Linux4Space.Org
A Reference Linux Distribution for Space Applications

Dr. Lenka Kosková Třísková
Lukáš Mázl

28.6.2023
Join at slido.com
#Linux4Space2023

Click Present with Slido or install our Chrome extension to display joining instructions for participants while presenting.
What is Linux For Space?

- Collaborative open source project.
- It was founded with the intention of creating an open source Yocto based Linux distribution suitable for the space applications.
- The project brings together all the stakeholders: the software and hardware developers, the suppliers, and technology companies.
- The Linux4Space is designed to be compliant with the ESA Standards (ECSS - European Cooperation for Space Standardization) and it is based on community defined requirements.

The Linux For Space has started in February 2022.
Linux4Space.org core team

- Lenka Kosková Třísková & Lukáš Mázl, Technical University of Liberec
- Tomáš Novotný & Martin Sabol, Czech Aerospace Research Centre
- Javier Fernandez Salgado, TEC-SWF, ESTEC, ESA
- Eric Weiss, Linux Consultant, PREVAS
- Tim - Principal Software Engineer, Sony Electronics
- Kaiwan - Linux Author, Trainer, Consultant
Who we are?

Payloads for space.
Several running space realizations.
Requirements collection.

VZLUSAT-2 with Linux currently in space.
Requirements collection.

Web, project organization, propagation and management
Requirements collection
Project implementation
What we do

- All the information available at [linux4space.org](http://linux4space.org)
- Community regular meetings
- Knowledge gathering ([http://wiki.linux4space.org/](http://wiki.linux4space.org/))
- Requirements gathering ([https://gitlab.com/linux4space/linux4space_requirements](https://gitlab.com/linux4space/linux4space_requirements))
- Standards analysis
- Implementation ([https://gitlab.com/linux4space/yocto](https://gitlab.com/linux4space/yocto))
- Networking (meetings, slack, other activities)
Why are you interested in Linux4Space?

Click Present with Slido or install our Chrome extension to activate this poll while presenting.
Linux4Space Use Case
**Payload system:** Linux4Space is designed as the primary system for payload, not as the central platform system.

**No GUI:** Linux4Space does not provide any graphical user interface. It is designed as a system based with a text-based UI (terminal), or as a fully autonomous system or a system using a mission specific communication protocol.

**Real-time operation:** The Linux4Space shall guarantee the time determination of events processing.

**Space environment ready:** The system shall operate reliably and safely in a typical space physical environment, focusing on the effects of high temperature and radiation. The system can trigger a safe mode at defined conditions. It is capable of a fast shutdown and redundant critical data backup.
Linux4Space UseCase

- **ESA Standards**: The system is designed to be as close to all the software related ESA standards and recommendations as possible.
- **YOCTO-based reference distribution**: The system definition is a set of the YOCTO meta layers.
- **Safety and reliability**: The system design is coordinated with the outputs of the ELISA open-source project to keep the system safe and reliable for critical applications.
- **Community-based**: The Linux4Space is an open-source project open to everyone interested in Linux space applications.
Linux in Space (cubesats)
Linux in Space - CubeSats 2022

- Data includes unique CubeSat missions
- CubeSat may have multiple systems
  - RTOS for OBC
  - Linux for Payload
- How to find this information?
  - Pick a CubeSat launched in 2022
  - Find information about this CubeSat with keywords Linux, software, RTOS, and firmware.
  - If you have not found anything, mark this CubeSat as SW Undefined.
  - Repeat these steps until you get mad.
## Other Linuxes in Space

<table>
<thead>
<tr>
<th>Year</th>
<th>2019</th>
<th>2020</th>
<th>2021</th>
<th>2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launches</td>
<td>188</td>
<td>162</td>
<td>326</td>
<td>334</td>
</tr>
<tr>
<td>Linux (%)</td>
<td>(no data)</td>
<td>(no data)</td>
<td>~15%</td>
<td>~20%</td>
</tr>
</tbody>
</table>
Linux4Space Wiki

● The main goal:

We want to share all the information about the "Linux" and the "space" we already have.

● Our "role model" is the Embedded Linux wiki (elinux.org)
● We started the Wiki in April, providing a general structure, and continue adding content day by day.
● Everyone is welcome to add any information.
● Currently we are hosting the system with the help of the Technical University of Liberec.
Linux4Space - Requirements summary
Linux4Space Requirements

- To be ready for aero/space and critical applications, traceable design important
- It means to go from requirements to implementation and have the test specification with the verification from the beginning
- The Requirements up to now were defined by the core group in 2022 and at the beginning of 2023 (the output in public github)
Linux4Space Requirements Summary

- “The Space” = Environment = Radiation:
  - The system shall switch off immediately.
    - The filesystems used to store mission critical data shall mostly work in read-only mode.
- Interfaces:
  - SpaceWire, SpaceFibre
  - CubeSat Space Protocol
- Power constraints:
  - The system shall have a configuration to define the process with a certain power budget.
- Not so much safety/reliability regarding the kernel itself
- Reporting, environment information etc.
From Requirements to Implementation

- The main decision: yocto-based open source
- Each of requirements has ID and an attribute “yocto”
- We want to keep the track from the requirements to the implementation, recipes etc.
- The main stakeholders: the core group and partners, for future also the related ECSS
Path to ECSS compliance
ECSS system of standards in general

General
System description

Space project management
- Project planning and implementation.
- Configuration and information management.
- Cost and schedule.
- Risk management

Space assurance
- Product assurance management.
  - Quality assurance.
  - Dependability.
  - EEE components
  - Materials, mechanic parts and process
  - Software product assurance.

