



Developing Linux Mobile Platform

Matti Salmi
23.10.2008

Public 10.10.2008 © Ixonos Oyj

Overview

Ixonos Mobile Linux Platform

Experiences in Linux development environment and tools

Experiences in management of Linux projects

Ixonos Linux platform

- The Ixonos Linux smartphone platform:

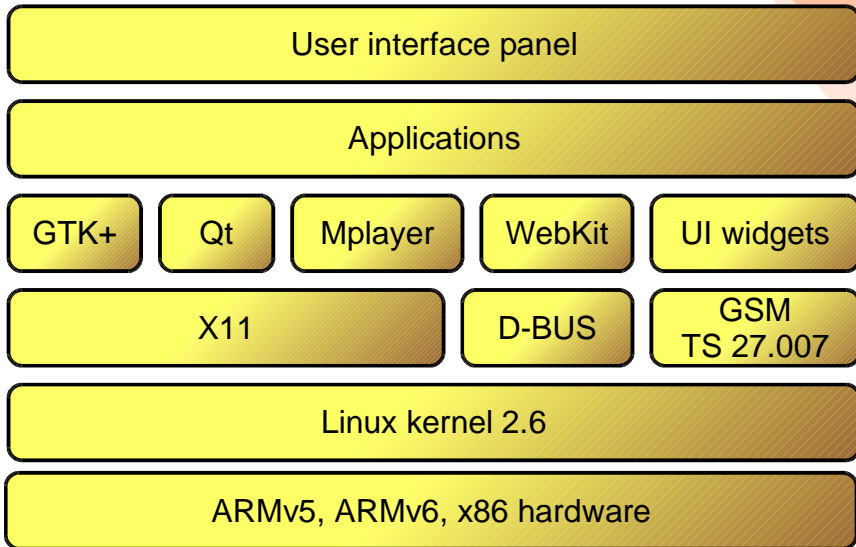
- Designs and implements a smartphone platform based on Linux
- Produces tools and components for Linux smartphone development
- Develops and extends Linux smartphone competences at Ixonos
- Contributes to the open-source community



Current platform applications

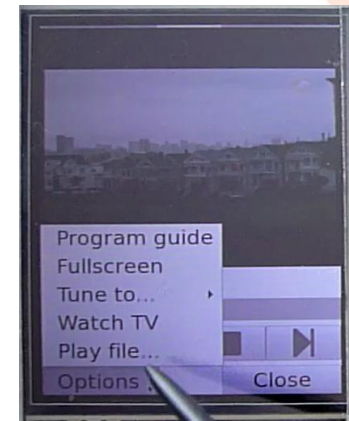


Linux platform details



Mobile television

- Technically, two main ways to deliver mobile TV
 - Via 3G mobile network
 - Dedicated broadcast network
- Benefits of DVB-H technology
 - Power saving due to burst transmission
 - Enhanced error correction
 - Bitrates, framerates of stream adapted to mobile devices
 - Handover features
 - DVB-T compliant transmission
- DVB-H is selected technology in EU
- Our mobile TV is based on DVB-H

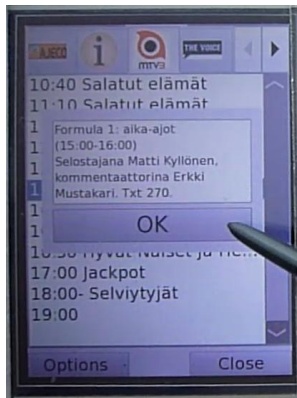


DVB Player in Linux Platform

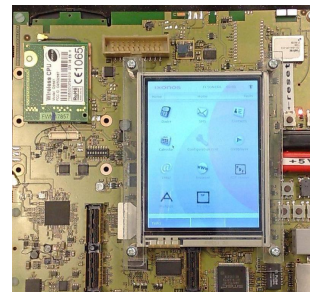
The DVB Player includes typical features that are found in DVB-T set-top boxes as well DVB-H compliant handsets.

