



QNX[®] Build
a more
reliable
world[™]

The Role of Linux in Standards-based Real-time & Embedded Computing

Steve Furr

QNX Software Systems

Background

- QNX Software Systems
 - Leader in real-time operating systems based on industry standards (POSIX)
 - Latest offerings approach full API compatibility with Linux
- Steve Furr
 - Senior Developer
 - Representative to Embedded Linux Consortium
 - Core member of Real-Time Specification for Java experts group

Premise

- Linux will emerge as the market share leader of standards-based operating systems for the real-time and embedded markets
 - Freely available open source implementation
 - Additional supported platforms
 - Adapt to suit requirements
 - Based on industry standards (POSIX, Unix, IETF)

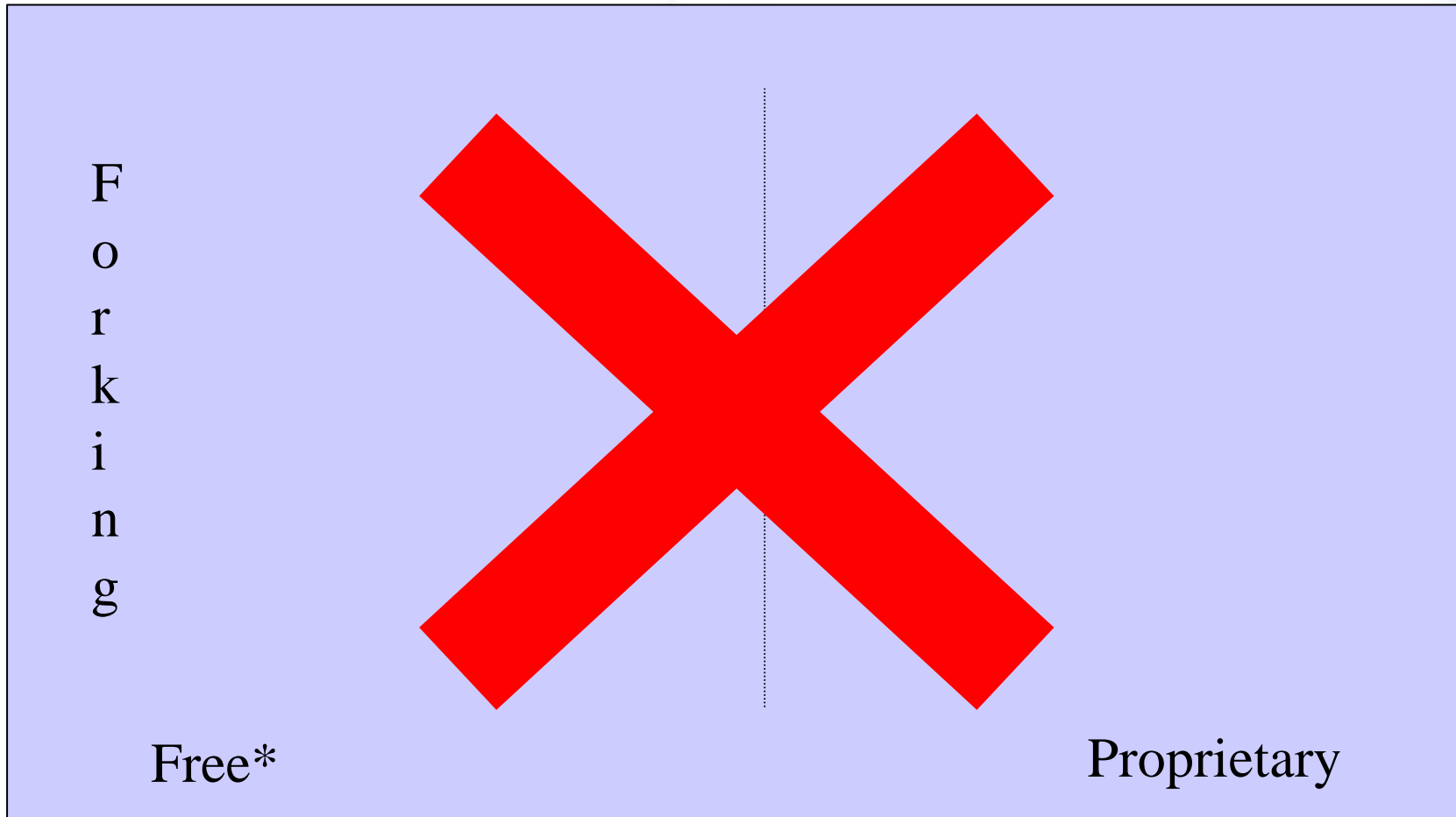
Concerns

- What do we mean when we say Linux?
 - Single definable market
- Will de jure or de facto standards prevail?
- Will API compatibility with existing standards remain in all domains?
 - Will device driver compatibility remain?
 - Will a single code base be retained?
- Will ongoing development recognize the relevance of standards bodies?
 - Or will it set the standards?

