Ritwik Takkar
10/02/22

Paper Report: On the Duality of Operating System Structures by Hugh C. Lauer and Roger M. Needham

ritwiktakkar.com

1 Summary

This is an empirical paper that demonstrates two categories of operating system designs which are a dual (equivalent models) of one another: *Message-oriented Systems* (read: event-based systems) and *Procedure-oriented Systems* (read: thread-based systems). The former indicates a relatively small and static number of processes with an underlying explicit message system, while the latter connotes a large and dynamic number of smaller processes synchronized based on shared data. The authors conclude that it is actually the machine architecture that must be the primary constraint when selecting the model, rather than higher-level applications.

2 Strengths of the paper

The three claims of equivalence made by the authors seem sufficient to me in making the argument that the dual programs are indeed similar to each other. Specifically, when using primitives of strictly one model, either model can be used to construct a program; both programs are logically identical to one another; and, given the same scheduling strategies, both models should offer identical performance. It seems designers of the past and still today passionately argue over one model's superiority over the other, whereas such arguments may be considered rather trivial in the grand scheme of things. The authors acknowledge the practical limitation of portraying an example to quote in support of their thesis. Namely, the "underlying addressing structures, use-of global data, and styles of communication (which) are usually so bound to the design and implementation that performing the transformation to a dual version would be a major exercise, not justified by the second order gains." Still, they discuss the Cambridge CAP Computer, which coincidentally lends itself to be an example due to its modular OS design.

3 Weakness of the paper

As is the case with the nature of empirical papers, those attracted to and only convinced by rigorous mathematical proofs might find themselves hard-pressed to be convinced by the authors' observations/arguments. Specifically, they may remain rather fervent in their designation of which category to implement and when to do so, thus disagreeing with this notion of duality between the two systems.

4 Future work opportunities

Despite the authors finding attempts to combine "fundamental characteristics" from the two categories to be unsuccessful, I wonder whether today's machine architectures have matured to the point that they can afford to handle such a mixture? For example, the implementation of a *Procedure-oriented system* within the block and a *Message-oriented system* for the blockchain so that each block can contain easily modifiable transactions within but each block itself can't be easily modified...