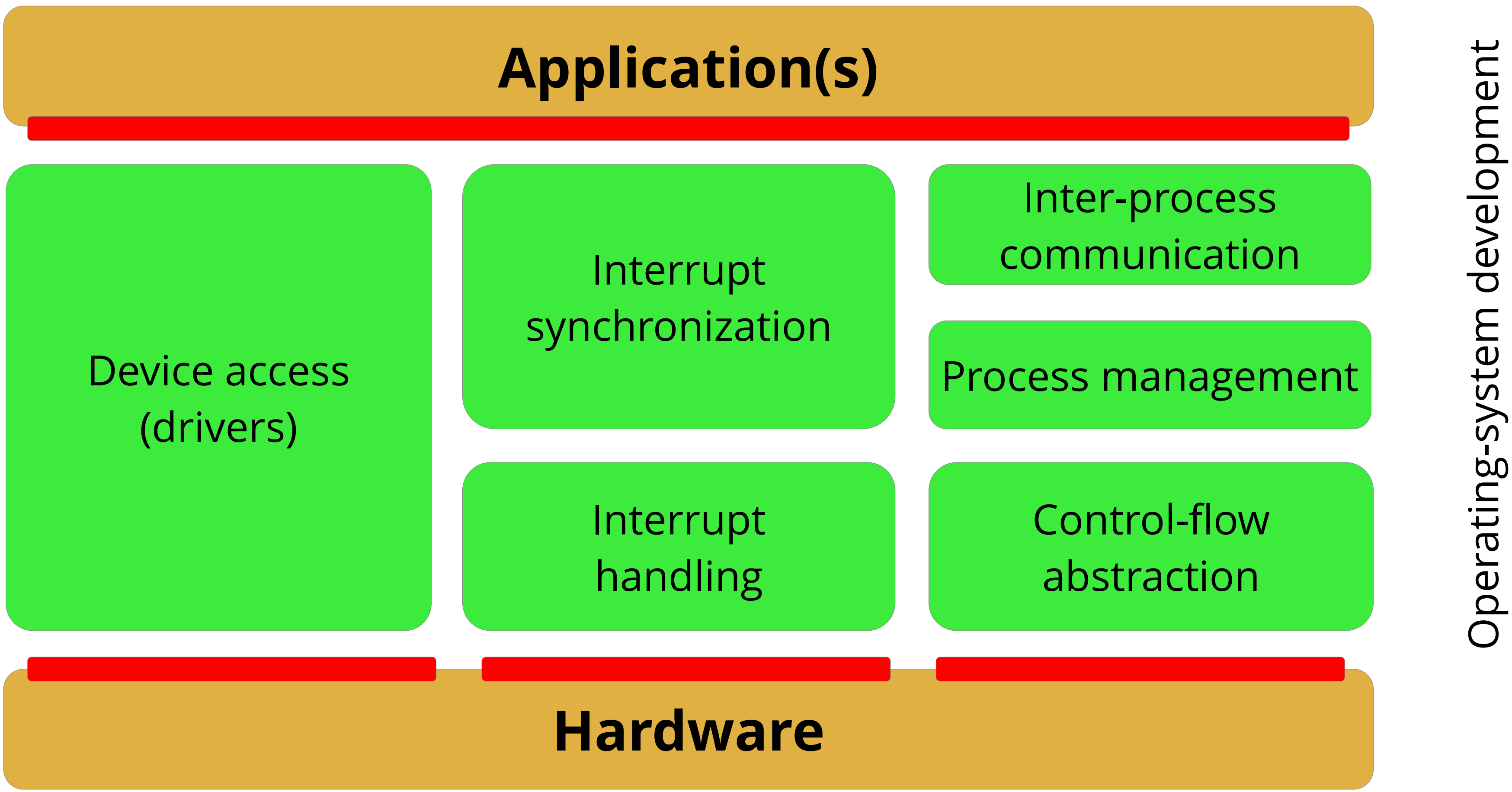
Overview

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

TECHNISCHE UNIVERSITÄT DRESDEN

