

Microkernels: From Mach to seL4 (Lecture 16, cs262a)

Ali Godsi & Ion Stoica,
UC Berkeley
October 21, 2020

Papers

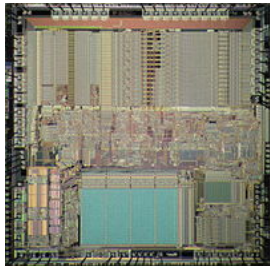
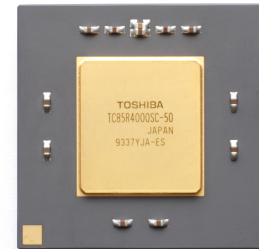
“Microkernel Operating System Architecture and Mach”, D. Black, D. Golub, D. Julin, R. Rashid, R. Draves, R. Dean, A. Forin, J. Barrera, H. Tokuda, G. Malan, and D. Bohman

(<https://amplab.github.io/cs262a-fall2016/notes/Mach.pdf>)

“seL4: Formal Verification of an OS Kernel”, Gerwin Klein , Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, Simon Winwood,

(<https://www.sigops.org/sosp/sosp09/papers/klein-sosp09.pdf>)

~1985: Lots of processors



Want to port the OS on a variety of processors

~1985: Opportunity for new OSes

Microprocessors fast enough to run multi-user systems, i.e., Unix

Microsoft's transitioning to Windows from DOS



- Windows 1 released in 1985



IBM (and Microsoft) working in OS/2

Excitement about adding new functionality to OSes

Tanenbaum vs Torvalds debate

Monolithic vs microkernel OSes

<https://www.oreilly.com/openbook/opensources/book/appa.html>

See how the changes in context let to changes in the tradeoffs!

Key Observation (~1985)

Modern OSes at that time (e.g., Unix, OS/2) primarily distinguished by the programming environment they provide and not by the way they manage resources

Opportunity:

- Factor out the common part
- Make it easier to build new OSes

Microkernels separates OS in two parts

Part of OS that control basic hardware resources (i.e.. microkernel)

Part of OS that determine unique characteristics of application environment (e.g., file system)

What problem do they try to solve?

Portability:

- Environment mostly independent on the instruction set architecture

Extensibility & customization:

- Can easily add new versions of environments
- Enable environments to evolve faster (decouples them from microkernel)
- Can simultaneously provide environments emulating interfaces

Sounds familiar?

- Microkernel as a narrow waist (anchor point) of OSes
- Provide **hardware independence**, similar to data independence in relational data models

What problem do they try to solve?

Easier to provide better functionality and performance for kernel:

- Real-time: no need to maintain lock for extended periods of time; environments are preemptable
- Multiprocessor support: simpler functionality → easier to parallelize
- Multicomputer support: simpler functionality → easier to distribute
- Security: simpler functionality → easier to secure

Flexibility (network accessibility):

- System environment can run remotely