Debate: Microservices and Microkernels

“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.*

My discussion:

In 1992 Linus Torvalds and Andrew Tanenbaum debated the 'best' architecture for operating system kernels. A kernel is the lowest layer of an operating system that interfaces with the hardware and manages system resources. Linus Torvalds was the primary developer of Linux and Andrew Tanenbaum was the primary developer of an operating system called Minix. Andrew argued that Linux' monolithic design was flawed and that microkernels designs are superior. Tanenbaum argued that Linuxs' monolithic design made the system hard to port. A monolithic kernel implements all the necessary features of the kernel in one, whereas a microkernel is broken into multiple separate services that implement the whole kernel. Monolithic kernels operate faster than microkernels, due to additional communication through each services interface. Microkernels can be more secure, errors in individual services do not crash the system.

I will now consider the statement "Torvalds has been proven wrong and it only took nearly thirty years. Microservices and microkernels are the future". Firstly, we can compare the success of the operating systems of each developer. Linux went on to have far greater prominence than Minix. The most common modern OS's are a hybrid between Monolithic and Microkernels. Windows and Linux are closer to Monolithic kernels. From this perspective, I would disagree with the statement. However, Microservices architectures have become dominant for modern web applications. For web applications, microservices offer greater reliability and scalability. The reason this architecture may be more applicable for web applications maybe because of the different performance requirements. Operating systems and Web applications are software that runs at different layers. The application layer has lower performance needs. A web application ironically may be best suited to a Microservice design, but the code will likely run on operating systems with monolithic architectures. As such, I partially agree with this statement. As such, I partially agree with this statement.

[308 words]