Embedded Linux Systems

Presented By:

Kitrek Riese

Presentation Overview

- Why Choose Linux
- Embedded Linux Architecture
- Options and Attributes of Embedded Linux
- Embedded Linux Examples

The Start of Embedded Linux

- At first there was VxWorks, pSOS, Neculeus, and Windows CE
- Although now a common practice, the first concepts of embedded Linux arrived on the scene around 1997.

Why Choose Linux

- Linux is Open Source.
  - Additions to functionality are made easier.
  - Large support network for developers.
  - Ability the fix code without help from outside organizations.

Why Choose Linux

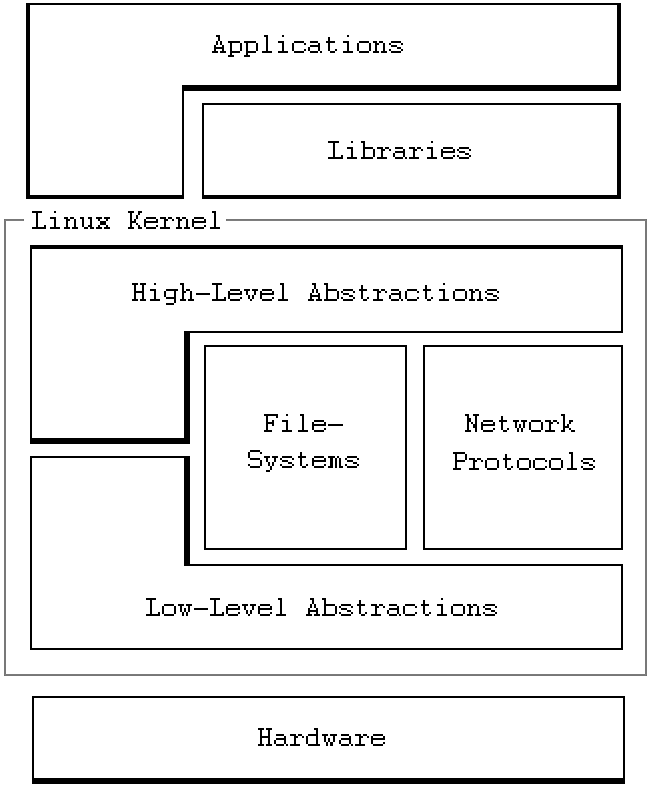
- Quality and Readability of Linux Code
  - Separate functionality is found in separate modules and the different modules are split into different files, reflecting their functionality. The result of this is high cohesion and low coupling.

Disadvantages of Linux

- Real Time Performance is provided through Kernel Modules
  - A code error can crash the entire OS

