Mainline Linux goes real-time ... really?

Here is something that only a few people know: About 90% of real-time capabilities have already been merged into mainline Linux. The remaining 10%, however, are of particular relevance to industrial users. User support, therefore, is urgently needed at this point.

Since the end of the 90s, plans have existed to turn Linux into a real-time operating system. One of the reasons for this was the rapidly accelerating speed of innovation in information technology. At the time, the available dedicated RTOS kernels had to be individually retrofitted each time a new technology swept the market. The effort that this required was enormous, and the delay in availability of the new technologies for industrial systems was increasingly criticized by users. This led to the realization that it would make more sense over the long term to turn a general purpose operating system kernel into a real-time operating system kernel than to repeatedly equip all RTOS kernels with the technologies of general purpose operating systems. However, this was no easy undertaking. Many respected operating system experts at the time felt that it was impossible to render an operating system real-time capable as an afterthought. However, the open structure and flexibility of well-coordinated Open Source development, as in the case of the Linux kernel development, made this feasible even though it was a daunting task. Among those who took up the challenge, were:

- Doug Niehaus, Professor at Kansas University, USA
- Ingo Molnár for Red Hat
- Thomas Gleixner, CEO of Linutronix, for several clients
- Paul McKenney for IBM and
- Steven Rostedt for Red Hat

After the basic components were developed from 2000 to 2006, they were gradually merged into the mainline Linux kernel, a task that is about 90% completed. The remaining 10% are available as the so-called PREEMPT_RT patch. At present, the patch is being maintained and adapted to the current Linux kernel by Thomas Gleixner and his coworker, Sebastian Siewior, at Linutronix. In addition, Steven Rostedt is taking care of the real-time capabilities of the long-term versions 3.2, 3.4, 3.10 and 3.12. The PREEMPT_RT patch supports more architectures and subsystems and is more closely linked with mainline development than any other method for achieving real-time for Linux.

Once the mainline Linux kernel has been equipped with real-time capabilities and is appropriately configured, a real-time operating system will be available that is largely capable of coping with established RTOS kernels in many ways. The two key benefits of a real-time mainline Linux kernel are

- guaranteed real-time features for the vast majority of industrial systems, and
- an API exclusively based on the POSIX standard.

The fact that the response behavior of such a real-time Linux kernel is predictable and can actually be guaranteed has been confirmed by long-term measurements at the test center of the Open Source Automation Development Lab (OSADL) in which 100 million trigger pulses per test system are evaluated daily under different stress and load scenarios. The results were depicted in sequential latency plots. The logarithmic scaling of the frequency values in the y-direction makes it possible to visualize even a single outlier. As can be seen in an example measurement, not a single outlier arose in over 60 billion cycles. The maximum measured latency of the 2700-MHz Intel processor of around 20 µs is about that which is achieved by dedicated RTOS kernels (see Figure 1).

