BlackBerry | QNX

5 Reasons to Consider an Alternative to Linux for Your Medical Device

Medical device designers have become increasingly interested in the Linux® operating system (OS), largely because of its open source model. Linux lets the designer leverage a large pool of developers, a rich legacy of source code, and industry-standard POSIX APIs. But Linux may not be the right choice for all projects, especially safety-critical medical devices such as those used in blood diagnostics, ultrasound imaging, infusion delivery, heart monitoring and resuscitation, and robotic surgery. For these demanding applications, the standard Linux kernel can't deliver the safety-critical capabilities that many medical embedded devices require.

While Linux may be acceptable for some medical applications, the QNX® OS for Medical is a compelling option for software development of safety-critical devices. This real-time operating system (RTOS) is POSIX-compliant, pre-certified to IEC 62304, and is supported by field-proven development tools, feature-rich middleware, and engineering services.

Developers of safety-critical medical devices should consider these 5 issues when deciding whether Linux is the best option for their OS.

Safety Certification

01

Regulatory requirements are driving the demand for software that is compliant to the IEC 62304 standard for "Medical device software – Software life cycle processes." IEC 62304 requires that manufacturers follow good development practices to produce high-quality software for medical applications. The standard has been endorsed under medical device-related directives by the FDA in the U.S. and by the Directorate-General for Health and Consumers in the E.U.

While certified systems can be designed based on an open source OS, the development efforts can be considerably more complex, expensive, and time-consuming than when using an OS designed specifically for this purpose. Even small patches or software updates to an open source OS can result in a large ripple effect of rebuilds and changes that can require expensive retest and recertification efforts. The Linux® Foundation estimates that the development community has been merging patches at an average rate of 7.71 patches per hour since the 3.10 kernel release (October 2011). Keeping up with this rate of change in terms of accepting or rejecting the patch, validating the ripple through effects of the change (such as addressing dead code), and testing is a monumental task.

The core of QNX OS for Medical, has been assessed by the independent third-party auditing body TÜV Rheinland® to meet IEC 62304 Class C, the highest class for applications where death or serious injury is possible. This pre-certification of the OS can significantly reduce certification efforts for medical device manufacturers.

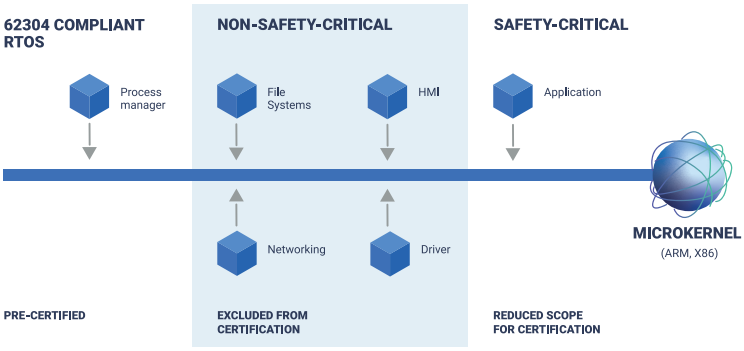


Figure 1:
Using a pre-certified RTOS helps reduce scope of certification

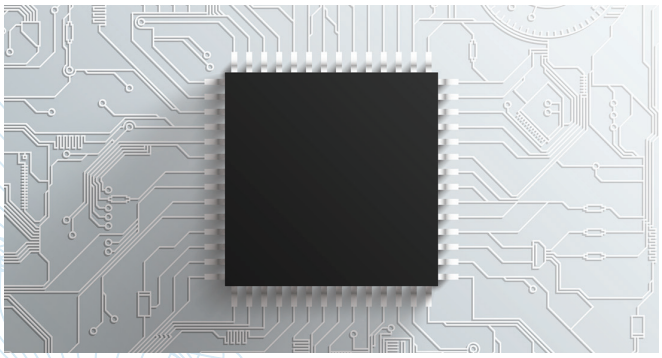
Code Traceability and SOUP

02

The IEC 62304 standard cautions on using software, particularly "software of unknown provenance" (SOUP). It spells out a risk-based decision model on when the use of SOUP is acceptable and defines testing requirements for SOUP to support a rationale on why it should be used. Certification also requires traceability of patch sets and code, as well as ensuring that the development processes are strictly followed.

