huld

Hannu Leppinen, 8.10.2021

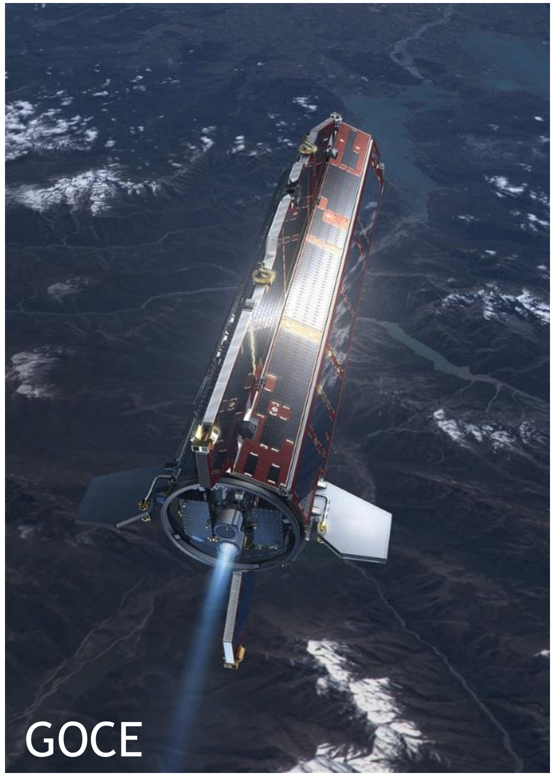
Linux in space

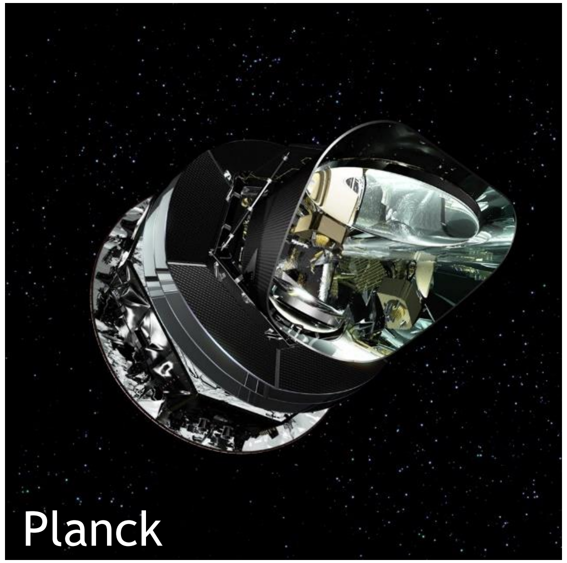
huld

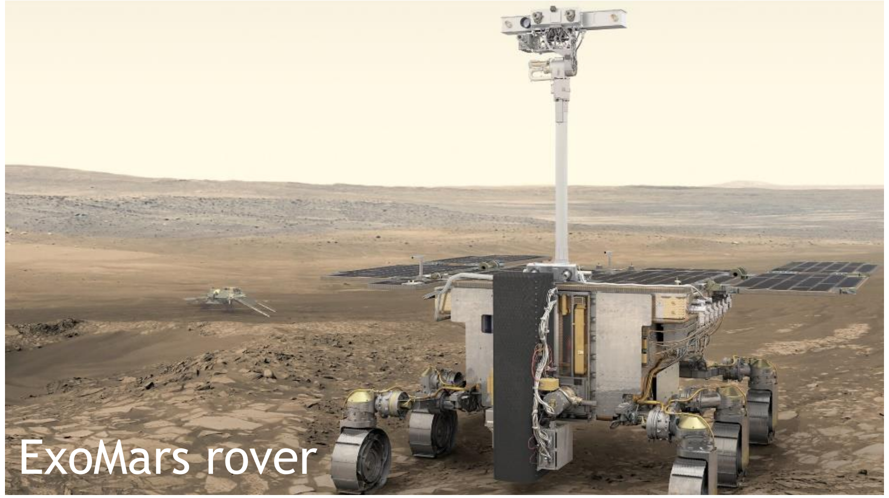
About Huld

- A European technology design house
  - Serving customers around Europe
  - Mechanical, electronic, software, and industrial design → complete products
  - Sectors: space, defence & national security, industrial, healthcare and medical
- ~400 people
- Turnover 30 M€
- Offices in Finland and in Czech Republic
- 150+ ESA projects completed
- I’ve worked since 2015 in the Space & Defence business unit on spacecraft software projects