The main features include:

- Channel tuning and selection
 - Playback of video and audio
 - Full ESG support
 - Playback of other media content
- DVB Player is based on Linux mplayer and the relevant codecs
 - DVB Player uses either commercially available DVB-T USB dongles or a dedicated DVB-H hardware



Variety of hardware platforms



Freescale iMX31



2530 EVM



OpenMoko

Contributions to open source community

GSMD2 – the GSM telephony daemon

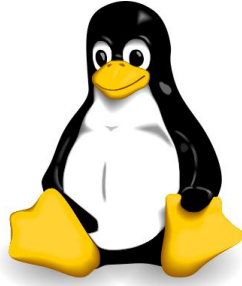
- Contributions to freesmartphone.org

FLAC codec

- Contributions to maemo.org

In addition, DVB-H support is planned to be contributed

- Planned contributions to linuxtv.org



Ongoing development

- Enhanced touch-based UI solutions
- Qt –port and UI base
- OpenGL based accelerated graphics
- Testing and test automation enablers



Experiences in Linux Development Environment

Software Development in Linux

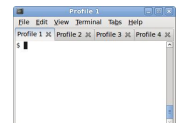
In Microsoft Windows, MS Visual Studio prevails

- Provides integrated compiler and IDE

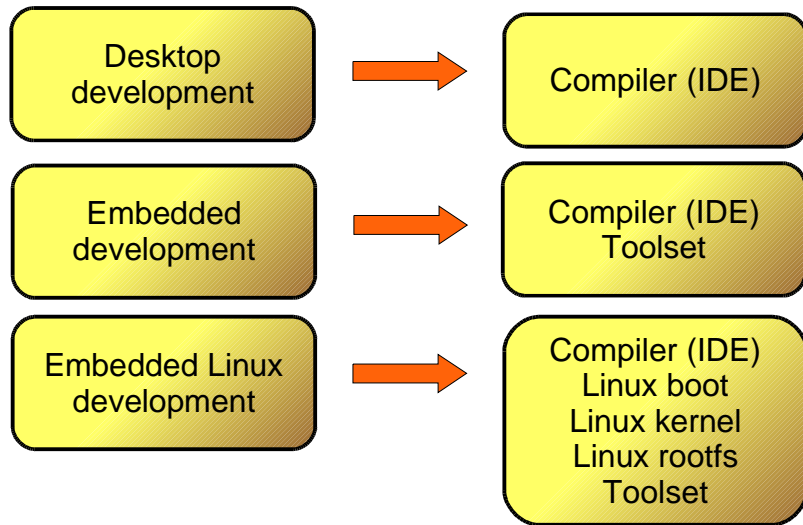


In Linux development, GNU Compiler Collection (GCC) is used most often

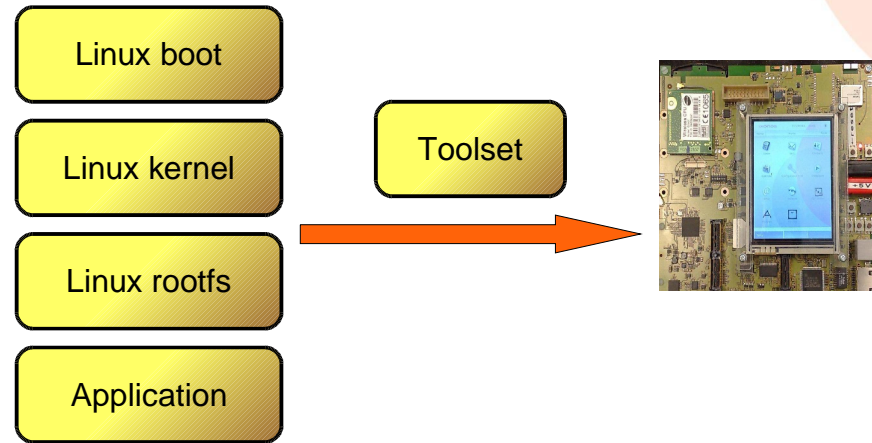
- Compiler only, but no IDE
- Separate IDEs, such as Eclipse, Anjuta, Kdevelop
- Development without IDE is very common method



