

Open Source Software

Concepts, Reasons, Effects &
Influences in society

Özlem Özgöbek

ozlem.ozgobek@ntnu.no

Knowledge

“Knowledge is a familiarity, awareness, or understanding of someone or something, such as facts, information, descriptions, or skills, which is acquired through experience or education by perceiving, discovering, or learning.” - Wikipedia

"If I have seen further it is by standing on the
shoulders of Giants."

Isaac Newton

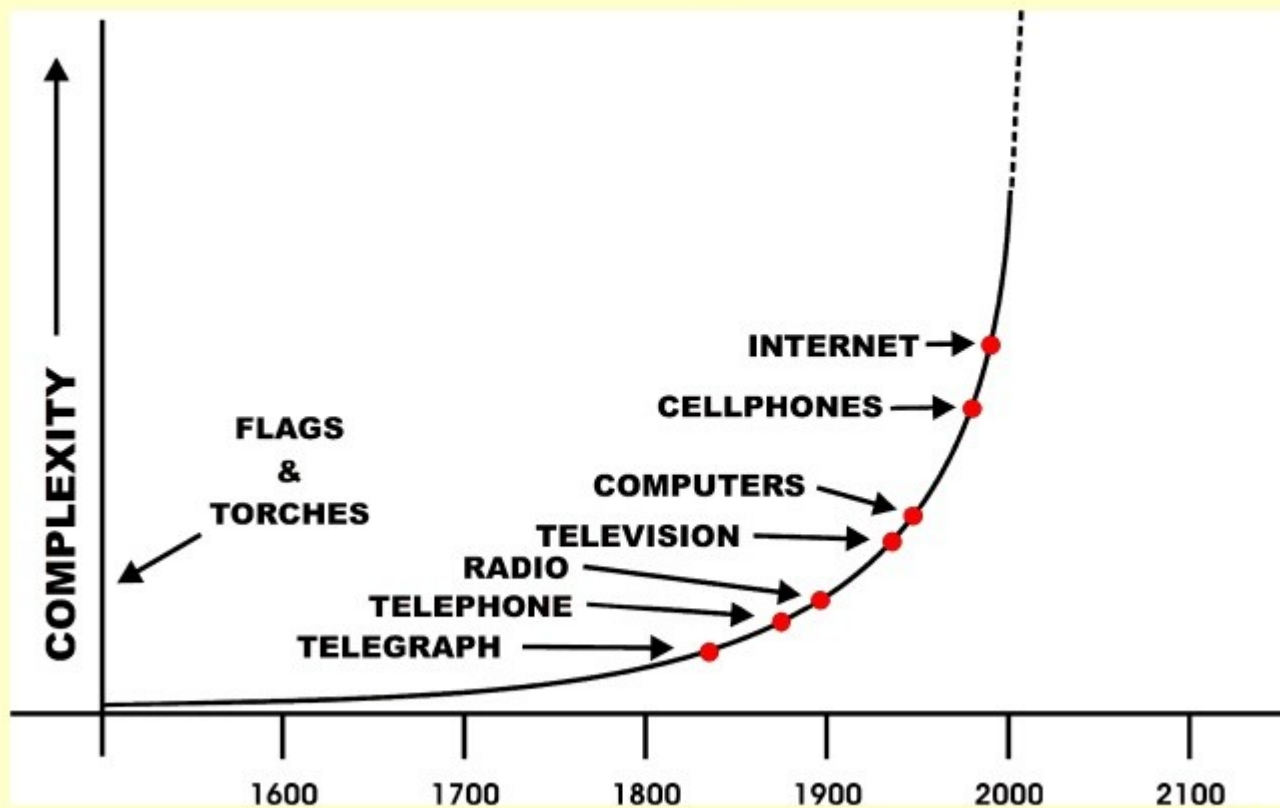
Changing World





Rapidly Changing World

Information-Age Timeline



In this rapidly changing world...

- Specialization increases.
- People specialize in very detailed topics.
- Collaboration becomes more important for bigger technological achievements.
- “Standing on the shoulders of Giants becomes harder.”

Computers

The first digital electronic programmable computer dates back to 1941.