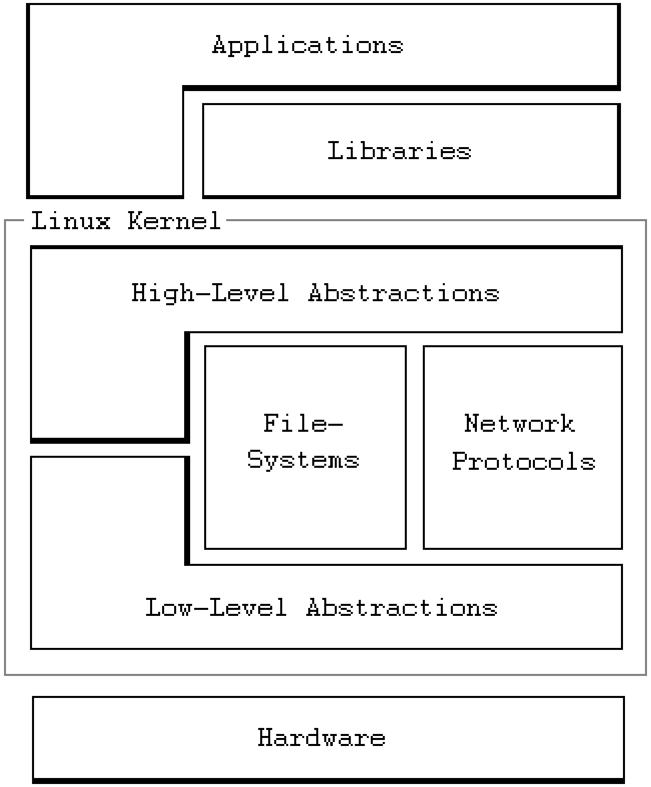
The Linux Architecture

- Same as ‘normal’ Linux Architecture
- Contains six layers of abstraction



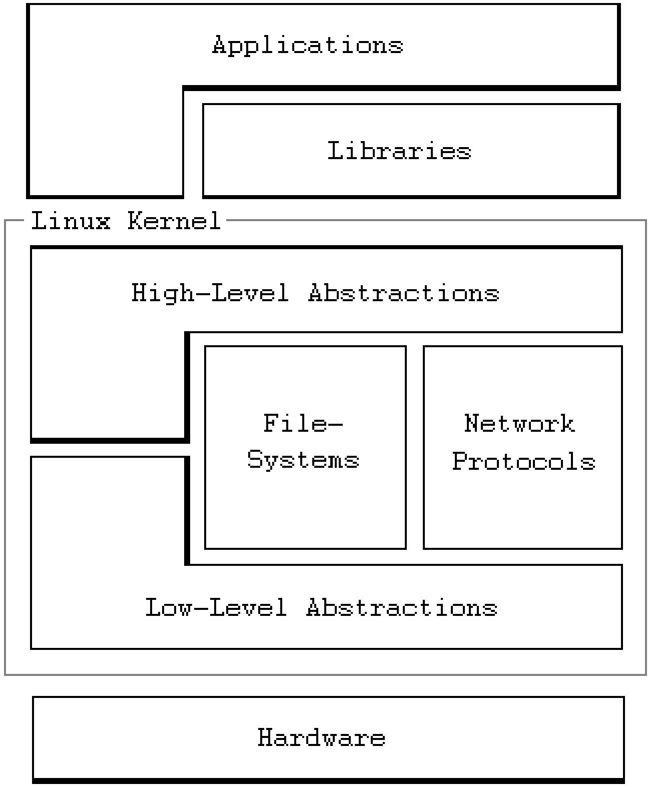
The Linux Architecture

- Applications: Includes your normal desktop applications



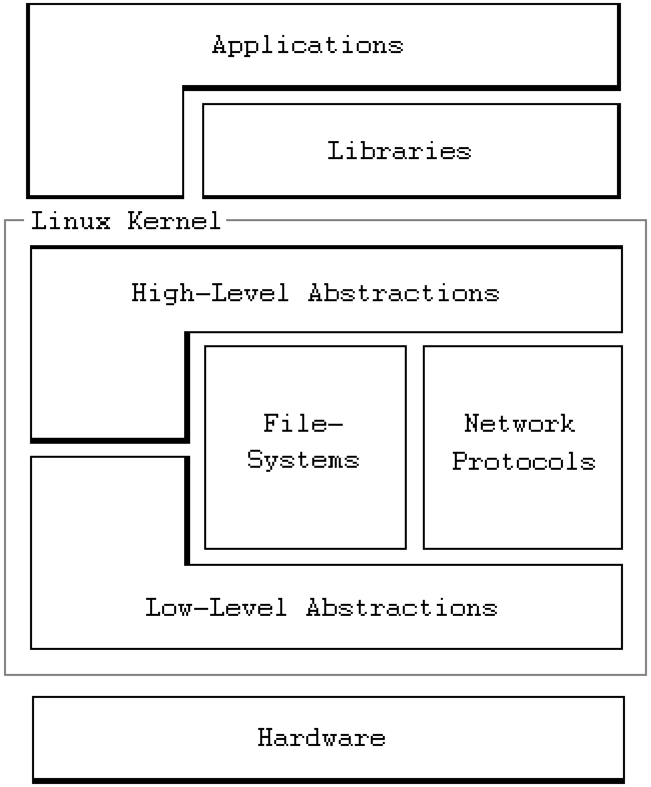
The Linux Architecture

- Libraries: Allow applications to communicate with the Linux Kernel
  - GNU C library
  - Qt
  - XML
  - MD5
- These libraries are not part of the application but rather exist separately within the Linux structure.
- Different applications to use the same instance of a library. This saves memory because you need only one copy of the library to be loaded into RAM



The Linux Architecture

- The Linux Kernel
  - Essentials:
    - High-Level Abstractions
    - Kernel Internals
    - Low Level Abstractions
  - Internal abstraction gives the Kernel ability to interface with many File-Systems and Network Protocols



Embedded Linux Necessities

- Minimalist Linux
  - A boot utility
  - The Linux micro-kernel, composed of memory management, process management and timing services
  - An initialization process
- To doing anything useful while remaining minimal, you also need to add:
  - Drivers for hardware
  - One or more application processes to provide the needed functionality
- As additional requirements become necessary, you might also want:
  - A file system (perhaps in ROM or RAM)
  - TCP/IP network stack
  - A disk for storing semi-transient data and swap capability
  - A 32-bit internal CPU (required by all complete Linux systems)

http://www-106.ibm.com/developerworks/linux/library/l-embl.html

Embedded Linux Hardware

- Many Hardware Options Available
  - x86
  - ARM
  - IBM/Motorola Power PC
  - MIPS
  - Hitachi Super H
  - Motorola 68000

Embedded Linux Hardware

- x86
  - Introduced by Intel in 1985
  - Large amounts of documentation available
