Betriebssysteme

0. Tutorium - Setup, Basics

Peter Bohner

24. Oktober 2024

ITEC - Operating Systems Group

Über uns

Über uns

Ich

- Peter Bohner (peter.bohner@student.kit.edu)

Über uns

Ich

- Peter Bohner (peter.bohner@student.kit.edu)
- armer Student im **7.** Semester

Über uns

Ich

- Peter Bohner (peter.bohner@student.kit.edu)
- armer Student im **7.** Semester
- überraschenderweise Informatik-Student

Über uns

Ich

- Peter Bohner (peter.bohner@student.kit.edu)
- armer Student im **7.** Semester
- überraschenderweise Informatik-Student

Ihr

- ~~Name?~~
- Studiengang? Ich tippe mal Info :)
- Programmierkenntnisse Richtung C?
- Schon mal was mit Betriebssystemen gemacht?
- Linux Erfahrung?
- Aktuelle Hardware
- Erwartungen?

Organisatorisches

Vorlesung

- Übungsschein? Nein.

Organisatorisches

Vorlesung

- Übungsschein? Nein. Aber C Teil in der Klausur...

Organisatorisches

Vorlesung

- Übungsschein? Nein. Aber C Teil in der Klausur...
- Klausur: Drei Aufgaben (je 20P) mit gemischter Theorie und Praxis

Organisatorisches

Vorlesung

- Übungsschein? Nein. Aber C Teil in der Klausur...
- Klausur: Drei Aufgaben (je 20P) mit gemischter Theorie und Praxis
  Das Klausurformat hat sich letztes Jahr geändert.

Organisatorisches

Vorlesung

- Übungsschein? Nein. Aber C Teil in der Klausur...
- Klausur: Drei Aufgaben (je 20P) mit gemischter Theorie und Praxis
  Das Klausurformat hat sich letztes Jahr geändert. *Frühere Altklausuren sind anders aufgebaut!*

Organisatorisches

Vorlesung

- Übungsschein? Nein. Aber C Teil in der Klausur...
- Klausur: Drei Aufgaben (je 20P) mit gemischter Theorie und Praxis
  Das Klausurformat hat sich letztes Jahr geändert. *Frühere Altklausuren sind anders aufgebaut!*
- Übungsblätter optional (aber sinnvoll!)

Organisatorisches

Vorlesung

- Übungsschein? Nein. Aber C Teil in der Klausur...
- Klausur: Drei Aufgaben (je 20P) mit gemischter Theorie und Praxis
  Das Klausurformat hat sich letztes Jahr geändert. *Frühere Altklausuren sind anders aufgebaut!*
- Übungsblätter optional (aber sinnvoll!) *Es muss nicht jedes Blatt abgegeben werden. Bei Abgabe müssen nicht alle Aufgaben bearbeitet werden.*

Organisatorisches

Fragen Stellen (in dieser Reihenfolge)

1. im Tutorium
2. im ILIAS-Forum
3. per E-Mail an euren Tutor
4. per E-Mail an Übungsleitung
5. per E-Mail an Professor

Organisatorisches

Tutorien

- auf Deutsch / slides meist gemischt

Organisatorisches

Tutorien

- auf Deutsch / slides meist gemischt
- ab nächstem mal mit *eurem* Tutor

Organisatorisches

Tutorien

- auf Deutsch / slides meist gemischt
- ab nächstem mal mit *eurem* Tutor (nächsten Freitag ist Feiertag, verteilt euch)

Organisatorisches

Tutorien

- auf Deutsch / slides meist gemischt
- ab nächstem mal mit *eurem* Tutor (nächsten Freitag ist Feiertag, verteilt euch)
- *per du*

Organisatorisches

Tutorien

- auf Deutsch / slides meist gemischt
- ab nächstem mal mit *eurem* Tutor (nächsten Freitag ist Feiertag, verteilt euch)
- *per du*
- aktive Mitarbeit, nimmt eure Laptops mit!