huld







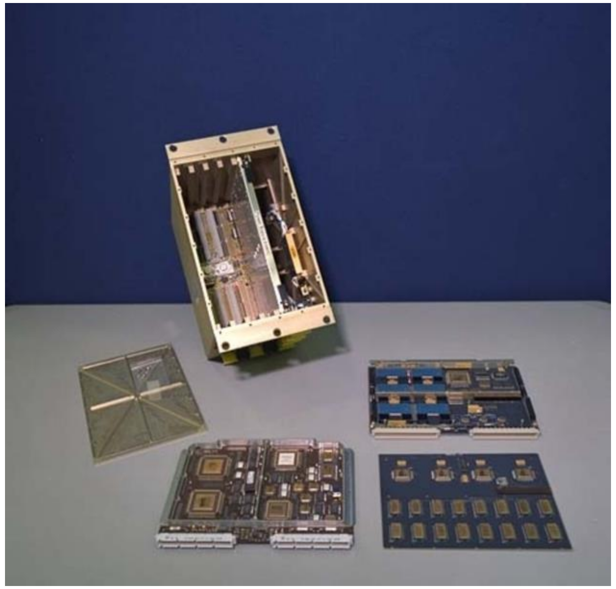
Contents

- Background: Space SW
- Linux in space: how and why
- History, future, and conclusions

Spacecraft computers

- Presentation focus: spacecraft-controlling embedded Linux
  - Linux also used e.g. on laptops on ISS
- A modern spacecraft typically has multiple computers
  - Main computer controlling the spacecraft
  - Instrument computer(s) controlling the payload(s)
  - Possibly also many microcontrollers in various devices

Some laptops on ISS run Linux

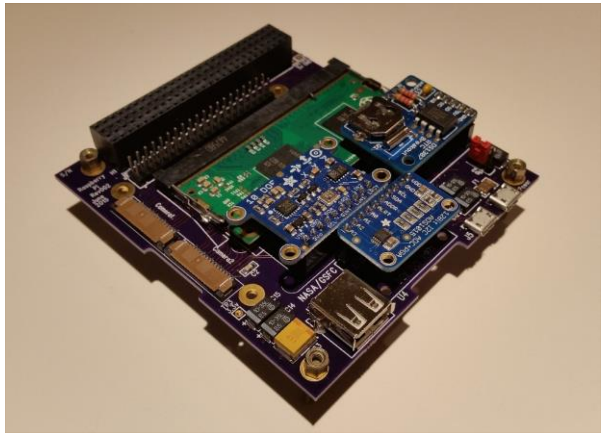


”Space-grade” computer parts

Hardware

- Larger ”traditional” spacecraft (e.g. ESA, NASA) usually use radiation-hardened computers
  - Expensive
  - Limited performance
- Academic and commercial entities often use off-the-shelf electronics for their spacecraft computers
  - Higher performance
  - Less reliable in space use