As open source, Linux encourages broad contributions and changes to the source base, with no requirements for standards-based development processes, making it literally "of unknown provenance." And while the Linux source code is open to scrutiny, the sheer number of developers (more than 12,000) contributing to it make it virtually impossible to trace everything in the code to meet the standard's stringent risk analysis requirements for certification.

If certification will be required for the final device, the ability to trace the source code is an important consideration. BlackBerry® QNX® maintains unique, patented systems that trace every build back to source origins and specific commits so that the lineage of all source code is known and tracked.



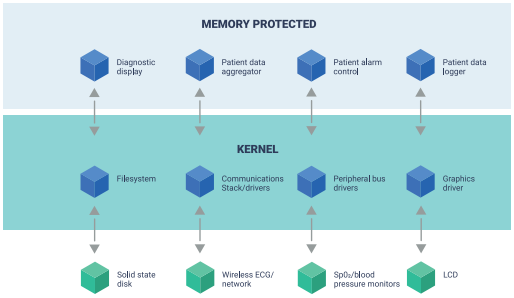
Drivers and OS Services

03

Leading-edge hardware is strongly tied to intellectual property (IP) rights, and manufacturers are hesitant to compromise their IP by releasing source code (which Linux requires). If the provided Linux driver isn't suitable for the device, developers must write and maintain their own drivers, which requires valuable time, developer resources and increases development costs. When using QNX products and tools, there are no legal obligations to disclose source, so hardware manufacturers have far less IP exposure.

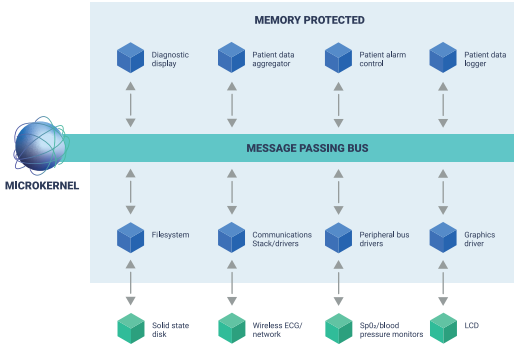
Figure 2 below shows a hypothetical medical monitoring device built on a monolithic OS. The drivers, file systems, communications stack, etc. must all be included in the risk analysis because they impact and can corrupt the kernel.

Figure 2:
Example of a monolithic OS architecture



The QNX OS for Medical also delivers a full range of OS services required for medical applications. These include networking services that support complex distributed systems in which multiple devices seamlessly share resources and communicate without custom protocols. The OS also supports a wide range of block and flash file system formats along with a power-safe disk file system for data integrity and reliable storage. For applications with a user interface, the screen framework enables developers to build graphically rich, compelling user interfaces using built-in, high performance, OpenGL ES-based transitions. Additional 3rd party user interface technology, such as Qt or Crank, seamlessly integrates with the screen framework. Figure 3 shows the microkernel architecture of the QNX RTOS which provides extensive fault containment and recovery as every driver, protocol stack, file system, and application runs outside the kernel, in the safety of memory-protected user space.