OS - Tasks and Responsibilities

OS - Tasks and Responsibilities

Abstraction / Standardization

OS - Tasks and Responsibilities

Abstraction / Standardization

- Devices and how to talk to them differ greatly
  $\Rightarrow$ Remember the „Driver-CDs“ shipped with motherboards back in the days so you could properly configure and run things?

OS - Tasks and Responsibilities

Abstraction / Standardization

- Devices and how to talk to them differ greatly
  ⇒ Remember the „Driver-CDs“ shipped with motherboards back in the days so you could properly configure and run things?
- Strange user demands: Programs *should* run on more than one hardware configuration

OS - Tasks and Responsibilities

Abstraction / Standardization

- Devices and how to talk to them differ greatly
  ⇒ Remember the „Driver-CDs“ shipped with motherboards back in the days so you could properly configure and run things?
- Strange user demands: Programs *should* run on more than one hardware configuration

⇒ Abstract away hardware details!

Operating System – Tasks and Responsibilities

Resource management

Operating System – Tasks and Responsibilities

Resource management

- You want to print? Too bad, another program is already using that printer.

Operating System – Tasks and Responsibilities

Resource management

- You want to print? Too bad, another program is already using that printer.
- You want to access the storage drive? Too bad, another program is already doing that.

Operating System – Tasks and Responsibilities

Resource management

- You want to print? Too bad, another program is already using that printer.
- You want to access the storage drive? Too bad, another program is already doing that.
- You want to get CPU time? Too bad, this `while (true)` loop is more important.
- ...

Operating System – Tasks and Responsibilities

Security and Protection

Operating System – Tasks and Responsibilities

Security and Protection

- You are a good citizen and use a password manager. What could happen without your helpful OS?

Operating System – Tasks and Responsibilities

Security and Protection

- You are a good citizen and use a password manager. What could happen without your helpful OS? ⇒ Other programs may read its memory!

Operating System – Tasks and Responsibilities

Security and Protection

- You are a good citizen and use a password manager. What could happen without your helpful OS? ⇒ Other programs may read its memory!
- You copied a password to your clipboard?

Operating System – Tasks and Responsibilities

Security and Protection

- You are a good citizen and use a password manager. What could happen without your helpful OS? ⇒ Other programs may read its memory!
- You copied a password to your clipboard? ⇒ Ooops. You're on your own there. The Clipboard is not provided by the OS and mostly has no special protections.

Operating System – Tasks and Responsibilities

Security and Protection

- You are a good citizen and use a password manager. What could happen without your helpful OS? ⇒ Other programs may read its memory!
- You copied a password to your clipboard? ⇒ Ooops. You’re on your own there. The Clipboard is not provided by the OS and mostly has no special protections.
- You write a cool little program that fills a buffer with a random value. Sadly you made a mistake and missed a bounds check. What happens?

Operating System – Tasks and Responsibilities

Security and Protection

- You are a good citizen and use a password manager. What could happen without your helpful OS? ⇒ Other programs may read its memory!
- You copied a password to your clipboard? ⇒ Ooops. You're on your own there. The Clipboard is not provided by the OS and mostly has no special protections.
- You write a cool little program that fills a buffer with a random value. Sadly you made a mistake and missed a bounds check. What happens? You crash, but your text editor doesn't suddenly have its memory overwritten!
- ...

Operating System – Tasks and Responsibilities

Provide an execution environment for applications

Operating System – Tasks and Responsibilities

Provide an execution environment for applications

- What does that mean?