BepiColombo (Mercury mission) on-board computer

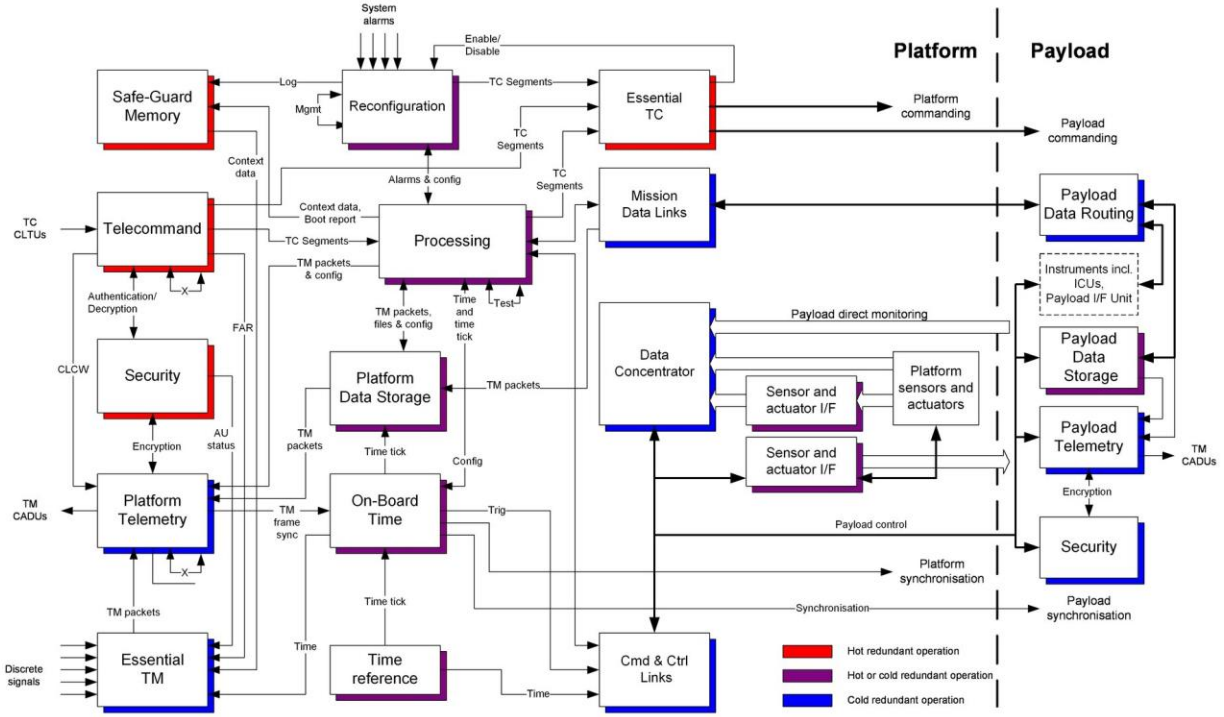


Raspberry Pi-based on-board computer

huld

Computer responsibilities

- Typical computer responsibilities include:
  - Processing and executing incoming telecommands
  - Producing outgoing telemetry packets
  - Collecting and storing housekeeping data
  - Controlling spacecraft subsystems
    - Communication buses



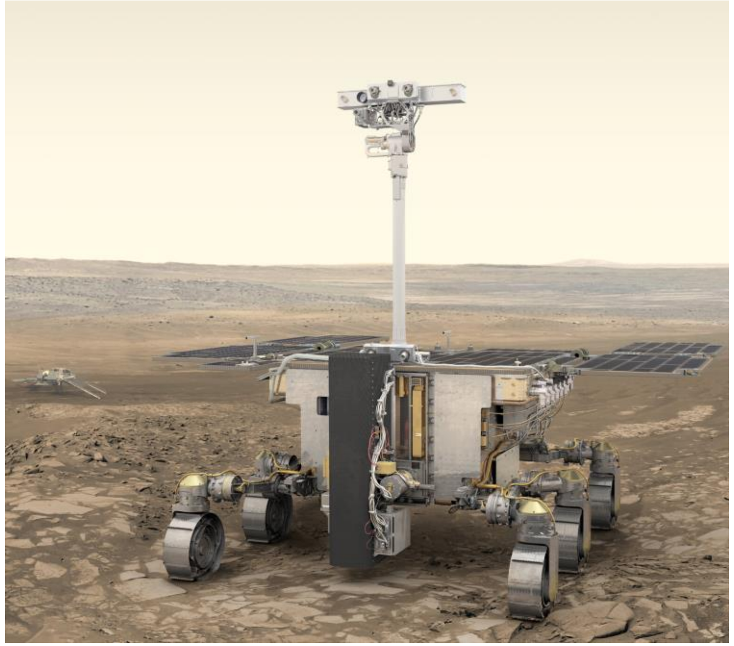
huld

Space SW characteristics

A spacecraft is an embedded system.

Some differences to common terrestrial ones:

- Redundancy and reliability
  - Hard to fix hardware in orbit
- Criticality
  - Hard deadlines (schedulability)
  - Possible loss of mission on failure
- Space environment considerations
  - Hardware choices need to consider the operational environment
- Limited TC/TM availability and bandwidth
  - Operations need to be planned beforehand



”Traditional” space SW

- Operating systems
  - None (single main function + infinite loop, i.e. cyclic executive)
  - Hard real-time operating systems
    - E.g. RTEMS, VxWorks, Ada runtime
- Programming languages
  - Assembly, C, Ada
- Each line of code justified, verified, unit tested…
- Spacecraft control SW:
  - SW size RoM around 100 000 LOC
  - Limited reuse of existing code
  - Typical project cost several million €

Contents

- Background: Space SW
- Linux in space: how and why
- History, future, and conclusions