Space engineering
- System, electrical, optical mechanical, software, communications, control engineering.
  - Ground systems and operations

Space sustainability
- Space debris
- Planetary protection
- Space situational awareness.
ECSS related to Linux4Space

- Linux4Space shall be as close to ECSS as possible
- Relevant document selection by these rules:
  - Include documents focusing on:
    - Software engineering
    - IO Communication (Drivers)
  - Exclude irrelevant documents focusing on:
    - Material
    - Mechanical
    - Electrical
ECSS related to Linux4Space

General System description

- Space project management
  - Project planning and implementation.
  - Configuration and information management.
  - Risk management.
- Space assurance
  - Product assurance management.
  - Quality assurance.
  - Dependability.
  - EEE components.
  - Materials, mechanical parts and processes.
  - Software product assurance.
- Space engineering
  - System, electrical, optical, mechanical, software, communications, control engineering.
  - Ground systems and operations.
- Space sustainability
  - Space debris.
  - Planetary protection.
  - Space situational awareness.

Documents:

- Space project management: 6 documents, 0 documents
- Space assurance: 62 documents, 9 documents
- Space engineering: 65 documents, 36 documents
- Space sustainability: 2 documents, 0 documents
ECSS implementation strategy

- We defined impact attributes for document as follow:
  - Document has impact on whole project (Documents with the highest impact)
  - Document has impact on unique user-space software
  - Document has impact on drivers
  - Document is unrelated to our project (Documents with the lowest impact)
ECSS related to Linux4Space

General System description

Space project management
- Project planning and implementation.
- Configuration and information management.
- Risk management.
- Cost and schedule.
- Product assurance management.
- Quality assurance.
- Dependability.
- EEE components.
- Materials, mechanical parts and processes.
- Software product assurance.

Space assurance
- System, electrical, optical, and mechanical, software, communications, control engineering.
- Ground systems and operations.

Space engineering
- Space sustainability
- 2st phase: 7 documents
- 3rd phase: 2 documents

Space sustainability
- 0 documents
- 0 documents

<table>
<thead>
<tr>
<th></th>
<th>2st phase</th>
<th>3rd phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space project</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Space assurance</td>
<td>2st phase: 7 documents</td>
<td>3rd phase: 2 documents</td>
</tr>
<tr>
<td>Space engineering</td>
<td>2st phase: 10 documents</td>
<td>3rd phase: 26 documents</td>
</tr>
<tr>
<td>Space sustainability</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Linux4Space Yocto Structure
General principles

- From requirements to yocto implementation
- Layers shall be added to existing yocto structure with any BSP
- 3 HW references: Raspberry Pi, VZLU, qemu

We shall try to keep the main layers as HW independent as possible
Linux4Space Layers

- **System Layer, meta-linux4space:**
  - The main distro configuration
  - Image definition
  - Kernel related recipes
  - Bootloader related recipes
  - All the kernel-space applications
  - System configuration

- **Application Layer, meta-linux4space-app**
  - The applications for the space use case
  - e.g., the app responsible for creating the diagnostic reports
  - Telemetry, navigation… everything not required to work in the kernel-space
OpenEmbedded ("main yocto")

BSP Raspberry Pi  |  BSP PetaLinux  |  BSP Freescale

meta-linux4space-app
(as close to ECSS as possible)
Useful apps for space, working in user-space (communication protocols, reporting etc.)

meta-space
(nobody checked the ECSS)
useful space apps

System related: boot, image selection, update, upgrade, interfaces…
Image definitions

Linux4Space

BSP Board with iMX

"Normal embedded stuff"

OpenEmbedded ("main yocto")

HW related stuff
From Requirements to Yocto

The requirement model has attribute named *yocto* for each of requirement.

Possible values:

- Distro feature.
- Recipe System Layer.
- Recipe Application Layer.
- Not Yocto related.
- Not yet defined.
Recipe System Layer

● A functionality has to be:
  ○ Configured for and existing package (provided config etc.)
  ○ Programmed (provided source code with configuration and other stuff).

● Why in System Layer? Required functionality needs to:
  ○ Update the bootloader
  ○ Update the kernel
  ○ Be a part of system services

● Example:
  ○ Random image selection during boot
  ○ Checksum for the rootfs
  ○ Idle mode and other modes of operation
Recipe App Layer

- Required Functionality is not necessary for the system.
- Required Functionality may be implemented as an application/service working in the user space.
- No bootloader/kernel source code/configuration update needed.
- Example:
  - Collecting and reporting the diagnostic information.
  - Service with configuration in /etc, preinstalled if required.
GitLab organisation

The public group Linux4Space (https://gitlab.com/linux4space):

Subgroups:

- **Linux4Space Requirements:**
  - A requirement model
  - The “human readable exports” from rexif and related xsl sheets

- **yocto:**
  - linux4space-meta
  - linux4space-meta-app
  - space-meta

- **Linux4Space yocto:**
  - Links together all the meta-layers and creates the working directory for the target HW
Reference hardware

- General useful and popular HW/emulators:
  - Qemu & x86
  - Raspberry Pi
- Space-ready HW:
  - The VZLU board
  - RISC-V (there is a space ready NOEL-V)
Ready to join?

Click Present with Slido or install our Chrome extension to activate this poll while presenting.
How to join?

- Fill the form at Linux4Space.org
Q&A