Operating System – Tasks and Responsibilities

Provide an execution environment for applications

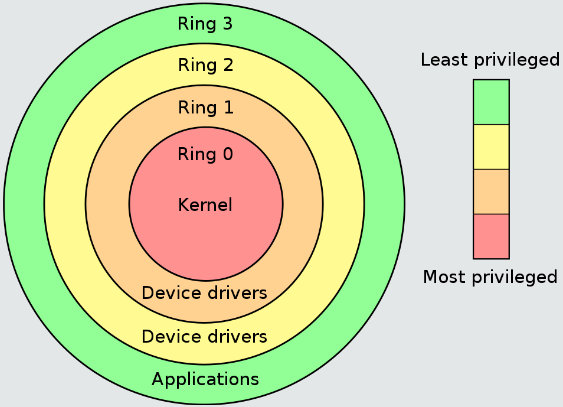
- What does that mean? Basically all of the above combined and more. Make a homely place where applications like to live!

Kernel and user mode

What are the differences between a processor running in kernel or user mode? Why are both modes needed?

Kernel and user mode

What are the differences between a processor running in kernel or user mode? Why are both modes needed?



Wikipedia - Hertzsprung

Kernel and user mode

What are the differences between a processor running in kernel or user mode? Why are both modes needed?

Kernel and user mode

What are the differences between a processor running in kernel or user mode? Why are both modes needed?

- In kernel mode you have full access to privileged instructions.

Kernel and user mode

Great, but what is a privileged instruction?

Kernel and user mode

Great, but what is a privileged instruction?

- Change control registers for memory mappings ⇒ Read other process's memory

Kernel and user mode

Great, but what is a privileged instruction?

- Change control registers for memory mappings ⇒ Read other process's memory
- Disable / Enable interrupts ⇒ No preemption for you.

Kernel and user mode

Great, but what is a privileged instruction?

- Change control registers for memory mappings $\Rightarrow$ Read other process's memory
- Disable / Enable interrupts $\Rightarrow$ No preemption for you.
- Access platform devices (network card, storage, printer,...)

Kernel and user mode

Great, but what is a privileged instruction?

- Change control registers for memory mappings ⇒ Read other process’s memory
- Disable / Enable interrupts ⇒ No preemption for you.
- Access platform devices (network card, storage, printer,...)
- Some nice registers: `LGDT` (Load Global Descriptor Table) or the `LLDT`, `INVD` (Invalidate cache), `HLT` (Halt processor!)

Kernel and user mode

Great, but what is a privileged instruction?

- Change control registers for memory mappings ⇒ Read other process's memory
- Disable / Enable interrupts ⇒ No preemption for you.
- Access platform devices (network card, storage, printer,...)
- Some nice registers: `LGDT` (Load Global Descriptor Table) or the `LLDT`, `INVD` (Invalidate cache), `HLT` (Halt processor!)
- Is `MOV` (Move) a priviledged instruction?

Kernel and user mode

Great, but what is a privileged instruction?

- Change control registers for memory mappings ⇒ Read other process's memory
- Disable / Enable interrupts ⇒ No preemption for you.
- Access platform devices (network card, storage, printer,...)
- Some nice registers: `LGDT` (Load Global Descriptor Table) or the `LLDT`, `INVD` (Invalidate cache), `HLT` (Halt processor!)
- Is `MOV` (Move) a priviledged instruction? Yes, if moving to debug/control registers!

ABI

What does that acronym even mean?

ABI

What does that acronym even mean?

Application Binary Interface

ABI

What does that acronym even mean?

Application **B**inary **I**nterface

And what does that specify?

ABI

What does that acronym even mean?

Application Binary Interface

And what does that specify?

- Interface of *binary* programs (i.e. *after* compilation)

ABI

What does that acronym even mean?

Application **B**inary **I**nterface

And what does that specify?

- Interface of *binary* programs (i.e. *after* compilation)
- Instruction Set (e.g. x86, ARM)

ABI

What does that acronym even mean?

Application Binary Interface

And what does that specify?

- Interface of *binary* programs (i.e. *after* compilation)
- Instruction Set (e.g. x86, ARM)
- Calling convention (e.g. `cdecl` or `System V AMD64 ABI`)

ABI

What does that acronym even mean?