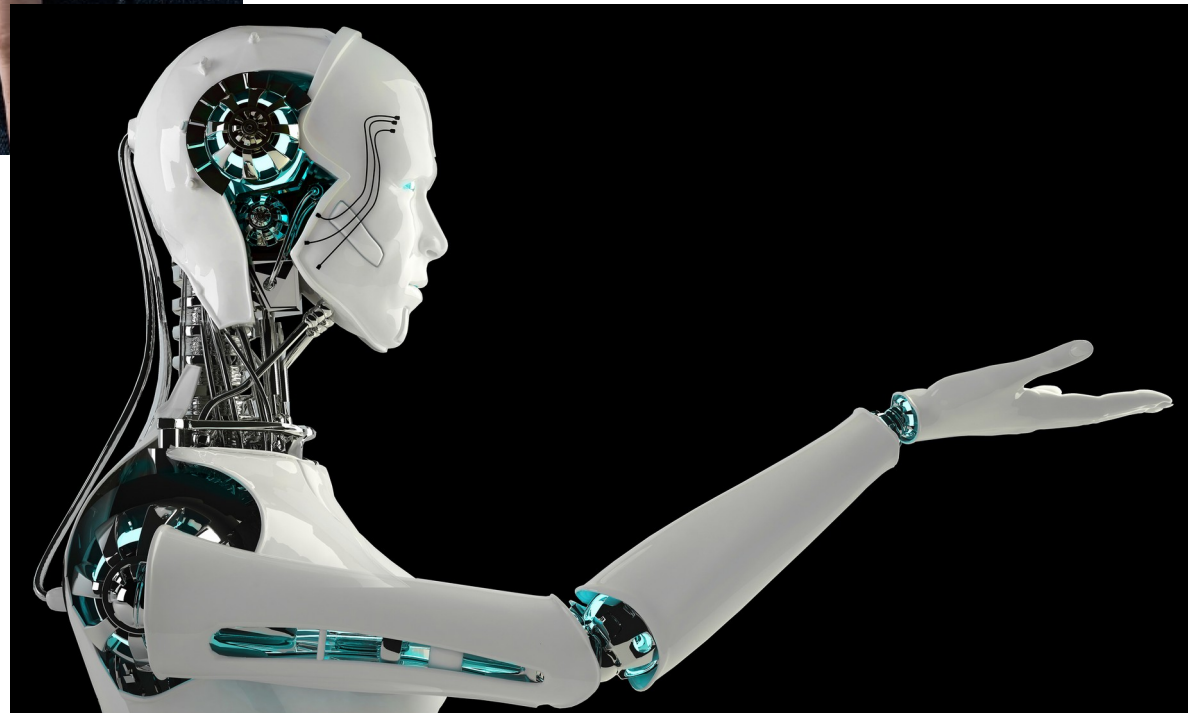
Personal computers are around only since 1980s.

Internet (public commercial use) has started in mid-1989.

Today our everyday lives are dominated by computers in every shape and size.



The future seems exciting



What makes the computer work?

