Secure Software Development (Computer Science) November 2021

Debate: Microservices and Microkernels

Read Appendix A: the Tanenbaum-Torvalds debate in DiBona & Ockman (1999) then read Fritzsch et al (2019).

The forum has a message that says: “Torvalds has been proven wrong and it only took nearly thirty years. Microservices and microkernels are the future. “

- On the forum post a message either agreeing or disagreeing with the above and give a justification (ideally with an academic reference) supporting your view.
- Outside the forum, discuss your positions in your team and come up with a team stance. This should be shared in Unit 12.

Learning Outcomes

- Critically analyse development problems and determine appropriate methodologies, tools and techniques (including program design and development) to solve them.
- Systematically develop and implement the skills required to be effective member of a development team in a virtual professional environment, adopting real-life perspectives on team roles and organisation.

1 discussion

Add a new discussion

Search

Michael Botha

Monolithic Kernel vs. Microkernel

1 day ago

Manage forum subscriptions

Secure Software Development (Computer Science) November 2021

« Debate: Microservices and Microkernels

Michael Botha

Monolithic Kernel vs. Microkernel

1 day ago

The statement that “Torvalds has been proven wrong” is misleading and muddies the rest of the comment where microservices and microkernels are said to be the future, which in and of itself is also not a complete statement. Therefore, I disagree with the entire message.

Firstly, the context in which the Tanebbaum-Torvolds debate takes place reveals the premise that Torvalds doesn’t completely disagree with the idea of a microkernel-based system, but that it was not practical and implementable at the time (DiBona & Ockman, 1999). Furthermore, he explains that his target audience required a system similar to the one he designed for various reasons (DiBona & Ockman, 1999). Therefore, in essence the debate was not whether the one paradigm was better than the other, but rather that the one was a better option than the other within the set of requirements. Additionally, even if Torvalds hypothetically did believe that monolithic systems were ultimately the best, and Tanebbaum that microkernels were superior it would merely show a misinformed understanding of the general principles of engineering, where it is commonly understood that every paradigm has advantages and disadvantages.

Secondly, although microservices and microkernels are mentioned in certain domains such as security as something that should be aspired to in the future (Biggs et al., 2018), there are a multitude of quality attributes which govern the choice of software architecture (Pillai, 2017). For instance, modifiability, testability, scalability, performance, availability, security, deployment (Pillai, 2017). Currently some of the major Operating System (OS) developers are still supplying monolithic systems (T4, 2021; Red Hat, 2019). This due to the fact that both paradigms present various pros and cons.

In conclusion, the below table presents a comparison of the two architectures.

Kernel Comparison Table (Pedamkar, N.D)

Basis of Comparison	Monolithic Kernel	MicroKernel
Execution Style	All processes are executed under the kernel space in privileged mode.	Only the most important processes take place in the Kernel space. All other processes are executed in the user space.
Size	Kernel size is bigger when compared to Microkernel.	Kernel size is smaller with respect to the monolithic kernel.
Speed	It provides faster execution of processes.	Process execution is slower.
Stability	A single process crash will cause the entire system to crash.	A single process crash will have no impact on other processes.
Inter-Process Communication	Use signals and sockets to achieve interprocess communication.	Use messaging queues to achieve inter-process communication.
Extensibility	Difficult to extend.	Easily extensible.
Maintainability	Maintenance is more time and resource consuming.	Easily maintainable
Debug	Harder to debug	Easier to debug
Security	Less Secure.	More Secure
Example	Linux	Mac OS

References:

Biggs, S., Lee, D., & Heiser, G. (2018) The Jury Is In: Monolithic OS Design Is Flawed: Microkernel-based Designs Improve Security. *Proceedings of the 9th Asia-Pacific Workshop on Systems (APSys '18)*. ACM 16:1–7.

DiBona, C., Ockman, S. (1999*) Open Sources: Voices from the Open Source Revolution*. 1st ed. Sebastapol: O'Reilly Media Inc.

Pillai, B. (2017) *Software Architecture with Python*. 1st Edition. Birmingham, UK: Packt Publishing.

Pedamkar, P. (N.D) Monolithic Kernel vs Microkernel. Available from: **https://www.educba.com/monolithic-kernel-vs-microkernel/** [Accessed 14 February 2022].

Red Hat. (2019) What is the Linux Kernel. Available from: **https://www.redhat.com/en/topics/linux/what-is-the-linux-** [Accessed 15 February 2022].

T4. (2021) Server Operating System Market Share. Available from: **https://www.t4.ai/industry/server-operating-system-market-share** [Accessed 15 February 2022].