



Blog Entry Quality Magazine

Date: 8-11-2009
Author: Ned Lecky
Lecky Integration
ned@lecky.com
Work/Cell: 518-258-5874

Real-Time Linux

“Controlling a laser with Linux is crazy, but everyone in this room is crazy in his own way. So if you want to use Linux to control an industrial welding laser, I have no problem with your using PREEMPT_RT.” -- Linus Torvalds

A few more months of Linux development are under my belt since my last blog entry, and I must say I'm pretty impressed. This stuff is not for the tiny embedded systems that many of us work on every day, but for the next level up in performance and capability, Linux is an awfully appealing option.

But what Linux? Since the early days of UNIX, there have always been many flavors available. How do you choose?

Furthermore, for embedded systems, a real-time response element is almost always required. So you really need to look at a Real-Time Operating System (RTOS) based on the Linux kernel, a generic area that people refer to today as “RT Linux.”

There are basically two ways to get there. You can build your own system using the open source Linux RT patch. This is a constantly-evolving kernel upgrade that adds a plethora of real-time features, such as priority control, forced kernel preemption, high-resolution timers, latency control, and event timing to the “standard” open-source Linux kernel. If you're an operating system aficionado, you'll have great fun getting all of these bits and pieces and playing with them. For more information, check out rt.wiki.kernel.org.

For those of us under time and debug pressure, there are some faster alternatives. More and more vendors are developing “complete” RT Linux products that are preconfigured and tested for your own embedded computer system or Single-Board Computer (SBC). These vendors have done the heavy lifting and due-diligence required to integrate the kernel, the RT technology, the device I/O and drivers, the development tools, and the communications protocol stacks. Using a complete package like this saves literally weeks up front and untold numbers of hours of debugging later on.

There are several big players here. MontaVista (www.mvista.com) really wrote the book on RT Linux and has been providing complete products with well-integrated Board Support Packages (BSPs) for the longest. Wind River (www.windriver.com) is a familiar player in the RTOS scene known for their proprietary VxWorks RTOS. They were just acquired by Intel and we're still waiting around to see what that means. But they have also released a RT Linux product that looks like it will compete well in the space. LynuxWorks (www.lynuxworks.com) is another well-known RT Linux vendor with well-supported and documented products. Embedded Alley (www.embeddedalley.com), a group of MontaVista escapees, was just acquired by Mentor Graphics and provides custom RT Linux for a variety of platforms. And there are more players almost every day.



With all of this development and acquisition activity, it is clear the RT Linux is bubbling up higher and higher on embedded radar screens. And in spite of Linus Torvalds' opening quote, RT Linux is now creeping not only into laser control, but into industrial control, medical systems, and even safety and flight-critical systems. It is not suitable for every application under the sun, but is a great alternative when you wish you had a more standard OS but can't even THINK about big OSs like Windows. Try it out in your lab and see if you get hooked for your next project! I did.