OS Motivation [1][2]

IT Systems, Summer Term 2026

Dr. Matthes Elstermann

Contents

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 IT Systems

Figure 1: "Notebook" under CC0 1.0; from Pixabay

  - Inspect and **control** any computer, at any level of interest
  - **Digital sovereignty, sustainability**
    - Knowledge empowers to use/build better solutions that serve our interests

[1] This PDF document is an inferior version of an OER in HTML format; free/libre Org mode source repository.

[2] Material created by Jens Lechtenbörger; see end of document for license information.

- So far, you learned how to build a computer.
  - How does that feel?
- Next, we investigate Operating Systems (OSs) to control computers.

Figure 2: “Search” under CC0 1.0; from Pixabay

  - OS controls what is executing when.
  - Who controls the OS?
  - Who controls the computer, then?

1.4 Computers

(a) Figure under CC0 1.0

(b) Figure under CC0 1.0

(c) Figure under CC0 1.0

(d) Figure under CC0 1.0

(e) Figure under CC0 1.0

(f) Figure under CC0 1.0

(g) Figure under CC0 1.0

(h) Figure under CC0 1.0

(i) Figure under CC BY-SA 4.0

1.5 Enlightenment

Enlightenment is man’s emergence from his self-incurred immaturity. (Immanuel Kant, 1784)

- Self-incurred immaturity
  - You let someone else decide and speak for you
  - Here: Someone else programs your computer beyond your control
- Emergence

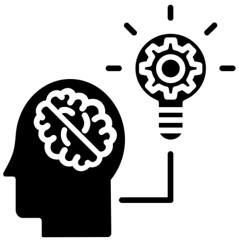

Figure 4: "Brain training" by Shocho under CC BY 3.0 US; cropped from the Noun Project

- You speak for yourself
- Here: You choose controllable programming for your computer

2 Choices and Consequences

2.1 Impact of Choices

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

Figure 5: "Man with megaphone" under CC0 1.0; from Pixabay

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

- Economic impact
  - Should be obvious

- Ecological/social impact

Figure 6: “Figure” under CC0 1.0; from Pixabay

- 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 or attention economy, you **impose** their mechanisms on your family, friends, colleagues …

2.2 Rationale

- **Ownership** of smart devices requires control over software
  - Recall slide on “computers”
  - (Including firmware and underlying hardware)
    - (Topic in its own right)
- Otherwise,

Figure 7: “Control” under CC0 1.0; from Pixabay

  - 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.

2.2.1 Sample Lecturer's Choices

- Smartphone running GrapheneOS
  - No Google "services" required (but possible)
  - F-Droid as app store
  - With preference for offline functionality
    - E.g., navigation (OsmAnd~), Wikipedia (Kiwix)
  - See Wikipedia for list of custom Android distributions
- Before buying a smartwatch with health tracking, make sure that Gadgetbridge supports it
  - Sensitive data does not flow to vendor's servers (as with default app)
  - No "cloud API" whose shutdown bricks the device

3 Software

3.1 Free Software

- Free Software = Free/Libre and Open Source Software (FLOSS)

Figure 8: "Photo of Richard Stallman" by Victor Powell under CC BY-SA 3.0; from Wikimedia Commons

  - 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

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

Figure 9: “Tux, the Linux mascot” under CC0 1.0; from Wikimedia Commons

- 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

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 mode** for programs (I/O and memory access restricted)
    - Ring 2, 1: Usually unused (originally for system services and device drivers)
    - Ring 0: **Kernel mode** for OS (traditionally, ring 0 was most privileged)
    - More rings (Frazelle 2019)
      - 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
  - (Frazelle 2019)
    - “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.”
  - (Frazelle 2020a)
    - “It’s an alarming problem that the code running with the most privilege has the least visibility and inspectability.”
  - (Frazelle 2020b)
    - “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

4.1 Summary

- Free software is a necessary precondition to control computers
  - … and everything embedding a computer
    - … which is a lot in times of ubiquitous computing and IoT!

Figure 10: "Internet of Things" by Wilgengebroed on Flickr under CC BY 2.0; from Wikimedia Commons

- Do you care?

Bibliography

Frazelle, Jessie. 2019. "Open Source Firmware." *Commun. Acm* 62 (10): 34–38. https://doi.org/10.1145/3343042.

———. 2020a. "Opening up the Baseboard Management Controller." *Commun. Acm* 63 (2): 38–40. https://doi.org/10.1145/3369758.

———. 2020b. "Securing the Boot Process." *Commun. Acm* 63 (3): 38–42. https://doi.org/10.1145/3379512.

The bibliography contains references used in this presentation.

License Information