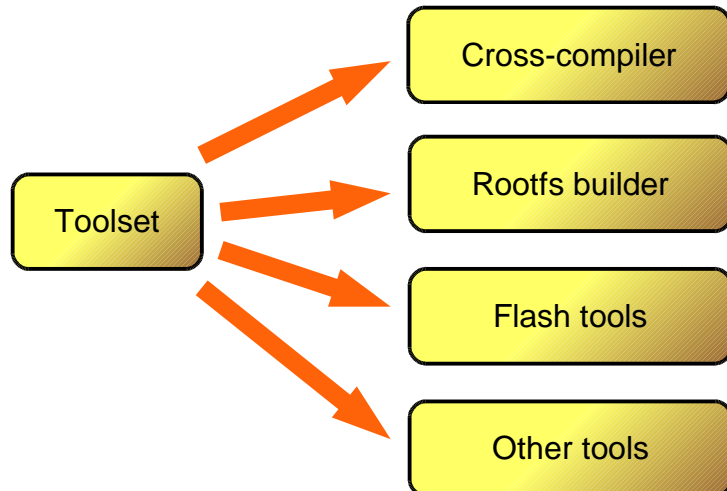
Linux as development environment



Embedded Linux development



Toolsets in Linux development



Documentation in Linux Platform

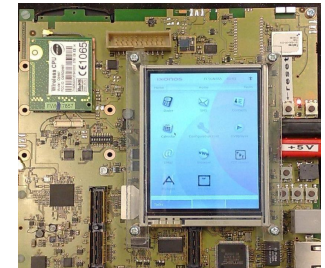
- Openoffice in Linux is a stable documentation toolset with
 - Writer – for documents
 - Impress – for presentations
 - Calc – for spreadsheets
- Openoffice is compatible with Microsoft formats
- Other documentation includes tools for pictures and diagrams
 - GIMP for still images
 - Inkscape for diagrams and vector graphics
- Code documentation tools:
 - Doxygen



Experiences in Project Management

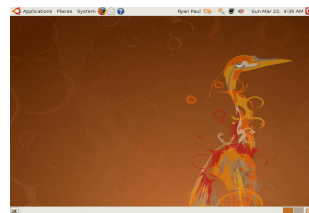
Development for target environment

- Building target environment is hard and complex task
- Avoid postponing of target environment building
- In many cases, only some part of people can solve building problems in target environment
- Documentation of building must be detailed
- Releasing need to be planned in detail
- Frequent releasing with small steps to avoid complex integration problems
- Plan enough time to target environment setup



Development in desktop environment

- Necessary to speed up development
- Can be postponed to later stage
- Often, just simple nested X11 server such as Xephyr is adequate
- High-level connectivity is easy (net, USB, serial, ...)
- Low-level connectivity may be problematic (I2C, SPI,...)
- Release separately from target environment
- Avoid cluttering code with #ifdef's

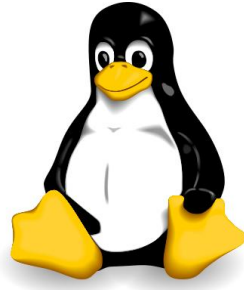


Personnel Management

- It is easy to assume that a Linux developer should master everything from kernel drivers to UI development, but:
 - Like in Windows – people should be able to specialise and use their deep knowledge in their area
 - Avoid "Linux Guru" aspect – all people should not be required to be capable of handling all things
 - Do not try to understand all technical details yourself
- It is easy to let Linux developers to decide how the documentation is done, but:
 - Documentation is often seen unnecessary, "code talks"
 - There are plenty of special Linux tools, but they are often tedious to use and result in uneven output
 - Instead, define and force limited set of mainstream document tools

Community Work

- Participate work where you have technical interests and plans
- "Code talks", avoid political aspects
- Long-time involvement needed – not just one month participation
- Requires technical strategy for open source participation
- Community work should not be "hobby" but resourced as normally
- Participants represent their company, but indirectly
- Licensing cannot be compromised – what you do is going to be open



Summary

- Setting up embedded Linux environment is time-consuming and requires strict releasing mechanisms
- Participation in open source communities requires interest, planning and resourcing – not just hobby
- Ixonos has gained good experience in embedded Linux development with Ixonos Linux Platform

Thank you

www.ixonos.com