Application Binary Interface

And what does that specify?

- Interface of *binary* programs (i.e. *after* compilation)
- Instruction Set (e.g. x86, ARM)
- Calling convention (e.g. `cdecl` or `System V AMD64 ABI`)
- Basic data types and their size / alignment (`int`, `sizeof(int)`)

ABI

What does that acronym even mean?

Application **B**inary **I**nterface

And what does that specify?

- Interface of *binary* programs (i.e. *after* compilation)
- Instruction Set (e.g. x86, ARM)
- Calling convention (e.g. `cdecl` or `System V AMD64 ABI`)
- Basic data types and their size / alignment (`int`, `sizeof(int)`)
- How to perform System Calls

ABI

You might also know the term „ABI of a library“?

ABI

You might also know the term „ABI of a library“?

This is used when talking about *binary compatibility* of different library versions.

⇒ Do you need to recompile your code against the new version?

ABI

What's the usual calling convention on modern Linux?

ABI

What's the usual calling convention on modern Linux?

`System V AMD64 ABI`

`https://godbolt.org/z/68xexn`

- Integer arguments in `rdi`, `rsi`, `rdx`, `rcx`, `r8`, `r9`, then stack
- FP arguments in `xmm0` to `xmm7`
- Integer return value in `rax`, `rdx`

Installation

Installation

x86_64 linux - install these packages

- gcc (nicht clang!), make, bash, git, gdb
- a text editor of your choice

MacOS - you are probably not afraid of your terminal either

- install UTM, Vagrant or qemu-system-x86_64 (homebrew)
- install a x86_64 (not ARM!) Linux guest `https://docs.getutm.app/guides/debian/`
- → Linux
- (optional: shared folder)

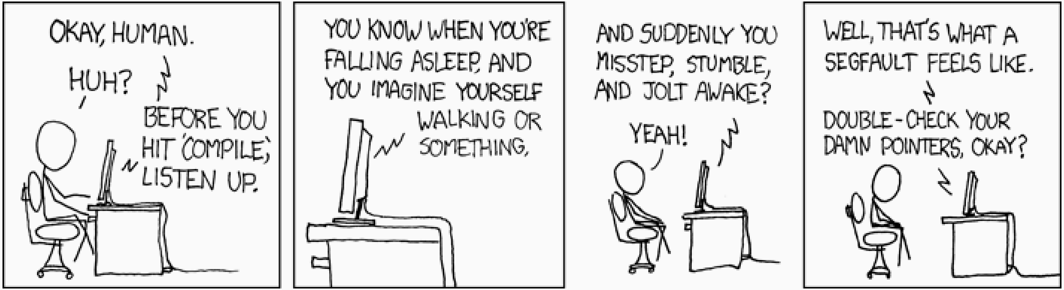
Installation

Windows

- install WSL2 (you may need to be on *KA-WLAN* instead of *KIT* wifi, because *$REASONS*) (admin Powershell: *wsl –install*)
- `https://learn.microsoft.com/en-us/windows/wsl/install`
- oder: `https://learn.microsoft.com/en-us/windows/wsl/install-manual`
- install a text editor (z.B. vscode, Notepad++)
- Ubuntu-shell: ∼ $ *sudo apt update && sudo apt install build-essential git gdb*

or: use ATIS ssh server

FRAGEN?


OKAY, HUMAN.
HUH?
BEFORE YOU HIT 'COMPILE,' LISTEN UP.
YOU KNOW WHEN YOU'RE FALLING ASLEEP, AND YOU IMAGINE YOURSELF WALKING OR SOMETHING,
AND SUDDENLY YOU MISSTEP, STUMBLE, AND JOLT AWAKE?
YEAH!
WELL, THAT'S WHAT A SEGFAULT FEELS LIKE.
DOUBLE-CHECK YOUR DAMN POINTERS, OKAY?

XKCD 372 - Compiler Complaint

Bis nächste Woche :)