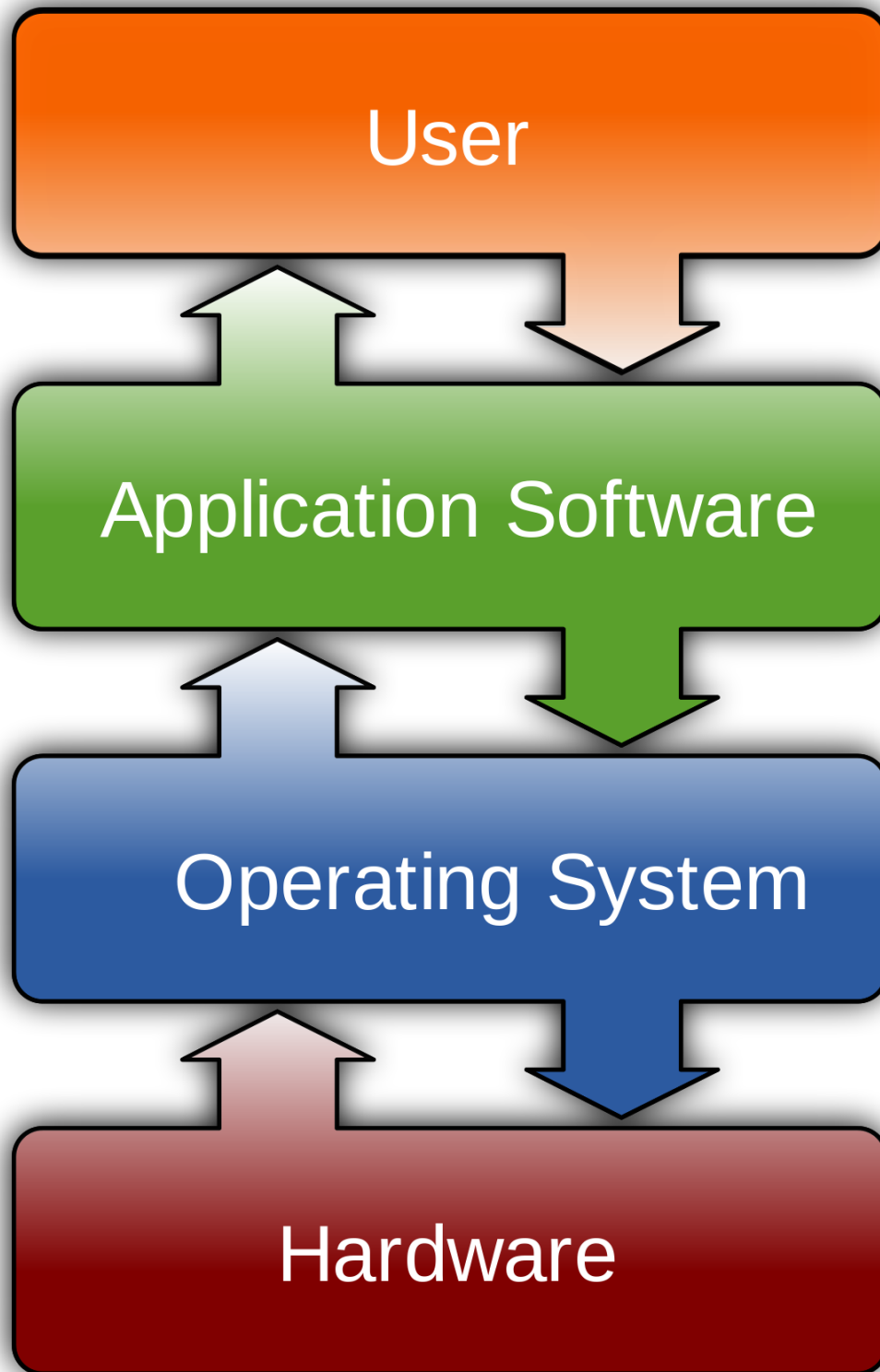


Hardware

+



Software



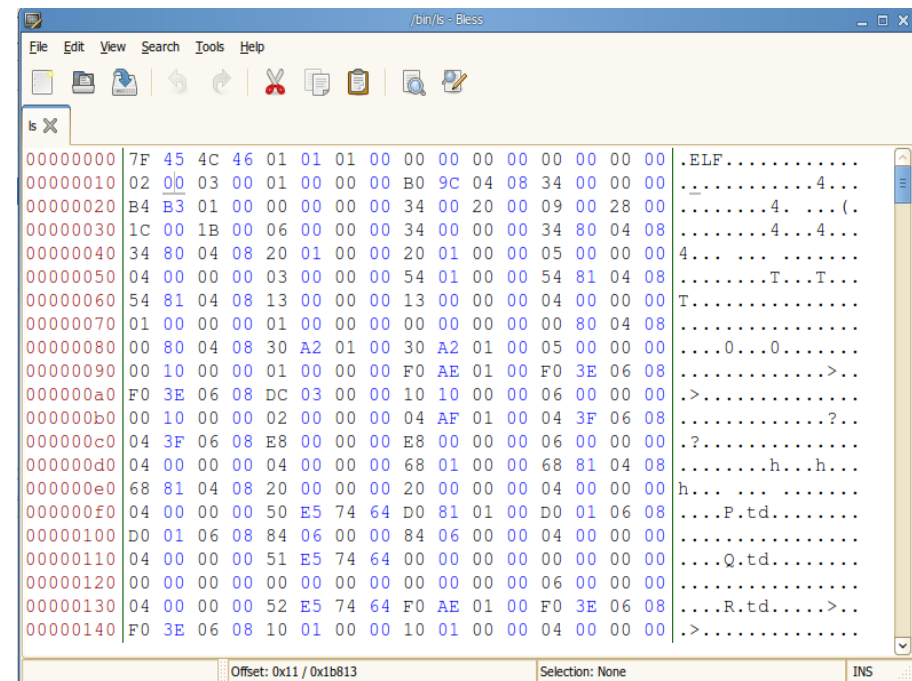
Software

Instructions/set of commands given to the hardware to to specific tasks.

Any program that you use on the computer.
Including the operating system.

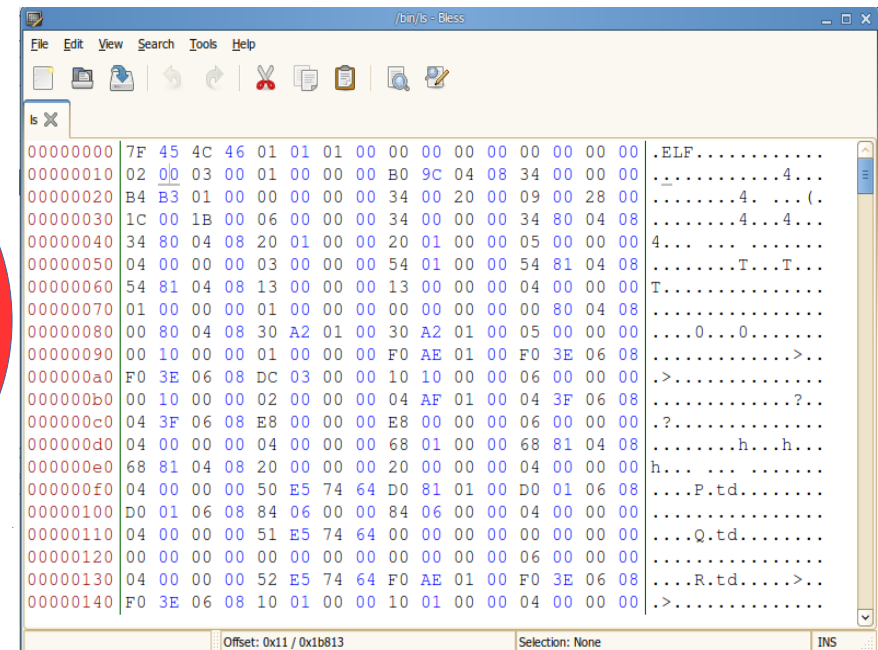
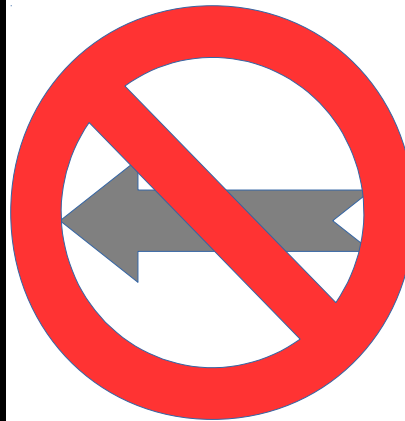
Software

```
public String getContentType() {  
    return contentType;  
}  
  
/**  
 * Buffers request body input stream.  
 */  
private void bufferContent() {  
    if (this.buffer != null) {  
        // Already been buffered  
        return;  
    }  
    if (this.content != null) {  
        try {  
            ByteArrayOutputStream tmp = new ByteArrayOutputStream();  
            byte[] data = new byte[4096];  
            int l = 0;  
            while ((l = this.content.read(data)) >= 0) {  
                tmp.write(data, 0, l);  
            }  
            this.buffer = tmp.toByteArray();  
            this.content = null;  
            this.contentLength = buffer.length;  
        } catch (IOException e) {  
            LOG.error(e.getMessage(), e);  
            this.buffer = null;  
            this.content = null;  
            this.contentLength = 0;  
        }  
    }  
}
```