  - Most widely used and tested on Linux platforms
  - Many applications developed on the x86 for the purpose of porting to other hardware.

Embedded Linux Hardware

- ARM (Advanced RISC Machine)
  - No particular manufacturer, maintained by ARM Holdings Ltd.
  - ARM Holdings designs CPU cores and the ARM instruction set
  - Customers find their own manufacturer which allows for greater customization
  - Linux supports ten different ARM CPUs on sixteen different platforms, along with two hundred related boards.

Embedded Linux Hardware

- Busses and Interfaces
  - PCI
  - PCMCIA
  - Parallel Port
  - SCSI
  - USB
  - FireWire
  - Serial Port

- Linux also supports a wide variety IO devices
  - Printers
  - Mice
  - Sound
  - Storage
  - Display Devices

Development Tools and Environments

- Required Software
  - Bootloader
  - Build-Tools
  - Debugging Tools
  - Linux Kernel

Development Tools and Environments

- GNU cross-platform development toolchain
  - Kernel Setup Software
  - Binary File Manipulation Utilities (Binutils)
  - C Library (Can be replaced with others)
  - Full C Compiler

Development Tools and Environments

- Available IDE’s
  - Anjuta
  - Eclipse
  - Glimmer
  - KDevelop
  - SourceNavigator

The Linux Kernel

- Development of Embedded Linux does not coincide with the development of the Kernel
- Each architecture will have its own Kernel.

The Linux Kernel

- Configuring the Kernel
  - Loadable Module Support
  - Network Options
  - SCSI Support
  - File Systems
  - Sound
  - Console Drivers

The Linux Kernel

- Configuration Options
  - ‘make config’ – Command Line
  - ‘make oldconfig’ – Premade Config still command line but faster
  - ‘make menuconfig’ – Text Based Config
  - ‘make xconfig’ – X Window Based Config

Root File System

- bin
- boot
- dev
- etc
- ~~root~~
- sbin
- tmp
- usr
- var
- ~~home~~
- lib
- ~~mnt~~
- ~~opt~~
- proc