What is Linux?

- Composition
 - Linux kernel
 - GNU/Linux
- Implementation vs. Interface
 - Linux as a Technology
 - Doesn't distinguish between API and implementation
 - Goal: avoid forking of technology
 - Result: Potential forking of APIs

Forking -- A Linux View



Forking -- Problems

- Pre-supposes correlation between openness and forking
 - Some evidence would suggest that no such correlation exists
- Ignores other important factors:
 - Need for critical mass of applications and developers to build Linux as a platform
 - Platform development driven by compatibility with existing practice and standards
 - Social pressures of Linux development community
 - May weaken as Linux is pushed into different domains

Forking - Flocking Model

- Flock:
 - Vendors, developers & end users behave like a flock of birds
 - Dynamic changes in behaviour from a common starting point
 - Direction changes result from complex socio-economic forces
 - e.g. market forces, such as niche market requirements
- Role of Open Source
 - Affects mobility of flock

Market Forces -- Real-time

- Address the need for guarantees that the system can meet the timeliness constraints associated with operations required by a real-time system for correct behaviour
- Standard Linux kernel doesn't provide POSIX real-time facilities
- Generally provide real-time facilities with a separate real-time kernel
 - Acts as a host for the standard Linux kernel
- Alternatively, provide POSIX real-time threads, timers and other facilities with a derivative of the kernel

Market Forces -- Embedded

- Scalability
 - Ability to scale facilities down to match the required capabilities and footprint of the device
- Configurability
 - Ability to support different run-time configurations of a given platform
- API compatibility
 - Profiles and feature testing
- Tools-based approaches
 - ABI compatibility using embedding tools

Linux Real-time APIs

- Orthogonal set of APIs providing real-time facilities
- Unifies existing real-time kernel hosted Linux implementations
- No uniformity with industry standards incorporating real-time (POSIX .1b, .1c, .1d, .1j)
- Interaction between real-time and regular threads aren't well defined

EL/IX

- Embedded Linux/POSIX API for real-time and embedded Linux
- POSIX 1003.13 minimum configuration, plus networking, BSD library and GNU library extensions
 - Multiple levels of conformance
- No additional API extensions
- Preserves API compatibility with POSIX real-time facilities
- Requires modification to Linux kernel code base
 - Not currently available

Process

- Define a process that formalizes practice within the Linux community
 - Enable evolution of Linux technology with continued compatibility
- Enable the evolution of specifications to match ongoing development
 - Feedback mechanisms
 - Lightweight
 - Proceeds at the pace of technology
 - Low barriers to entry

Linux Community Process

- Project-oriented
 - Focus on implementation
- Contribution-based
 - Ad hoc membership
 - Consensus building
 - Ability to contribute
 - Forward progress in implementation
- Informal Specification
 - Implementation considered specification
 - No implementation == No specification
 - Fluid

Acceptance Criteria

- Open implementation
 - An implementation based on Linux technology must be available
 - No encumbrances -- higher standard for IP
- Innovation over invention
 - Codifying existing practice yields maximal results
 - esp. standard Linux kernel
 - Extensions outside of the kernel
- Low barriers to entry
 - Any party materially affected and willing to contribute must be able to participate

Lessons & Observations

- Ad hoc process doesn't necessarily yield faster results
 - KDE, Gnome, Linux timelines to viability
- Open participation isn't incompatible with focused effort
 - Projects usually include an architecture board overseeing efforts, based primarily on resource availability
- Formal balloting isn't a precondition to achieving consensus
 - Adoption is proof
- Open-ness in specification doesn't have to introduce delays

Role

- Promote adherence to existing API standards across domains
- Develop standard profiles encompassing both POSIX and GNU/Linux facilities
- Foster community involvement
 - Joint working groups
 - Cross-organization membership
 - Process adaptation and optimization
 - Provide an open home for specification IP
- Fast-track community developed specifications