Software

```
public String getContentType() {  
    return contentType;  
}  
  
/**  
 * Buffers request body input stream.  
 */  
private void bufferContent() {  
  
    if (this.buffer != null) {  
        // Already been buffered  
        return;  
    }  
  
    if (this.content != null) {  
        try {  
            ByteArrayOutputStream tmp = new ByteArrayOutputStream();  
            byte[] data = new byte[4096];  
            int l = 0;  
            while ((l = this.content.read(data)) >= 0) {  
                tmp.write(data, 0, l);  
            }  
            this.buffer = tmp.toByteArray();  
            this.content = null;  
            this.contentLength = buffer.length;  
        } catch (IOException e) {  
            LOG.error(e.getMessage(), e);  
            this.buffer = null;  
            this.content = null;  
            this.contentLength = 0;  
        }  
    }  
}
```



Compiling process is not reversible.

Buying and using software...

... usually you actually buy:

The rights to use it on a single device.

You get only the compiled code.



How did it all begin?

In mid-1970s (1974-1976) copyright laws in the USA begin to restrict software users.

Before that software was not considered copyrightable, had no licenses so it belonged to the public domain.

Beginning from the late 1970s companies started to charge for software licenses to increase revenues.

Okay, but what is the problem?

RESTRICTIONS

People were not anymore allowed to:

- 1) see how the program works,
- 2) share it with other people,
- 3) modify the program if there is a problem.

Let's look into these issues separately.

People were not anymore allowed to:

- 1) see how the program works**
- 2) share it with other people
- 3) modify the program if there is a problem

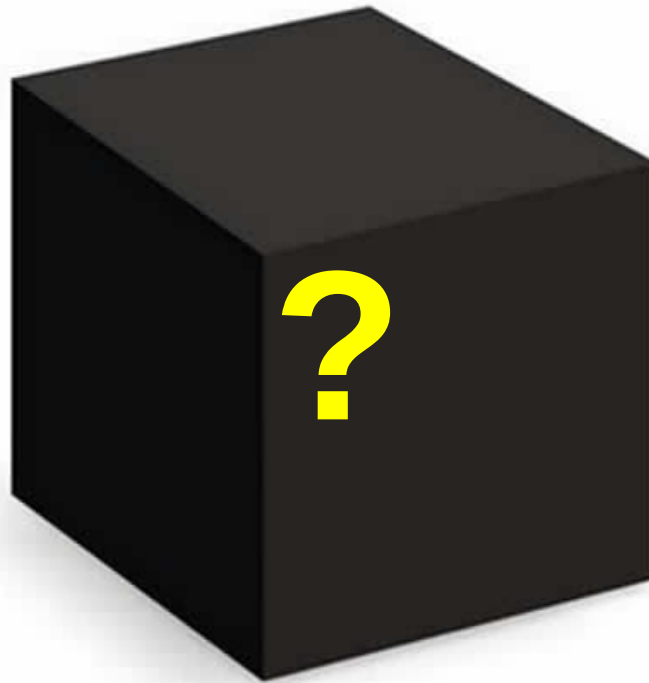
Let's look into these issues separately.

1) See how the program works

Why would you like to see how a program works?

- a) Educational purposes
- b) Privacy and security reasons
- c) To develop an existing idea

a) Educational purposes



b) Privacy and security reasons

Data is important and valuable!

