Critique 1 of 10

Written by: Victor Palffy

Submitted on: 05/18/2020

Publication venue: HotOS

Title: “Composing Abstractions using the null-Kernel“

In this article the author opens by describing a common paradigm found in programming and designing an operating system (OS) interface. The paradigm of two extremes. Example 1, the monolithic kernel design, which has a high-level portable interface and hides hardware specific information from the user. Example 2, the exokernel, which opens access to low-level hardware firmware and enhances performance. After briefly describing the advantages and disadvantages of the two designs the author then suggests the null-kernel approach to interface design.

The Null-kernel, which has both a high- and low-level interface and opens low-level hardware utilization, but also scalability for high-level hardware/firmware implementation and optimization. In other words, there is more support for low-level interfaces development. The null-kernel works by utilizing abstract machines (AMs) to handle the interface; AMs are located on low and high levels. The null-kernel architecture is comprised of the null-kernel, caller and abstract machines. An abstract machine could be a hardware AM or a virtual machine AM (VM-AM). Higher-level AMs receive their capabilities and permissions from lower-level ones. The null-kernel can also be integrated within existing OS such as FreeBSD to enhance its high- and low-level interface.

In an extended freeBDS system EXO AM can be found which is a substitute for the exokernel system. In this hybrid version of freeBDS callers are used to access the EXO AM. With the Null-kernel layer bypassing is a possibility. New hardware can easily be integrated with the EXO AM and later be built upon. This gives free BDS better garbage collection and overall better performance.

What I liked most about this article was the author’s ability to draw a picture of the integration process of the null-kernel. I especially liked the brief examples given on other kernel types and the authors summary for how the system could be improved via null-kernel. This piece definitely got me looking back to my current and previous textbooks for references.