EECS 3221 Report

A Comparison of Real Time Operating Systems and the Linux Operating System

Daniel Di Giovanni — 218204818

February 9, 2024

My signature below attests that this submission is my original work:

Following professional engineering practice, I bear the burden of proof for original work. I have read the York University Senate Policy on Academic Integrity and the EECS Academic Honesty Guidelines and confirm that this work is in accordance with the Policy.

February 9, 2024

Signature **Date**

Contents

Introduction and Background

An operating system is a computing layer that separates the hardware of the computer from the programs that run on it. It provides the *environment* for other programs to do useful work [1, p. 4]. The fundamental tasks of an operating system include allocating resources (such as memory and CPU time), handling the control of input/output (I/O) devices, and ensuring proper usage of the computer and preventing errors [1, pp. 3-5].

The most important part of an operating system is the *kernel*. It is the first program loaded into memory on startup and is the one program that is always running on the computer [1, pp. 6-7, 22]. Along with the kernel, operating systems also include *middleware frameworks* that ease application development, and *system programs* that help the system run but are not part of the ever-running kernel. All of this supports the execution of *application programs*, which are the programs that provide functionality to the end user [1, p. 4, 7].

In industrial and commercial computing applications, the choice of an operating system is crucial. It affects the performance, security, and maintainability of the system. As an example, consider the secure boot of an embedded system. Secure boot, an important security technique to ensure that the kernel code has not been modified, is often neglected in embedded systems. The absence of secure boot allows the system to boot faster with less memory and energy consumption—at the cost of leaving the boot process and internal software vulnerable. However, it was discovered that the introduction of secure boot software caused boot-up time to increase by only 4%, whereas a hardware implementation of secure boot caused a 36% increase [2, pp. 11-12]. Clearly, the operating system has a significant impact on the overall quality of the system.

Two important classes of operating systems/kernels will be discussed here: the Linux operating system and real-time operating systems (RTOSs). The Linux kernel is a free and open source implementation of an operating system kernel. It is used ubiquitously not only for desktop computers, but also for servers and embedded devices with a broad range of commercial and industrial applications [3]. The Linux kernel is a tried-and-tested system with high flexibility and extendability. RTOSs are more vague, being defined not by a specific implementation, but by the ability to manage systems with complex time and resource constraints [4]. RTOSs need to be able to meet strict deadlines associated with external events using limited resources. In short, "a real-time system is one whose correctness involves both the logical correctness of the outputs and their timeliness" [5].

The objective of this report is to provide a thorough comparison of the Linux operating system/kernel with RTOS/real-time kernels to aid in the decision of which operating system to use.

Overview of Linux

Overview of Real-Time Operating Systems

Conclusion

References

[1] A. Silberschatz, P. B. Galvin, G. Gagne, *Operating System Concepts*, 10th ed., John Wiley and Sons, Inc., 2018.

[2] J. Ingelhag, “How to choose an operating system for an embedded system”, Örebro Universitet, 2023. Accessed February 9, 2024. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1773441/FULLTEXT01.pdf.

[3] The Linux Foundation, “What is Linux?,” *The Linux Foundation*. Accessed February 7, 2024. [Online]. Available: https://www.linux.com/what-is-linux/.

[4] W. Cedeño, P.A. Laplante. “An Overview of Real-Time Operating Systems,” JALA: Journal of the Association for Laboratory Automation. 2007, ch. 12, sec. 1, pp. 40-45. Accessed February 7, 2024. [Online]. Available: https://doi.org/10.1016/j.jala.2006.10.016.

[5] P. A. Laplante, “Real-Time Systems Design and Analysis,” 3rd ed., Hoboken, NJ, Wiley, 2004, p. 505. Accessed February 7, 2024. [Online]. Available: https://doi.org/10.1002/0471648299.fmatter.

Appendix A Code

A.1 Q1.m