Are apps secretly listening to your calls? Security experts warn of gyroscones

INDEPENDENT

WS POLITICS VOICES FINAL SAY

SPORT

CULTURE

VIDEO

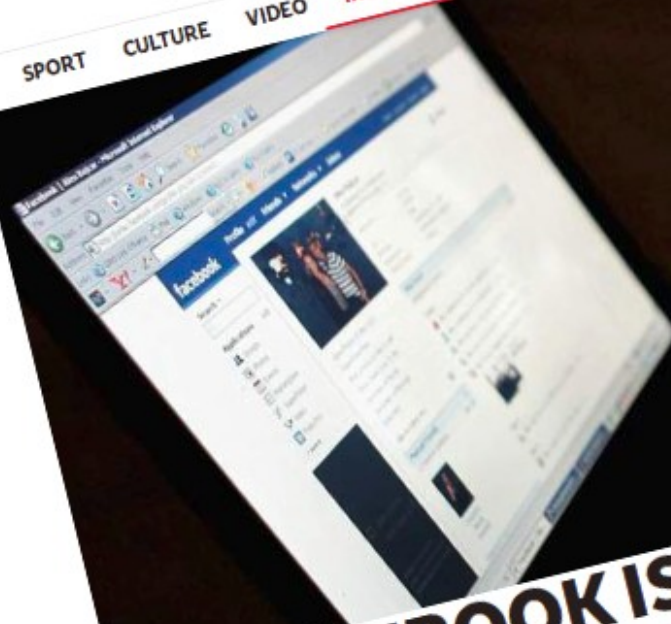
INDY/LIFE

INDYBEST

LONG READS

INDY100

VOUCHERS



FACEBOOK IS USING SMARTPHONES TO LISTEN TO WHAT PEOPLE SAY, PROFESSOR SUGGESTS

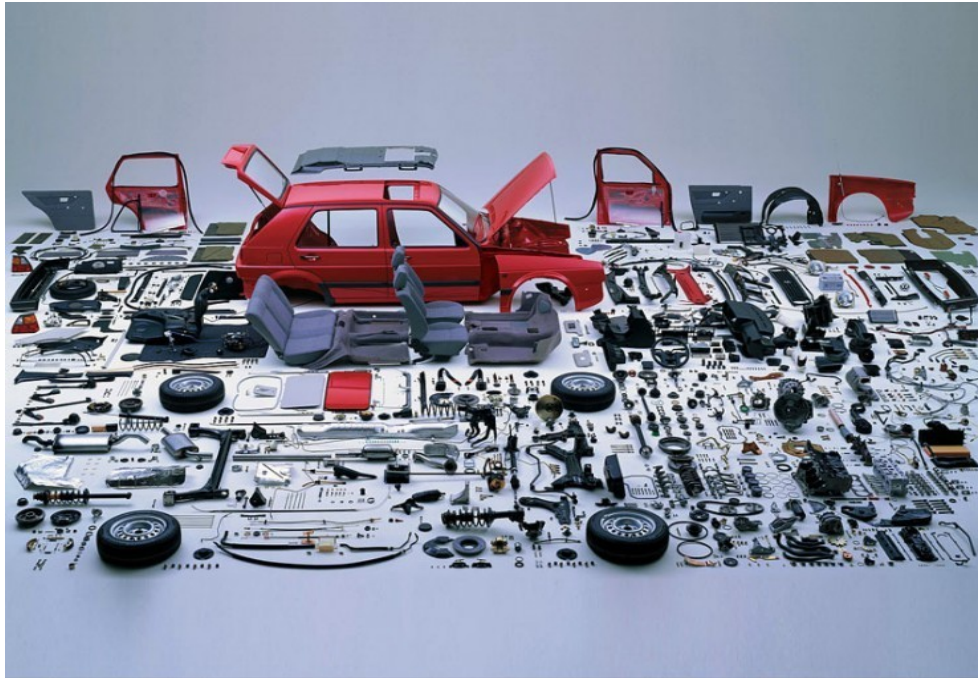
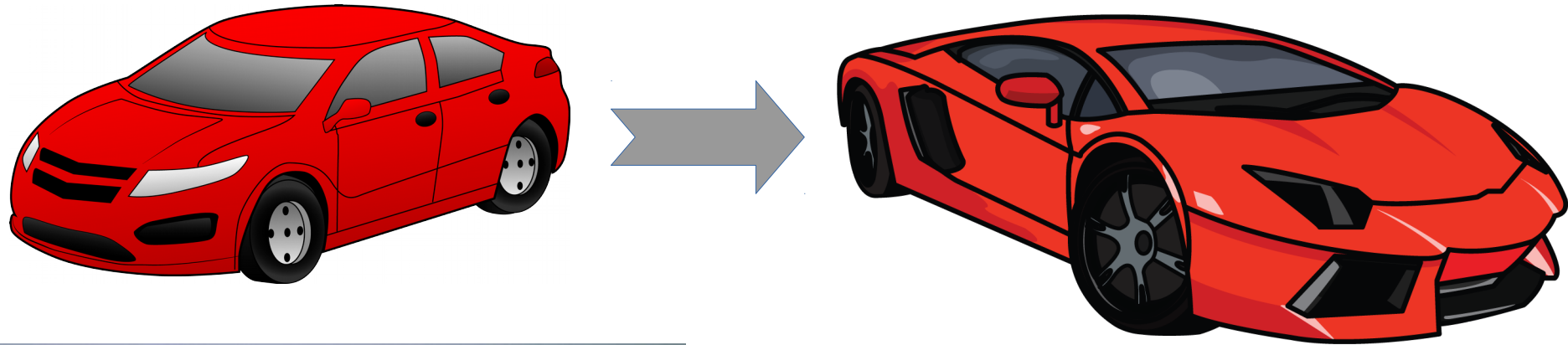
The company says that it does use peoples' microphones, but only to help them out – and there's an easy way of turning it off