Bootloader Concepts

- The bootloader loads the Linux Kernel
- Thousands of choices in bootloaders to choose from and many configurations of each
- Many architectures have well known or commonly used bootloaders
- Some architectures have no standard bootloader

Bootloader Concepts

- X86 Bootloaders
  - GRUB (Grand Unified Bootloader)
  - LILO (Linux Loader)

  - ROLO
  - loadlin
  - Etherboot
  - LinuxBIOS
  - blob
  - PMON

Networking Services

- Linux Networking Features
  - SNMP - Remote Administration
  - Network Login through Telnet
  - SSH
  - HTTP
  - DHCP

Networking Services

- SNMP
  - Almost every device which connects to a TCP/IP network has SNMP
  - Net-SNMP
    - Free
    - Large
    - SNMP agent is all that is necessary

Networking Services

- HTTP
  - Apache HTTP Servers
    - Not for embedded systems
  - Boa
  - thttpd

Networking Services

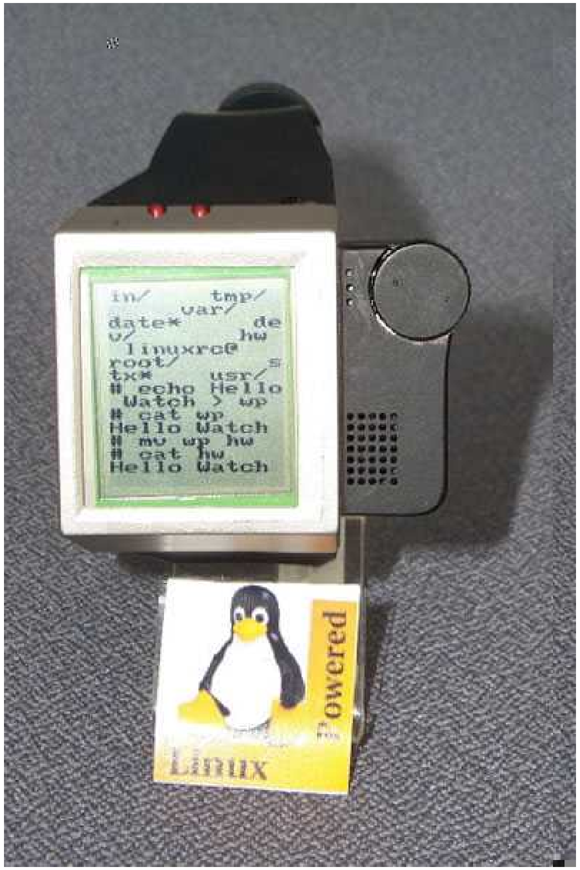
- DHCP
  - Dynamic Host Configuration Protocol
    - Automatic network configuration of hosts
    - Standard DHCP package is free with most Linux distributions

The Linksys Wi-Fi Router

- Linux gives us "the premium OS for inexpensive, feature-packed wireless networking." (Ewing)

The IBM/Citizen Linux Watch

- 56 X 48 mm
- Motherboard – 27.5 X 35.2 mm
- 8 MB Flash Memory
- 8 MB DRAM
- Touch Sensitive Display
- 1.5 oz
- 32-bit RISC Processor (74-18 MHz)



```
in/      tmp/
     var/
date*      de
v/        hw
 linuxrc@
root/       s
tx*     usr/
# echo Hello
 Watch > wp
# cat wp
Hello Watch
# mv wp hw
# cat hw
Hello Watch
```

How is This Possible?

- A fully featured Linux kernel requires about 1 MB of memory
- The Linux micro-kernel actually consumes only 100 K

http://www-106.ibm.com/developerworks/linux/library/l-embl.html

How is This Possible?

- With the networking stack and basic utilities, a complete Linux system runs quite nicely in 500 K of memory on an Intel 386 microprocessor, with an 8-bit bus
- A Linux system can actually be adapted to work with as little as 256 KB ROM and 512 KB RAM

http://www-106.ibm.com/developerworks/linux/library/l-embl.html