






# OS Motivation

([Usage hints](#) for this presentation)

Computer Structures and Operating Systems 2023  
Dr. Jens Lechtenbörger ([License Information](#))

**Data Science: Machine Learning and Data Engineering (Prof. Gieseke)**  
Dept. of Information Systems  
WWU Münster, Germany

## Speaker notes

- To toggle these notes, press `v`
  - If a slide contains audio, notes might show transcript
- Press `?` for key bindings (in particular, `a`, `o`, `n`, `p`, `Ctrl-Shift-f`)
- Presentations support two different PDF formats, see [usage notes](#) 
  - Both hyperlinked on index page
    - Concise PDF format (replace `.html` and whatever follows in [address bar](#)  with `.pdf`)
    - Print browser view to PDF (add `?print-pdf` after `.html`, then print to PDF; [suggested settings](#) )
- If you find the amount of outgoing links to be distracting, see [usage notes](#) 
  - Add `?hidelinks` (maybe with a number) after `.html`
- See [usage notes](#)  for other non-obvious features

# Agenda

- 1. Introduction
- 2. Choices and Consequences
- 3. Software
- 4. Conclusions

# 1. Introduction

# 1.1. Learning Objectives

- Discuss role of OSs to control computers
  - Performance aspects
  - Ownership and control
- Discuss role of licenses to control software

## 1.2. Performance Aspects

- OS manages computer's resources: CPU, memory, I/O
- OS understanding helps to identify and reason about resource bottlenecks.
  - Improve design, analysis, and implementation of information systems.
  - E.g., why is my computer/application slow? How to improve that?
- (Above are “traditional” topics for OS courses; this presentation takes a different direction.)

# 1.3. Whom do Computers Obey?

- Recall (long-term) goal of CSOS
  - “Play and experiment with and control any computer, at any level of interest”
- In CS part, you learned how to build a computer.
  - How does that feel?
- In OS part, we investigate Operating Systems (OSs) to control computers.
  - OS controls what is executing when.
  - Who controls the OS?
  - Who controls the computer, then?



“Notebook”  
under CC0 1.0;  
from Pixabay



“Search” under CC0 1.0;  
from Pixabay

# 1.4. Computers



“Server” under CC0 1.0;  
cropped from Pixabay



“C64” under CC0 1.0;  
from Pixabay



“Notebook” under CC0  
1.0; from Pixabay



“Router” under CC0  
1.0; from Pixabay



“Car” under CC0 1.0;  
from Pixabay



“Doll” under CC0 1.0;  
from Pixabay



“TV” under CC0 1.0;  
from Pixabay



“Smart watch” under  
CC0 1.0; cropped from  
Pixabay



“Fairphone 2” by  
Kaihsu Tai under CC  
BY-SA 4.0; cropped  
from Wikimedia  
Commons



# 1.5. Enlightenment

*Enlightenment is man's emergence from his self-incurred immaturity. ☒ (Immanuel Kant)*

- Your computer does whatever it is programmed to do.
- You can succumb to somebody else's programming.
- Or emerge.

# 2. Choices and Consequences

# 2.1. Impact of Choices

- **Economic** and **ecological/social** impact

- What do you buy/use/create?
  - What do you advertise?
  - What do you impose on others?

- **Economic** impact

- Should be obvious

- **Ecological/social** impact

- If you use some social/communication service, you increase its appeal and value
  - **Network effects** ↗, feedback loops, externalities
- If you **choose** communication under **surveillance capitalism** ↗, you **impose** surveillance on your family, friends, colleagues ...



“Man with megaphone”  
under CC0 1.0; from  
Pixabay



“Figure” under CC0 1.0;  
from Pixabay

## 2.2. Rationale

- If you want to **own** a device containing a computer (“smart” anything, recall [☒ slide on computers](#)), you need to control its software.
  - (Including firmware and underlying hardware)
    - (Topic in its own right)
- Otherwise,
  - device may work against you, see [IoT Hall of Shame](#) [↗](#),
    - e.g., [smartphones](#) [↗](#) and [diesel engines](#) [↗](#),
  - device may stop working any time,
    - e.g., useless fitness tracker ([2019](#) [↗](#)), bricked smart home [entertainment device](#) ([2017](#)) [↗](#) and [hub](#) ([2016](#)) [↗](#), [Orwell books removed from Kindle](#) ([2009](#)) [↗](#),
  - device may stop receiving (security) updates any time.