actions:

television may track what you watch. Some LG televisions were found to spy on not only what channels were being watched, but even transmitted back to LG the names of files on USB drives connected to the television. Hackers have also demonstrated that they can hack some models of Samsung TVs and use them as vehicles to capture data from networks to

You may want to know what your program/online service is doing with your data.

c) To develop an existing idea



- 1) see how the program works
- 2) share it with other people**
- 3) modify the program if there is a problem

2) Share it with other people

Why would you share a program with others?

- Discussions,
- Getting help or helping to others,
- Just because you like/find it useful, etc.

- 1) see how the program works
- 2) share it with other people
- 3) modify the program if there is a problem**

3) Modify it if there is a problem

- Solving problems quicker.
- Adding some extra functionality specific to your needs.

It can be you or someone you know / hire.

3) Modify it if there is a problem

A real story from 1983:

Richard Stallman, an engineer working at MIT Artificial Intelligence Lab, had a problem with a Xerox printer. → **GNU Project**

1986 - The term “free software” defined.

1989 – GNU General Public License published.

1992 – Linus Torvalds licensed Linux with GPL.

Free Software

Freedom 0: The freedom to **run** the program for any purpose.

Freedom 1: The freedom to **study** how the program works, and change it to make it do what you wish.

Freedom 2: The freedom to **redistribute** copies so you can help your neighbor.

Freedom 3: The freedom to **improve** the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

Free Software

It is about rights and sharing.



Free to Use

The freedom to run the program as you wish, for any purpose.



Free to Access

freedom to access to the source code of the software.



It's all about sharing!

From before we can even remember, most of us were taught to share.



Free to

The freedom to study how the program works, and to modify it to suit your needs.



Free to Modify

The freedom to distribute copies of your modified versions to others.

Open Source

Source code is publicly available.

But there may still be some restrictions.

Rights and restrictions are described in the license and they are legally valid.

Open Source

Every free software is open source but not every open source software is free software.

Open source software \neq Free software

There are so many different software licenses with different levels of openness.

What to call it?

Debate: *Open source vs. free software*

Free and open source software (FOSS)

Software Licenses

Software licenses in plain English:

<http://tldrlegal.com/>

Comparison of FOSS licenses:

https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses

Let's have a brief look!

Why are software licenses important?

It is about your digital legal rights!

It is about society!

It is about future!

Why is FOSS important?

- Transparency, privacy, security.
- Equal rights.
- Collaborative working culture.
- Competitive products.
- Ethical work.
- Re-usability and reproducibility.

Transparency

It is important to **keep in control** of one's own data.
Know what your computer/device/app does.

Transparency: Better privacy and security.
But how?

More people checking the source code and testing the software is always better than a few people doing it.

When there is a known vulnerability in the software, it is more faster to fix it.

Transparency

If less secure systems are more profitable, companies wouldn't work hard to produce more secure systems.

Being open source is no guarantee of security but there is a higher chance of being secure.

Equal rights

Equal rights to study, learn, develop, use and modify the software for everyone.



Collaborative working culture

A few examples:

Linux kernel development: <https://git.kernel.org/>

LibreOffice: <https://gerrit.libreoffice.org>

WordPress: <https://core.trac.wordpress.org/timeline>

Competitive products

Provides many alternatives.

Best ones survive.

Ethical work

Transparent & legal

Re-usability

Re-usability of ideas and codes allows faster development of technology.

You don't have to invent the wheel every time.

Reproducibility of scientific research.

Sustainable.

A recent campaign: **Public money? Public code!**

<https://publiccode.eu/>

<https://www.youtube.com/watch?v=iuVUzg6x2yo>

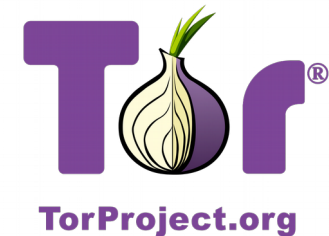
Misconceptions – Free software

- It is always free of charge.
- It is not secure.
- It doesn't have an owner.
- Users don't get enough technical support.
- Software developers doesn't earn money or benefit being a part of free software development.
- No one is following, anyone can use the open source code to develop a proprietary (closed source) software.

Misconceptions - Open source software

- All open source software give the users same rights.
- Very similar to free software misconceptions, however answers may be different.

How common is free/open source software (FOSS)?



FOSS as an influence

Free and open source software movement influenced many other movements in different domains.

Open Standards

Documents, software, communication etc.