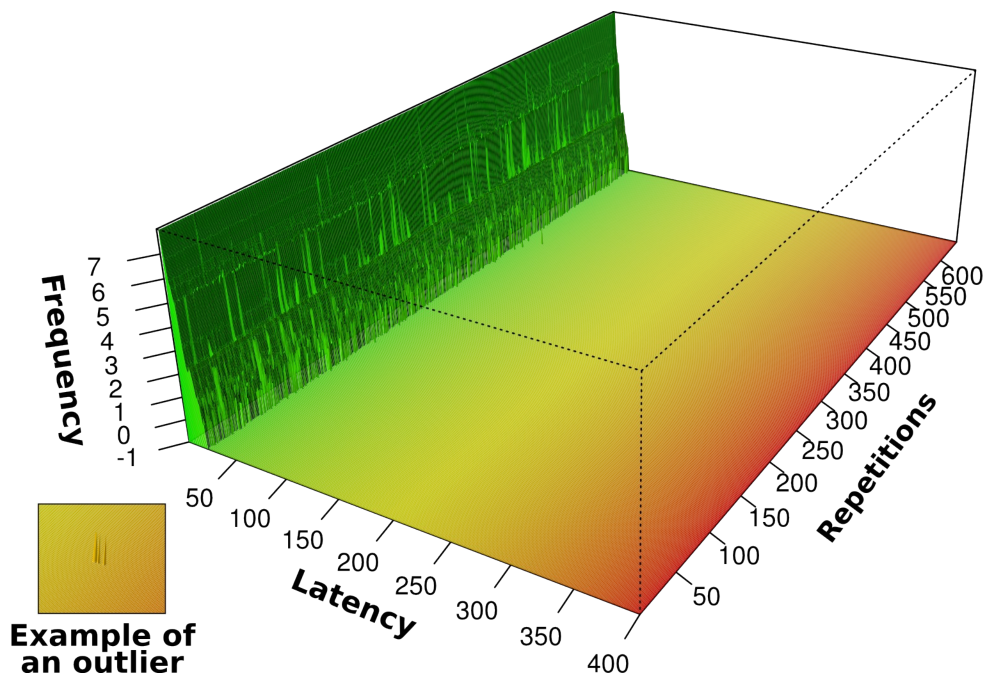

Figure 1: Sequential latency plots of 100 million test cycles. In this example of a 2700-MHz Intel processor, the overall latency of the mainline Linux kernel was measured twice a day over five-and-a-half hours for an entire year. A total of 60 billion test cycles without a single outlier. What an individual outlier would look like is shown at the bottom left. The greatest measured latency of slightly more than 20 µs is about the same as that achieved by dedicated real-time kernels.

Despite the impressive success of the real-time Linux kernel, we should not loose sight of the fact that this software development requires a functioning ecosystem, as is the case with other components of the Linux kernel and Open Source software in general. Such ecosystem works like a charm when the Linux kernel is used in servers, smartphones and especially in telecommunications. Companies who rely on specific Linux components in the relevant branches either provide their own personnel to work in the Open Source communities or become members in the relevant communities of interest such as the Linux Foundation in the USA. In this manner, the beneficiaries of the software participate in its development, and a classic balance of give and take is generated in the community, a process manifested in the implementation of real-time Linux kernel features. In the beginning, companies such as IBM and Red Hat shouldered a major portion of the costs in a concerted effort because real-time Linux was sorely needed for customer projects. Their requirements have now largely been fulfilled. Nonetheless, the non-existing mainline consolidation of the remaining components does not constitute a disadvantage since the systems can be individually addressed by a small specialist team using a separate business model. The corresponding product by Red Hat is called MRG (Messaging/Realtime/Grid); it contains a real-time Linux kernel yet is only available for x86 systems (32 and 64 bit) and is only certified for a relatively small number of hardware platforms.

The use of Linux within industry is a different matter altogether: Real-time features and other branch-specific extensions are required in a variety of architectures and in hundreds of different systems. Reliable and sustainable development is only feasible if the missing components are incorporated in the main branch of development of the Linux kernel. Linus Torvalds has affirmed this repeatedly, with the proviso that additional components must be incorporated by a team of appropriately qualified kernel developers. The Open Source Automation Development Lab (OSADL), founded at the end of 2005 for this purpose, has experienced consistent growth and functions as a nexus for a significant number of industrial companies. The available funds of OSADL have been used for a while in quality assurance of the real-time Linux kernel. If the anticipated memberships materialize by the end of 2014, OSADL will also be able to fund the core of the required team. To finance the entire team, the assistance of a larger community is required. Recently, Thomas Gleixner expressed a degree of skepticism about the real-time Linux kernel and feared a deceleration of development, if long-term financing could not be guaranteed. OSADL is counteracting this apprehension by further expanding its community. From the initial development of real-time capability for the mainline Linux core, we can learn that a group of motivated individuals can develop the required Linux technologies for a specific branch which can be incorporated in the mainline kernel, thus ensuring long-term availability. To translate this model into the current situation, more of the large number of industrial Linux users must help develop a corresponding ecosystem to fully integrate the missing real-time components with the real-time features into the main development branch of the Linux kernel.

Dr. Carsten Emde (OSADL)/jk

This article is an English translation of the article "Mainline-Linux wird echtzeitfähig - oder nicht?" which appeared in the Journal Elektronik on September 9, 2014.