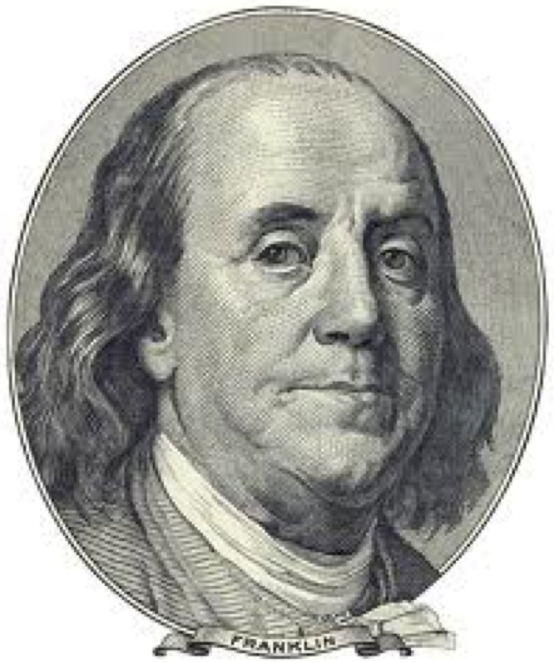
Exokernel and Technical Writing

Baris Kasikci

Hint

It's more "fun" if you work in a team

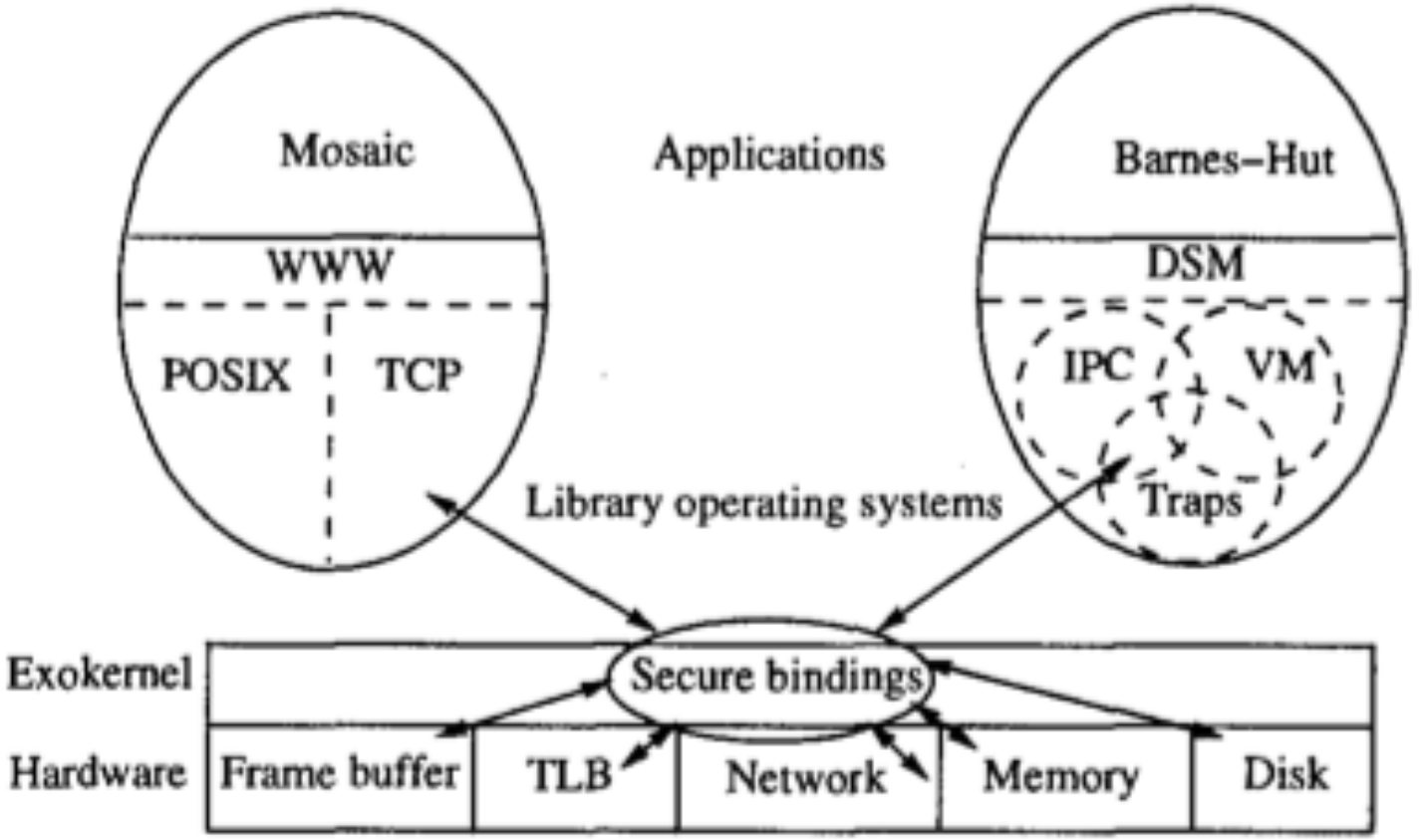
We must indeed, all hang together or, most assuredly, we shall hang separately