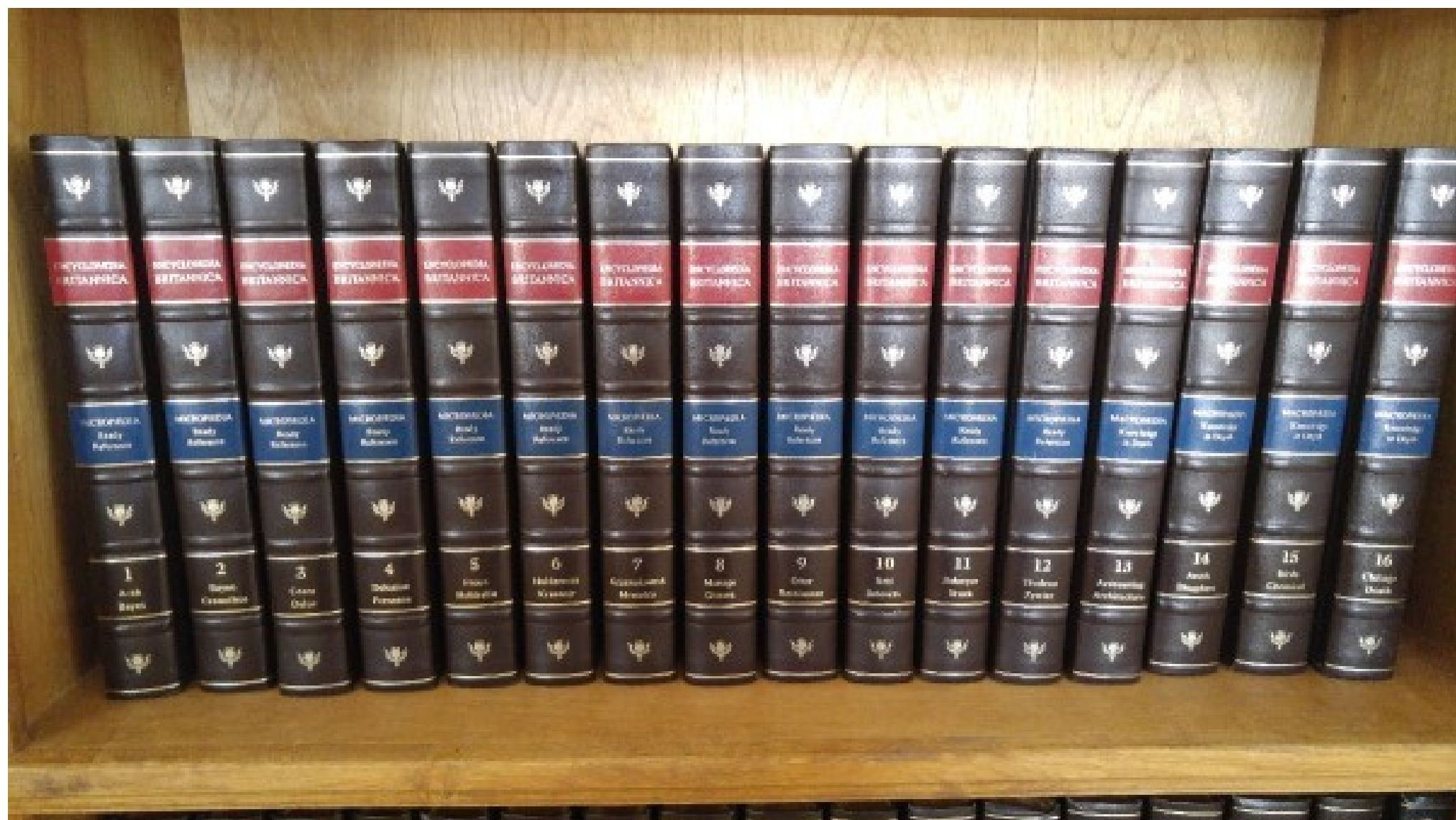
Readability, interoperability, backwards compatibility.

Can be implemented by anyone without restriction.

Organization for the Advancement
of Structured Information
Standards (OASIS)

<https://www.oasis-open.org/org>

**WE CANNOT
READ YOUR
DOCUMENTS**
DOCUMENTFREEDOM.ORG





WIKIPEDIA
The Free Encyclopedia

<https://creativecommons.org/>



Attribution

Noncommercial

Non- Derivatives

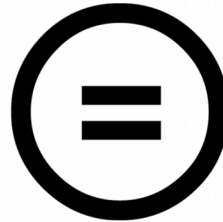
Share alike



Compulsory - Must always **credit** me.



