Why RTLinux?

"Realtime" is is an over-used and often ill-defined term. One frequently hears it used to mean "right away" or "fast," as in "realtime stock quotes." But "hard realtime" is a precise term—it refers to processes with timing deadlines that cannot be missed.

There is no general-purpose realtime computer operating system (RTOS). While there are a variety of proprietary realtime solutions, each has drawbacks—price, platform restrictions and nonstandard APIs, to name a few.

Extensively tested and freely available, Linux is an ideal candidate for extension into realtime territory. With thousands of developers from around the world making contributions to its development, Linux is a solid, well-supported, multi-platform solution.

The most reliable realtime programming model

The traditional method of mixing RT and non-RT code leads to unmaintainable systems with errors that are hard to detect but that can have catastrophic effects in operation. RTLinux™ from FSMLabs solves this problem by decoupling realtime and non-realtime software components.

The RTLinux™ programming model is simple: Anything that has strict timing requirements should be written as a thread or signal (interrupt) handler, and anything that doesn't need hard realtime should go through regular Linux. This allows us to keep the RT side small—and as fast as the hardware will permit—while still drawing on Linux for sophisticated services and applications.

RTLinux™ is a small, deterministic, RTOS that is somewhat like a single POSIX thread sitting on a bare machine. Hard realtime applications are threads and signal handlers in this process. The regular Linux kernel runs as the lowest-priority thread of RTLinux™ and is always pre-emptible.

System Features

Developed and supported by FSMLabs, Inc., the inventors of hard realtime Linux, RTLinux™ provides a number of features that make it ideal for all your realtime programming needs:

Deterministic Realtime: RTLinux supports realtime interrupt handlers and realtime periodic tasks with interrupt latencies and scheduling jitter close to hardware limits. RTLinux™ reliably give worst case interrupt latencies of less than 15 microseconds on a standard x86 PC, with better figures on single board computers and PowerPC and Compaq Alpha platforms. (Don't be misled—"worst case" interrupt latency is not the same as "typical"!)

Standard API: FSMLabs RTLinux™ uses a POSIX threads API and follows the POSIX 10003.13 PSE31 ``minimal realtime system model" specification. RTLinux™ runs Linux as its lowest priority thread and provides access to the full power of Linux through a variety of communication methods.

Open Source: FSMLabs RTLinux™ can be freely downloaded from the Internet. The core OS code is available under the GNU Public License, but applications and drivers running on RTLinux are *not* obligated to follow the GPL. FSMLabs will also sell non-GPL licenses to customers needing to protect proprietary changes to RTLinux code.

Rich toolset: All the standard tools—the GDB source level debugger, GNU compilers, Perl, TCL/TK, etc.—are available and easily interfaced with realtime code. In addition, there is a growing list of RTLinux-specific open source tools: from the powerful RTiC instrumentation lab to the CarbonKernel RTLinux simulation for prototyping and validation. There are also numerous vendors of non-open source, RTLinux-specific code, including the CAD-UL RTLinux debugger and Quanser's MatLab/Simulink tool for generating RTLinux controllers from MatLab equations.

Support and maintainance from the industry leader: FSMLabs, Inc. created hard realtime Linux and has taken this patented technology from the research lab to industry. With years of experience with Linux and RTLinux, FSMLabs is a core technology company, with contracts with Compaq, IBM, Maxtor, the Jim Henson Creature Shop and other industry leaders. Additional support is available from MontaVista Software.

"We chose RTLinux because of stability, networking, user community, extenstive tools and price. We have been unable to find [another] solution that offers the abilities of RTL that is even remotely close in price... but I suppose you already know all of this."

—John van Loon
Oregon Cutting Division/Blount, Inc.

A System for Every Need

Three versions of RTLinux™ are available:

RTLinux V1 is exceptionally stable. It provides a simple API for scheduling realtime tasks;

RTLinux V2 offers a simplifed version of the POSIX pthreads API and is compliant with the POSIX "minimal realtime" standard. V2 was released in November, 1999 and is an SMP-capable x86 system; and

RTLinux V3pre6, released in June, 2000. V3 supports x86 (uni-processor and SMP), and PowerPC and Alpha (uni-processor only). Based on Linux 2.4.0test1, it features many performance improvements, including:

- Mutex priority ceiling support
- Spinlock mutexes
- Conditional variables
- Mixing v1 and v2 APIs

All versions of RTLinux are available for free download at ftp://ftp.fsmlabs.com.

We're Not Alone

RTLinux™ is used by a wide and growing variety of industrial and scientific organizations. Current FSMLabs clients include:

CosineCom. Uses RTLinux in telecommuncations hardware sold to Internet service providers and other ecommerce-based businesses.

SynergyMicro. Maker of embedded system boards for SMP. RTLinux development on PowerPC boards.

Compaq Computer Corporation. A port to the Alpha AXP architecture.

IBM. Development of PowerPC Linux for servers platforms.

Sandia National Labs. RTLinux technology development.

Jim Henson Muppet Works. RTLinux general technology development.

Oregon Chain. An international manufacturer using RTLinux to automate production in chain manufacturing.

Kaiser Aluminum. Control of roller mills.

Lang GmbH & Co. Control of 3D engraving machine tools.

NASA. Instrumentation control.

NIST (National Institute of Standards and Technology). Machine tool control.

You're Not Alone

The core development group at FSMLabs are groovy guys who do lots of really cool things with computers.

Victor Yodaiken , director and founder of FSMLabs, has nearly 20 years of experience in operating systems. He helped design and build one of the first commercially-distributed fault-tolerant UNIX systems for Auragen in 1982. As a professor of computer science, he has worked in fault tolerance, OS architectures, realtime, OS optimization, configurable computing, cluster operating systems and mathematical validation of operating system correctness. He is responsible for the design and technical basis of RTLinux.

Cort Dougan is the primary developer and gatekeeper for PowerPC Linux and has coordinated a distributed development effort involving thousands of programmers, millions of users, and many companies.

Michael Barabanov was the original implementor of RTLinux. He has extensive experience in operating systems and instrument control.

Nicholas McGuire is groovy guy who does lots of cool things with computers. He is a groovy guy who does lots of cool things with computers. He is a groovy guy who does lots of cool things with computers. He is a groovy guy who does lots of cool things with computers.

FSMLabs
P.O. Box 1822
Socorro, NM 87801

yodaiken@fsmlabs.com
http://www.fsmlabs.com
http://www.rtlinux.com

Finite State Machine Labs

RTLinux

Hard Realtime plus Linux

for—

Industrial Automation
Telecom
Data Communications
Instrumentation & Control