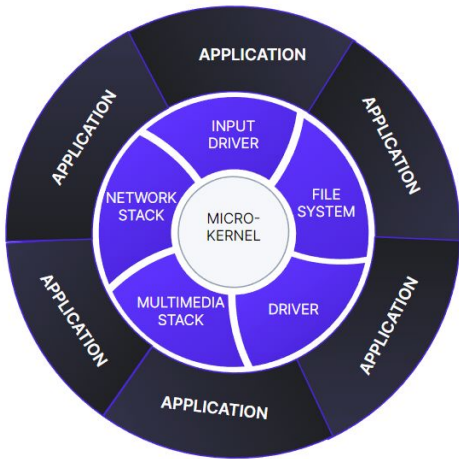


Best practices for performance optimization in Real Time Operating Systems using Codee

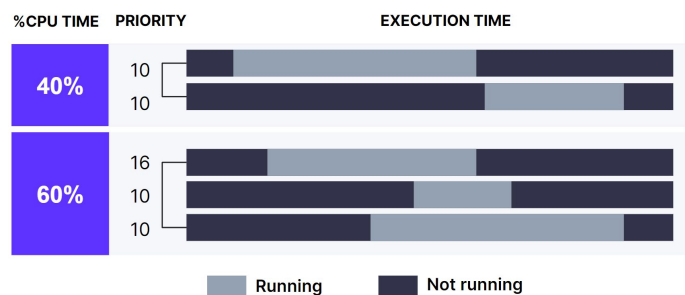
Real-Time Software operates within an immediate time frame; sensing, analyzing, and acting on streaming data as it happens. Common examples of real-time systems include air traffic control systems, process control systems, and autonomous driving systems.

Real-time software applications run on a Real-Time Operating System (**RTOS**) that processes data and events that have critically defined time constraints. The RTOS is responsible for running real-time applications, task scheduling, memory management and interaction with the file system.



How does a RTOS work?

Running real-time applications in the scope of hardware resources constraints makes performance and efficiency a high priority. The RTOS schedules the execution of real time applications on the system guaranteeing the priorities set.



What is not addressed by the RTOS is the optimization of the performance of the application.

This is the responsibility of the software development team, which optimizes the environment setup and tunes the source code for the target processor, compiler and operating system.

How do developers improve the performance of applications?

Typically the code is written and tested relying on the environment to optimize performance. Only very late in the software development cycle the application source code is optimized if performance issues are detected.

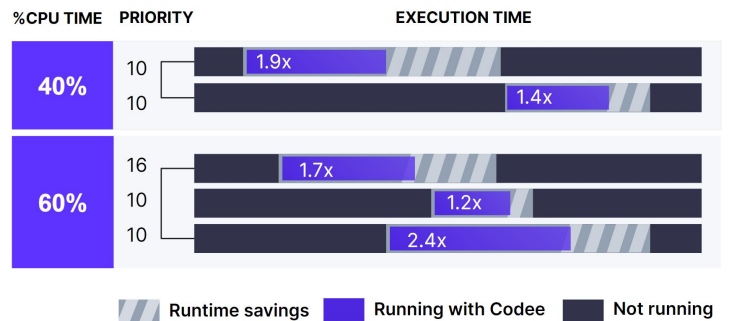
It usually requires hiring expert software performance engineers, which are expensive and scarce. At this point it is very expensive in cost and time to address performance issues.

In modern agile software development the benefits of Shift Left are very well-known.

There are static code analyzers that address security, bugs, compliance and quality. However existing tools do not look at the source code from the performance point of view. Today it remains a costly manual process yet.

Codee Static Code Analyzer is the first tool of its kind specialized in performance.

It helps enforce performance optimization best practices in the source code of the application. It brings the benefits of Shift Left to performance optimization.



Example: Real-time applications running on RTOS QNX

There is a great variety of free and commercial RTOS's in the market. For illustrative purposes, consider the Linux-based RTOS QNX 7.1 equipped with the GCC 8.4 compiler targeting x86/Arm processors. **Significant performance gains are enabled by enforcing the performance optimization best practices detected by Codee Static Code Analyzer:**

MbedTLS cryptographic algorithms library: Codee completed the analysis in **1 minute 43 seconds**, finding **292 checks** over **370 files** with **325822 lines** of code.

AES_XTS encryption code runs **1.3x** faster enforcing 5 checks related to single-core optimizations for vectorization and memory efficiency on x86 and Arm processors.

Take advantage of Codee Static Code Analyzer today and enforce performance optimization best practices in your real-time software.

Go beyond security, bugs and compliance and benefit from Shift Left in performance as well.

For more information and to arrange a demo/discovery, contact us at codee.com/request-demo/.