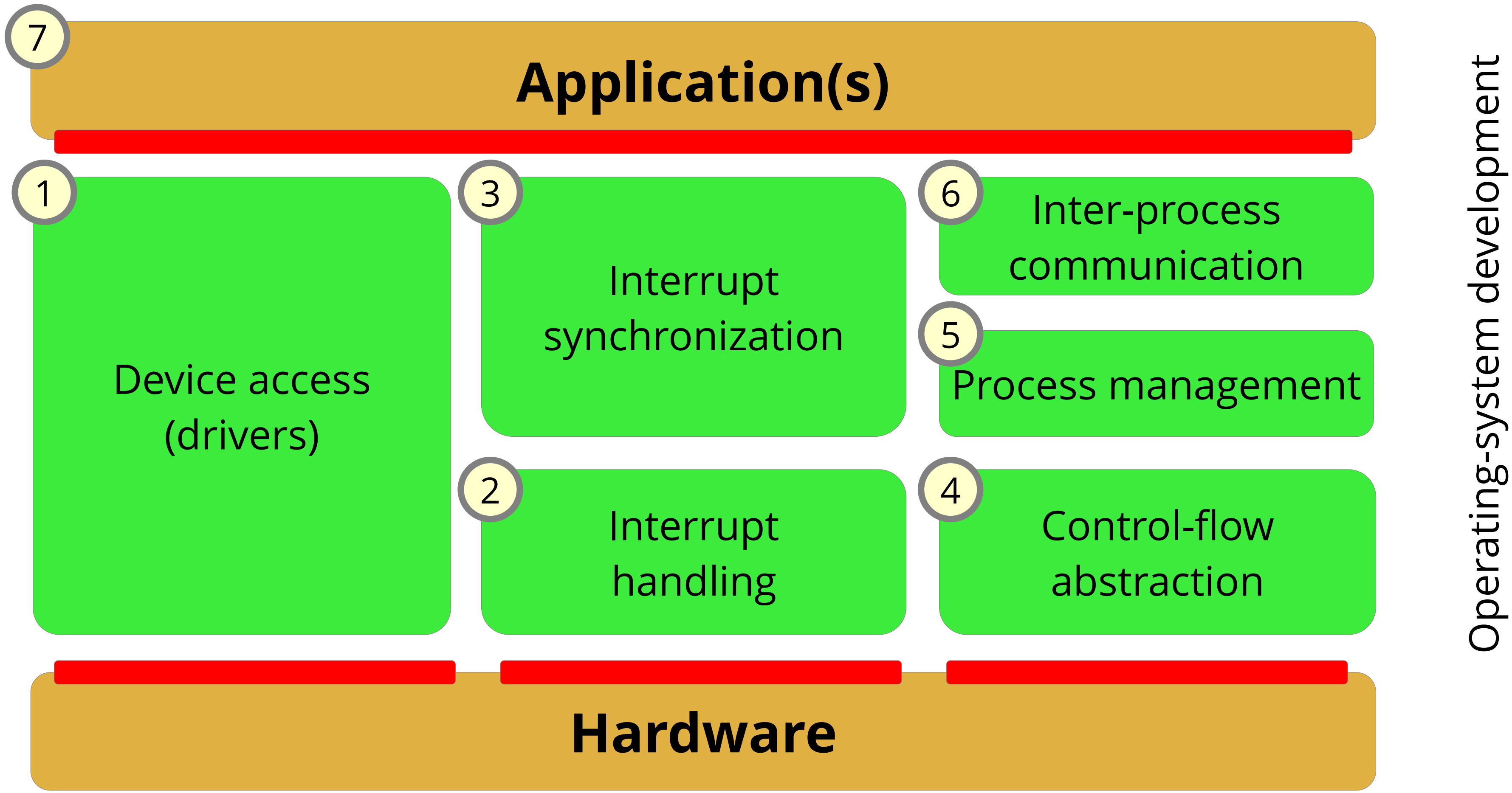
Overview: Exercise and Lab

Structure of the “OO-StuBS” operating system:

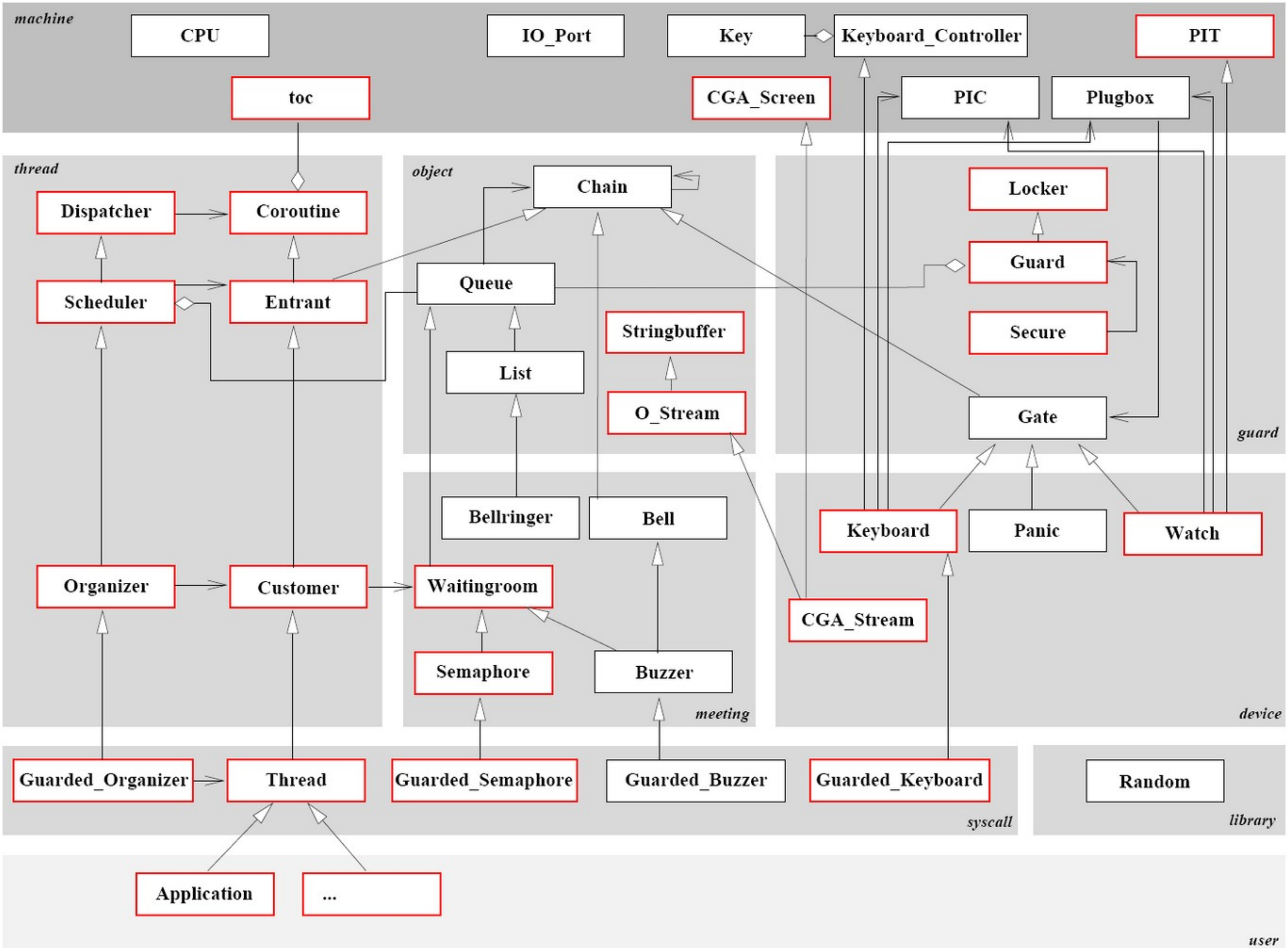


Overview: Exercise and Lab

Programming tasks:



TECHNISCHE UNIVERSITÄT DRESDEN



TECHNISCHE UNIVERSITÄT DRESDEN

Lehrveranstaltungen

Mon 11:10
(English, hybrid)

Mon 14:50
(German, presence)
(Probably) Last chance!

Thu 13:00
(English, hybrid)

Pros

Verteilte Betriebssysteme

Betriebssystem-bau

Komplexpraktikum Systemnahe Programmierung

Echtzeitsysteme

Mikrokernbasierte Betriebssysteme

Komplexpraktikum Mikrokernsysteme

Schedulingtheorie

Mikrokern-konstruktion

Paper Reading Group

Hauptseminar / EZAG

+ spannende Themen für Abschluss- und Forschungsarbeiten

Sommer

Winter

t

Operating-System Construction

**See you tomorrow
in the first exercise!**