Embedded Linux

Standards, Challenges, Obstacles

A Red Hat Perspective

Manfred Hollstein

redhat

April 25, 2001

redhat

Embedded Linux – Is It Real?

redhat

Internet Connectivity is Driving Post−PC Computing

- 1 in 5 Japanese (over 27 million) has an internet connection. More than 10 million use mobile connections to the Internet.
- By 2003, more than 50% of the people using the Internet will do so with handheld and/or portable (Post−PC) devices (IDC).
- By 2002, more than 55 million Post−PC devices will ship, and by 2005 shipments will exceed shipments of PCs (IDC).

Square Pegs/Round Holes

- Proprietary RTOSes have been successful in special−purpose devices that have limited application support
- IA−32 appearing in some designs
- PocketPC has been offered in sub−PC configurations
- The Post−PC market remains an open field

Many−to−Many Aspects of Open Source Software

- Open source compilers and debuggers run on many host platforms and support many target architectures
- Open source operating systems support many CPUs and embedded boards
- Open source simulators help embedded designers partition hardware and software functionality

Linux

- Linux supports target−as−host development model
- Linux supports many popular CPUs
  - ✔ IA−32
  - ✔ PowerPC
  - ✔ ARM, StrongARM
  - ✔ MIPS
  - ✔ Alpha
  - ✔ SH−3
- Linux is extensible

Embedded vs. Desktop

- It is rarely possible to create compelling applications only by stripping away functionality
- Reductions must result in improvements:
  - Size
  - Power
  - Speed
  - Cost
  - Operation Environment

Embedded Linux

- Boot parameters
- Board parameters
  - Networking
  - Filesystems
  - Terminal/Graphics Support
  - Device Support
- Run−Time environment
- Configurations for multiple applications

Fragmentation

- Application−level
- Platform−level
- Development−level
- What’s the difference between innovation and fragmentation?

Configurability

- Configurability can solve different kinds of fragmentation
- Configurability is not a magic bullet
  - Complex configuration tools can be more difficult to manage than fragmentation
- Is configurability a means or an end?

Standards

- Standards come in many flavors
  - De facto
  - Reference implementation
  - Specification
- Standards nail down things that cannot be changed (usually interfaces) so that other things (usually implementations) can be changed
- Good standards are hard to find
  - And even harder to maintain

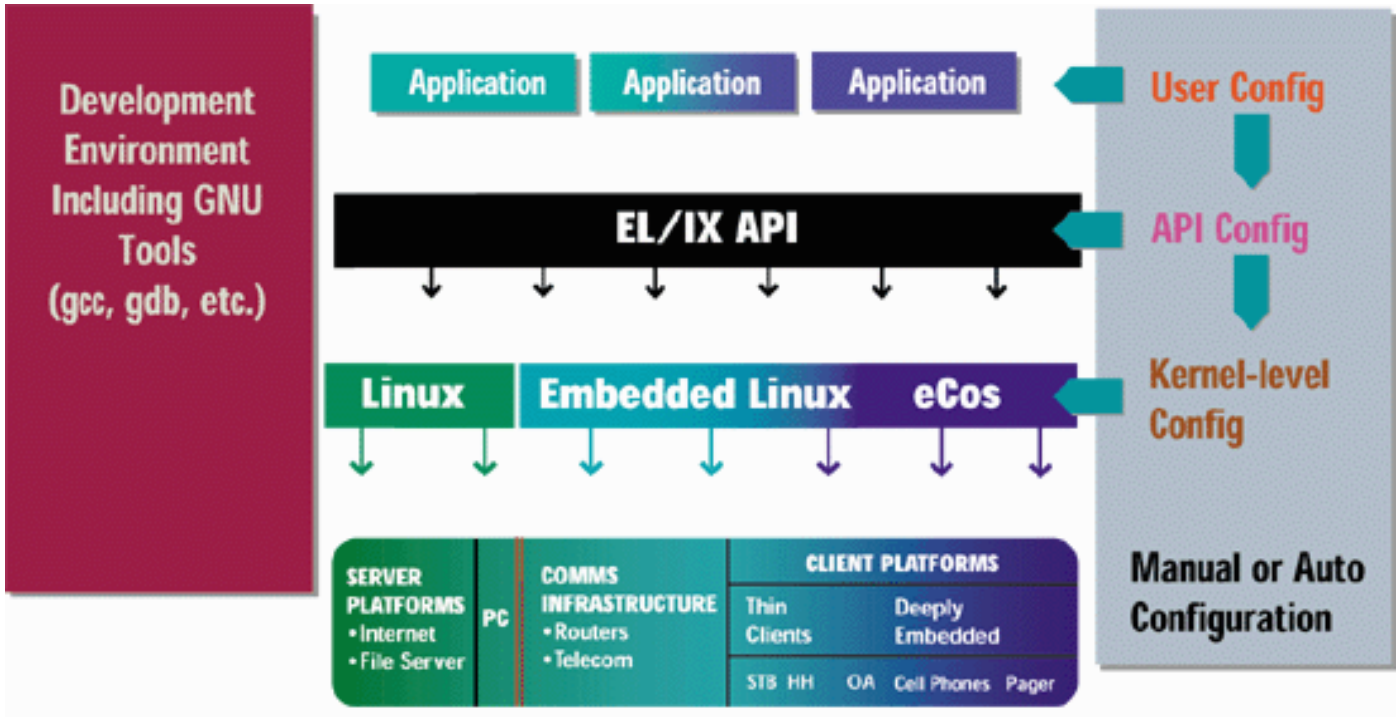
POSIX (ISO 9945)