Why Linux: technology

- Support for many hardware architectures
- The Linux code base can be relied upon due to its very wide adoption
  - Availability of source code → supports troubleshooting
- Support for many communication protocols and standards
- Existing Linux SW catalogue potentially useful for space applications as well:
  - Data compression
  - File systems
  - Operations scheduling and scripting
  - Data processing and algorithms
  - Security-related software
- Flight SW can be modularized into sets of programs and processes
  - Easy to update individual programs instead of patching the whole SW
- Optional real-time capabilities (PREEMPT_RT kernel patch)

Why Linux: tools and people

- Development tools available, e.g. GNU project
  - Support for many programming languages
- Desktop Linux for development → same OS in development environment and in the embedded target
  - May help debugging
- Large developer community
  - Support
  - Developer recruitment pool
- Vendor independence
- Open-source licensing
- Potentially low cost of adoption

huld

Challenges: space-grade HW

- Typical ”space-grade” computers often not powerful enough to run Linux
- Standard industrial ”off-the-shelf” hardware used instead
    - Suitability for space, especially radiation?
- Mitigation
    - Hardware redundancy
    - For short lifetime missions, space-grade not necessarily needed
    - Also ”powerful enough” space-grade computers becoming available

Challenges: hard real-time

- Can some low-priority load cause some critical computation to miss its deadline?
    - E.g. thruster fires too late…
- Is Linux good enough for hard real-time applications?
- Mitigation
    - PREEMPT_RT patch
    - Thorough load testing for real-time behaviour likely needed

Challenges: complexity

- Complexity of validating Linux-based systems
  - The kernel has millions of lines of code
  - Perhaps not possible to justify, unit test, verify, validate every individual line of code?
- Could the SW end up in unrecoverable states?
- Safety-critical certification?
- Mitigation
  - Recovering from unexpected states → heartbeat and an external watchdog chip
  - Stripping unnecessary code components → complexity perhaps reduced

Example: SpaceX

- SpaceX uses Linux at least in Falcon 9, Dragon, and Starlink spacecraft
- Off-the-shelf x86 processors used
  - Redundancy used to improve reliability
- Custom Linux distribution
  - Unnecessary parts stripped → complexity reduced
  - PREEMPT_RT patch → improved real-time behaviour
- Modern technologies
  - C++: vehicle control
  - Python: tools, testing, automation
  - Javascript/HTML/CSS: spacecraft displays
- Apparently ”good enough for NASA”
  - Interesting: how was it certified?

Astronauts operating the Dragon displays.

Contents

huld

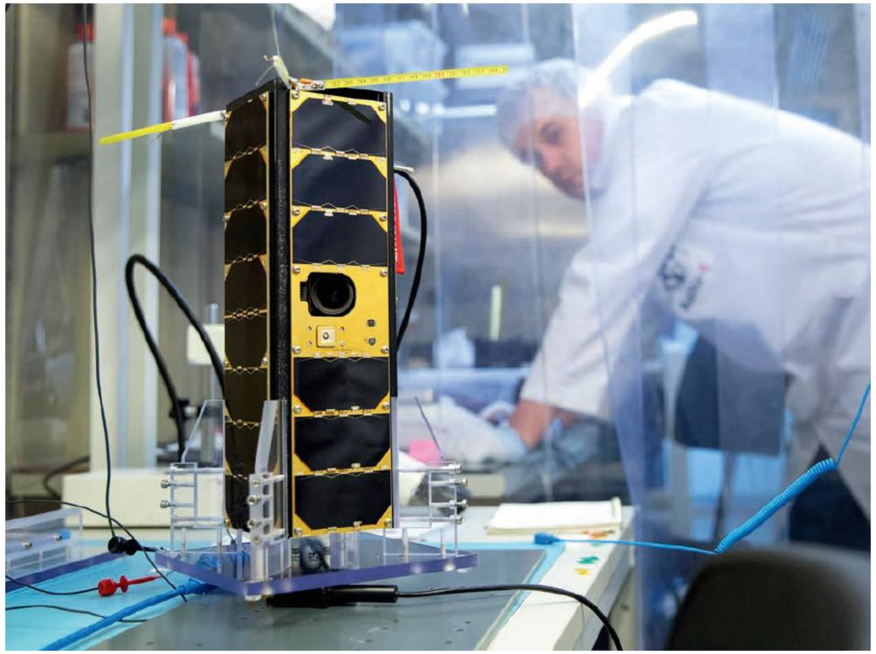
Beginnings: 1999-2002

First ”Linux in space” study: FlightLinux in 1999 – 2002

- NASA research project
- Planned to test a simple Linux binary on the UoSat-12 satellite
  - Small satellite launched in 1999, operated by Surrey (UK)
  - Tested e.g. printing ”Hello World” to a serial port
  - Unclear if this was ever run