Use it but don't make **money**



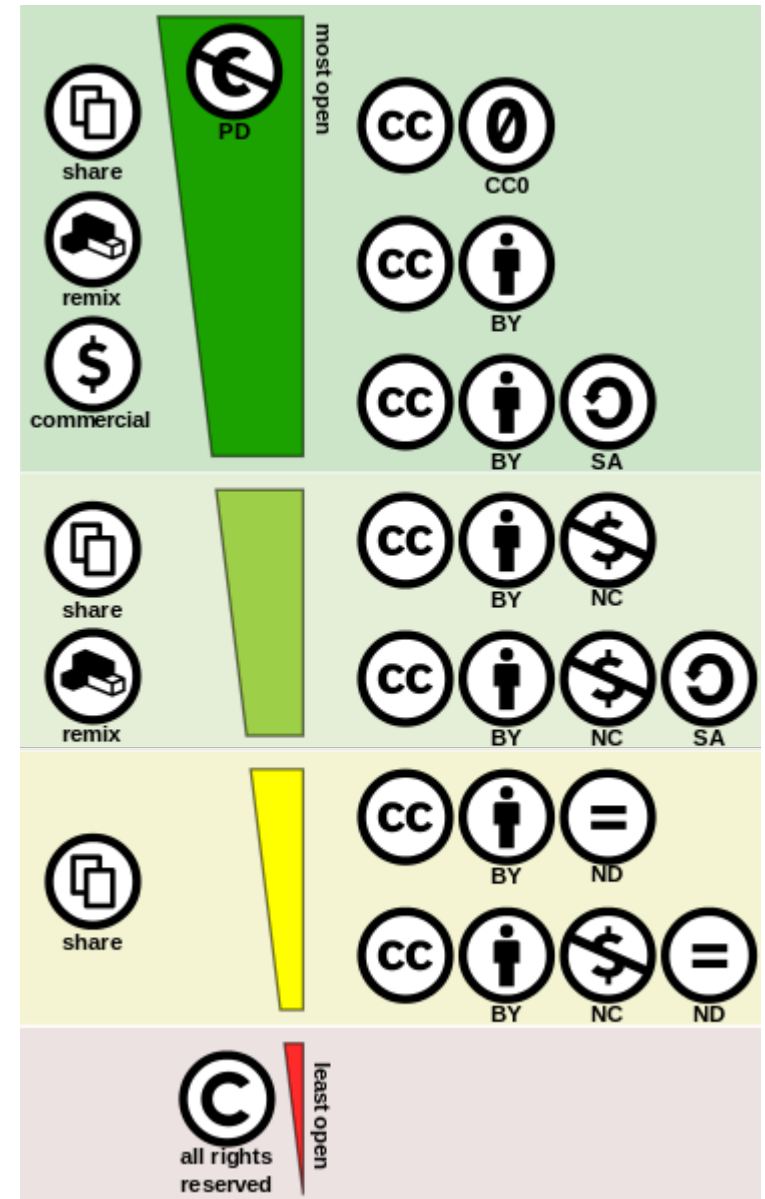
Your version must **equal** mine - no changes



If I allow you to change it, **repeat** my CC **licence**

by @EduWells more info at EduWells.com

<https://vimeo.com/13864570>



Open Access

Research outputs which are distributed online and free of cost or other barriers.

<https://sparcopen.org/>



Open Data

“The idea is that some data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents or other mechanisms of control.” - Wikipedia

Some examples: data.norge.no, data.gov, data.gov.uk, linked open data

Open Street Map

“OpenStreetMap is built by a community of mappers that contribute and maintain data about roads, trails, cafés, railway stations, and much more, all over the world.”

<https://www.openstreetmap.org>

Emphasizes local knowledge, community driven, open data.

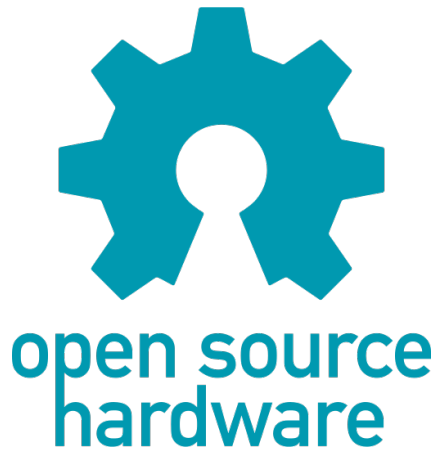
Search

Where is this?

Go



Open Source Hardware



Open design of hardware.

“Hardware design (i.e. mechanical drawings, schematics, bills of material, PCB layout data, HDL source code and integrated circuit layout data), in addition to the software that drives the hardware, are all released under free/libre terms.” Wikipedia

Open culture: <http://www.openculture.com/>

Elephants Dream: *“The world’s first open movie, made entirely with open source graphics software such as Blender, and with all production files freely available to use however you please, under a Creative Commons license.”*

<https://orange.blender.org/>

FOSS as an influence

- Wikipedia
- Creative commons
- Open access
- Open data
- Open street map
- Open source hardware
- Open education
- etc.

Free Software Foundation: <https://www.fsf.org/>

FSF Europe: <https://fsfe.org/>

Creative Commons: <https://creativecommons.org/>

Digital Freedom Foundation:

<http://www.digitalfreedomfoundation.org/>

GNU: <https://www.gnu.org/>

Software licenses in plain English: <https://tldrlegal.com/>

Open Source Initiative: <https://opensource.org>

<https://opensource.com/>

<https://opensource.com/resources>

Trondheim Free/Open Source Software Facebook Group

<http://www.facebook.com/groups/15010456913322>

Thank you!

Questions?