Figure 3:
Example of a microkernel OS architecture



Cost and Time-to Market

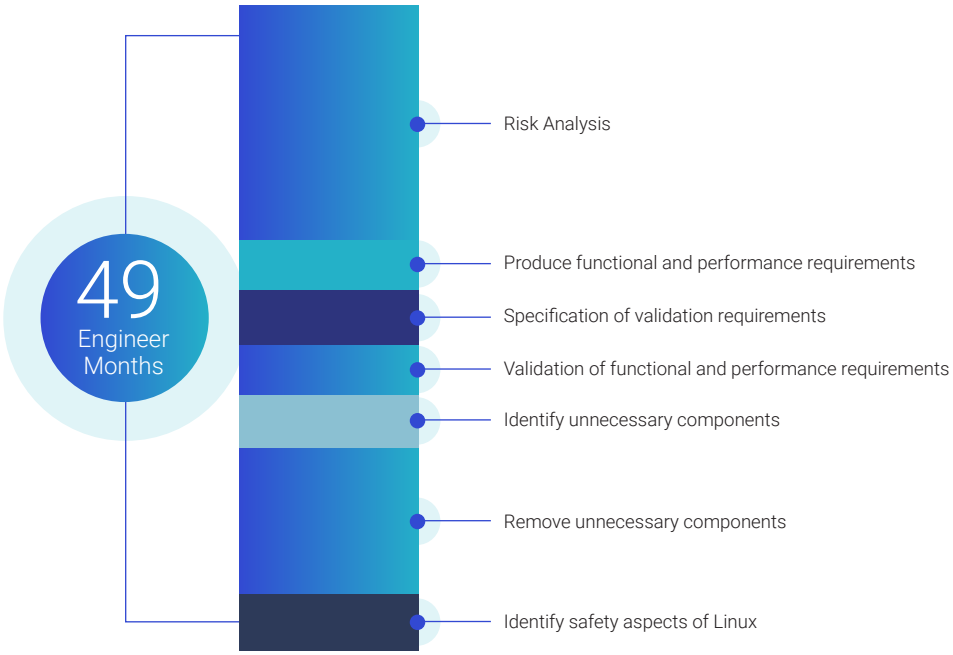
04

Linux is often considered a "free" operating system because it provides open access to its source code. The total cost of ownership must be considered, however. Using an open source OS may incur extra costs for the development time and testing to certify the system, and this additional development time can result in lost revenue due to delays in bringing the device to market. Most medical device developers want to focus their efforts on their proprietary, value-added application, but an open source OS may require an additional investment to sustain an in-house team of OS experts to configure, build, support, certify, test, and maintain a large code-base that is the foundation of the product.

QNX OS for Medical is a full-featured RTOS that has been proven in millions of the most demanding and mission-critical medical devices. This POSIX-compliant OS greatly simplifies the migration from a Linux-based prototype to a production system. The LinuxAPI compatibility resulting from POSIX compliance can increase code re-use and eliminate the learning curve that often accompanies the adoption of a commercial RTOS. It helps reduce program cost and risk and shortens the time-to-market for medical device developers.

Estimation of Engineering Effort

Figure 4:
Details the activities required for the initial deployment of a Linux-based system in a device requiring safety certification or pre-market safety approval.



Support and Licensing

05

The community development model of open source software can make it difficult for developers to get the help they need, when they need it. With demanding time-to-market expectations for medical devices, waiting for an OS issue to be solved can have a huge impact on the company's success. And if the solution is an upgrade or patch, that introduces additional complexities and opportunities for more support issues. In addition, it's up to the application developer to ensure that the correct licenses are used and have been attributed properly to avoid contamination of a proprietary source base.

QNX® products include a licensing support system that helps ensure the provenance of all of the source code. And QNX offers a range of additional support services including customized courses tailored to distinct project needs, technical requirements, and project-specific challenges. To complement QNX OS for Medical, BlackBerry QNX also provides training and architectural reviews as well as custom engineering services. Our expertise in designing safety-certified embedded systems means we can guide developers through the safety certification process to help meet important product launch dates.

Don't Rush an OS Decision Without All the Facts

Not every medical device application requires a safety-certified operating system. For many of those applications, Linux can be an appropriate option. But for life- or safety-critical embedded systems that must operate 24 hours a day, 365 days a year, without failure, the QNX OS for Medical is a proven, reliable choice.

About BlackBerry® QNX®

BlackBerry QNX is a leading supplier of safe, secure, and trusted operating systems, middleware, development tools, and engineering services for mission-critical embedded systems. BlackBerry QNX helps customers develop and deliver complex and connected next generation systems on time. Their technology is trusted in over 150 million vehicles and more than a hundred million embedded systems in medical, industrial automation, energy, and defense and aerospace markets. Founded in 1980, BlackBerry QNX is headquartered in Ottawa, Canada and was acquired by BlackBerry in 2010.

© 2020 BlackBerry Limited. All rights reserved. QNX, Momentics, Neutrino, are trademarks of BlackBerry Limited, which are registered and/or used in certain jurisdictions, and used under license by BlackBerry QNX. All other trademarks belong to their respective owners.