Academic use: 2003-2013

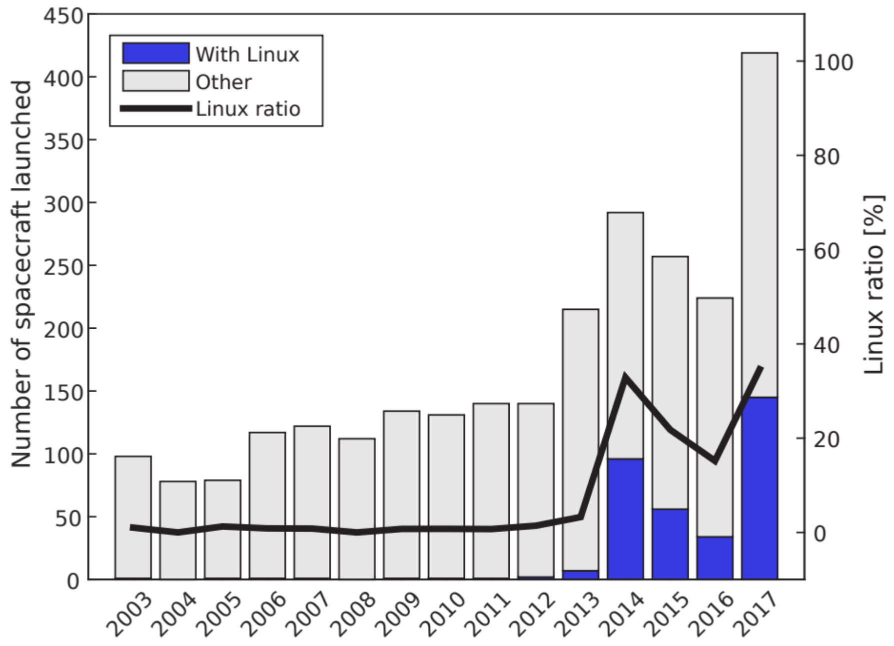
- QuakeSat in 2003 was the first satellite to launch with a Linux-based main computer
- For the next ten years, only a handful of Linux-equipped spacecraft were launched
- Most of these academic small satellite projects
  - E.g. Aalto-1 from Aalto University



The Aalto-1 nanosatellite. Photo: Aalto University.

Commercial interest: 2013-2018

- Linux use increased also in the commercial sector
    - Planet Labs: Dove satellites
    - SpaceX: Falcon 9, Dragon
- Planet Labs launched around 300 Linux-based spacecraft in 2014-2018
    - Small, approx. 4 kg ”CubeSats”
    - Earth imaging, remote sensing



”The Linux era”: 2019-present

- SpaceX leads the pack
  - Starlink satellites: thousands of satellites launched with Linux
    - Satellite internet
    - E.g. 2020: 1283 satellites launched, of which 833 were Starlink
  - Crew Dragon: manned spacecraft controlled by Linux computers
- Many other commercial users as well; not all publish their use
- Institutional players (ESA, NASA) slower to adopt
  - But e.g. Mars drone copter Ingenuity (NASA) runs Linux

huld

Future of Linux in space

My prediction:

- SpaceX and other ”NewSpace” companies will continue to lead the pack
- ”Traditional” space industry will be slower to adopt, but use of Linux will increase in the payload side
    - main spacecraft control still with a more critical computer?
- In 5 – 10 years: Linux just a common part of the toolbox?

SpaceX Starship on the Moon. Image: SpaceX.

Conclusions

- Linux already surprisingly popular in space
  - The trend will continue
- Challenges remain
  - Difficult to rigorously verify & validate complex Linux-based systems
  - Solutions from other industries using Linux in safety-critical applications?
- However, SpaceX has shown it can be done
  - Manned spacecraft controlled by Linux
  - ”Good enough for NASA”

Crew Dragon docked to the ISS

Thank you!
Questions?

Hannu Leppinen
Hannu.Leppinen@huld.fi