- "Portable Operation System Interface"
  - Application−oriented
  - Interface, Not Implementation
  - Source, Not Object Portability
  - Minimal Interface, Minimally Defined
  - Broadly Implementable
- IEEE Std 1003.1, 1996 Edition
  - Part 1: System API
  - Part 2: Shells and utilities
  - Part 3: System Administration

POSIX.13

- September 1999
- POSIX.13 presents four real time system profiles
  - Minimal Realtime System Profile
  - Realtime Controller System Profile
  - Dedicated Realtime System Profile
  - Multi−Purpose Realtime System Profile
- EL/IX presents four "levels" (***E***mbedded ***L***inux based on POS***IX***)

EL/IX – Linux based open standards for embedded development



Find details at http://sources.redhat.com/elix/

redhat

Goals for the EL/IX API

- Support development of embedded applications using the Linux desktop environment as both host and target platforms.
- Provide scalability of that functionality according to the requirements of the embedded application.
- Provide portability of applications between operating systems that provide the same levels of API functionality.
- Provide real time functionality on real time operating systems.
- Respect existing standards by adopting their APIs and functionality where appropriate, but also adopt a pragmatic approach to compatibility.

EL/IX Levels and Options

- **Level 1:** *RTOS compatible layer.* Functions available in both Linux and a typical deeply embedded operating system (eCos , RTEMS, VxWorks, pSOS, VRTX32 etc.). Some functions may have reduced or modified semantics.
- **Level 2:** *Linux single process only.* Includes level 1 plus any functions from Linux that are not easily implemented on an RTOS. Also "full" implementations of reduced functions in Level 1.
- **Level 3:** *Linux multiprocess for embedded applications.* This is basically POSIX.1 with some of the functions that are obviously not for embedded applications (such as job control) removed.
- **Level 4:** *Full POSIX or Linux compliance.* Essentially these are functions that are present in a standard Linux kernel but are irrelevant to an embedded system. These functions do not form part of the EL/IX API.

PE51 vs. Level 1

- Basic functionality for single−process deeply embedded system
- Signal and some memory management options enabled

EL/IX Level 1 is a flexible subset of PE51

PE52 vs. Level 2

- Extends PE51 with support for a filesystem and asynchronous I/O
- Signal, some memory management, and filesystem options enabled

EL/IX Level 2 is a flexible subset of PE52

PE53 vs. Level 3

- Extends PE52 with support for multiple processes, richer filesystem capabilities
- Signal, memory management, and filesystem options enabled

EL/IX Level 3 is a flexible subset of PE53

PE54 vs. Level 4

- Superset of other PEs, containing essentially POSIX.1, POSIX.1b, and POSIX.1c standards
- Remove GNU, BSD, and SYSV compatibility

EL/IX Level 4 is a strict superset of PE54
EL/IX Level 4 is exactly Linux

Real−Time Linux

- Much work going on
  - FSM Labs has working PE 51 w/realtime performance (down to 5 microsecond response time)
  - TimeSys experimenting with Priority Inversion
- Realtime Linux market may be limited
  - Telephony (not exactly small...)
  - Multimedia (video, audio, etc.)

Memory Mangement

- Option for some, not others
  - Silicon and cost overheads
  - Security and Reliability
  - Real−time
- POSIX defines interfaces
- Red Hat is actively working to make hardware MMU an option in the "standard" 2.4 kernel

Putting It All Together

- Many vendors combine Linux, configuration tools, libraries, board support, tools, and sample applications
- Customers are all asking for the same thing
  - Reference implementations
  - Tools
  - ISV and IHV support
  - Standards

Application 1: SAN

- Storage−attached networks depend on complex OS functionality
  - Cluster management
  - Filesystem management
  - Volume management
  - SMP performance
- Flexible client/server partitioning favor an OS strategy that is more common between client and server

Application 2: Web Phone

- Portability and battery life are key
- Must support wireless stacks
- DSP functionality
- Web browser
- Graphics
- Embedded storage devices
- Minimal memory footprint

Application 3: SOHO Router

- Must support complex broadband protocols
- Configurable to support new protocols
- Remote manageability
- Configure to support application partitioning (N−tier Post−PC architecture)
- Environmental concerns require low−power CPUs

Application 4: PDA

- Wireless support
- Graphics
- Browsers
- Data synchronization
- Remote manageability
- Security

Application 5: Servers!

- i−mode Phones in Japan sell over 5 million units in less than six months
- i−mode phone production nearly halted because *server capacity* is exceeded
  - Initial installation not powerful enough to judge fad from trend
  - IT staff not able to install machines fast enough
  - Server facility not large enough to handle low−density web serving solutions

Summary

- The Post−PC market is wide open
- Linux has the flexibility, configurability, and functionality to support a very wide range of applications and appliances
- Control standards by contributing to them
- Linux standards should be 100% open source software
- Register your interest to participate in the development of EL/IX