“Control” under CC0 1.0; from Pixabay

# 3. Software

# 3.1. Free Software

- Free Software = Free/Libre and Open Source Software (FLOSS) [↗](#)
  - Term coined by Richard Stallman, [Free Software Foundation](#) [↗](#)
    - [Neal Stephenson, 1999](#) [↗](#): “The most hackerish of all the hackers, the Ur-hacker as it were, was and is Richard Stallman”
  - “Free” as in free speech or freedom, not free beer
- [Four freedoms](#) [↗](#)
  1. Run software (also changed versions)
  2. Study software
  3. Redistribute copies
  4. Distribute modified versions



“Photo of Richard Stallman” by Victor Powell under CC BY-SA 3.0 - from

## 3.2. Free vs Open Source

- Whether software is free (libre) or open source (or both, FOSS, FLOSS) or something else, depends on the **license**
  - 1986: Free software defined
    - See [GNU's Bulletin 1](#) for origin
  - 1998: [Open Source Software \(OSS\)](#) defined
  - See [Comparison of free and open-source software licenses](#)
    - Well-known FLOSS licenses include Apache License, GNU Public License, Eclipse Public License, Mozilla Public License
- Mostly philosophical distinction, sometimes religious
  - I prefer the term “free software” because it emphasizes freedom
  - If you do not want **your** academic works (theses, code, project results) to disappear, publish them under free licenses

## 3.3. GNU/Linux

- [Linux](#) is a [free OS kernel](#)
- GNU/Linux is a family of [free OSs](#)
  - [Naming controversy](#)
  - I encourage you to try out GNU/Linux, which is free software,
    - the major OS for cloud infrastructures,
    - the server OS in lots of **project seminars**.
  - Upcoming **assignments** are based on GNU/Linux
    - Several GNU/Linux distributions can be started as live systems from CD/DVD or USB stick without changing your current installation
      - [Help on getting started](#)



“Tux, the Linux mascot” under CC0 1.0; from Wikimedia Commons



## 3.4. Firmware

- Firmware = Software that is embedded in hardware by vendor
  - Stored in (EP)ROM, flash
  - Initialization and control of hardware
    - E.g., [BIOS](#), [\(U\)EFI](#); but also video BIOS of graphics card, Management Engine of Intel CPUs
      - Typically shipped as binary blobs
      - [NSA ANT catalog](#) also contains firmware trojans (in wake of Snowden revelations)

## 3.4.1. CPU Rings

- Outlook: CPUs have [rings/privilege levels](#)
  - Instruction set restricted depending on ring
    - Ring 3: User programs (I/O and memory access restricted)
    - Ring 2, 1: Usually unused (originally for system services and device drivers)
    - Ring 0: OS kernel (traditionally, ring 0 was most privileged)
    - More rings [[Fra19](#)]
      - Ring -1: Hypervisor ([virtual machine monitor](#))
      - Ring -2: System management mode (SMM), unified extensible firmware interface (UEFI)
      - Ring -3: Management Engine

## 3.4.2. Free and Open Firmware

- Articles by Jessie Frazelle

- [Fra19]

- “Between Ring -2 and Ring -3 there are at least two and a half other kernels in our stack that have many capabilities. Each of these kernels has its own networking stacks and web servers, which is unnecessary and potentially dangerous, especially if you do not want these rings reaching out over the network to update themselves. The code can also modify itself and persist across power cycles and reinstalls. There is very little visibility into what the code in these rings is actually doing, which is horrifying, considering these rings have the most privileges.”

- [Fra20a]

- “It’s an alarming problem that the code running with the most privilege has the least visibility and inspectability.”

- [Fra20b]

- “If you would like to help with the open source firmware movement, push back on your vendors and platforms you are using to make their firmware open source.”

# 4. Conclusions



# Bibliography

[Fra19] Frazelle, Open Source Firmware, Commun. ACM 62(10), 34-38 (2019).

<https://doi.org/10.1145/3343042> ↗

[Fra20a] Frazelle, Opening up the Baseboard Management Controller, Commun. ACM 63(2), 38-40 (2020). <https://doi.org/10.1145/3369758> ↗

[Fra20b] Frazelle, Securing the Boot Process, Commun. ACM 63(3), 38-42 (2020).

<https://doi.org/10.1145/3379512> ↗

# License Information

This document is part of an [Open Educational Resource \(OER\)](#) course on Operating Systems. [Source code and source files](#) are available on [GitLab](#) under [free licenses](#).

Except where otherwise noted, the work “OS Motivation”, © 2018-2023 [Jens Lechtenbörger](#), is published under the [Creative Commons license CC BY-SA 4.0](#).

*No warranties are given. The license may not give you all of the permissions necessary for your intended use.*

In particular, trademark rights are *not* licensed under this license. Thus, rights concerning third party logos (e.g., on the title slide) and other (trade-) marks (e.g., “Creative Commons” itself) remain with their respective holders.