The Exokernel – SOSP 1995 (Copper Mtn, CO)

- After the monolithic vs. microkernel debate
  - Mach 1986; Amoeba (Kasshoek); Tannenbaum vs. Torvalds, …
  - Key question was “Where to put functionality” / “modularize the kernel”
  - Exokernel asked a different one
- One of the most influential academic publications
  - Controversial then and now
- Great example of academic research
  - Take an idea to the extreme
  - Pragmatism will emerge over the years

What's in a name ?



Abstract (1)

- “Traditional operating systems limit the performance, flexibility, and functionality of applications by fixing the interface and implementation of operating systems abstractions such as IPC and virtual memory.”

Abstract (2)

- "The *Exokernel* OS architecture addresses this problem by providing application-level management of physical resources."
- "In the Exokernel architecture, a small kernel securely exports all hardware resources through a low-level interface to implement system objects and policies."
- "This separation of resource protection from management allows application-specific customization of traditional OS abstractions by extending, specializing, or even replacing libraries."

S1. Separation of policy and mechanism

- "To provide an interface … an exokernel designer has a single overriding goal: to separate **protection** from **management**"
- Q – how is this different from the principle of separating **policy** from **mechanism** ?
- "One approach is to give each application its own virtual machine [17]. […] virtual machines can have severe performance penalties.
- Q – how do virtualization and exokernel differ ?

Central mechanism of Exokernel

- Securely expose hardware resources (rather than emulate them)
  - Expose allocation
  - Expose names
  - Expose revocation

- Three techniques:
  - Secure bindings
  - Visible resource revocation
  - Abort protocol

- Q – how does this compare with mono-k, micro-k, and VMM ?

Secure Bindings

- (S3.2) Three mechanisms to implement secure bindings
  - Hardware mechanisms (e.g., privileged instructions)
  - Software caching
  - Downloading code
- Q – which are necessary, and which are optimization ?
- Downloading code … non-trivial
  - Does it break abstraction and modularity ?
  - How do you “tame” code ?
  - Is that code Turing complete ? (subject to the Halting problem) ?

Resource multiplexing

- (S3.2) Resource multiplexing via secure bindings of
  - CPU
  - Physical memory
  - Network

- Q – what are the limits of the Exokernel flexibility? Are there hardware resources that cannot be securely multiplexed ? Are there abstractions that must be implemented, after all ?

Questions

- What is the lowest possible implementation of an Exokernel ?
- Would differences in hardware change the principles behind an Exokernel? What if you had to build one today ?

Exokernels ca. 2018 ?

- Today, VMs are everywhere and Exokernels non-existent. Why ?

- The Exokernel role is to share hardware resources between different processes in the system. Nowadays servers are often dedicated to a single task. Would the exokernel design need to be changed to efficiently work in such environments ?

Technical writing

Literature vs. technical writing

- "Thanks to art, instead of seeing one world only, our own, we see that world multiply itself and we have at our disposal as many worlds as there are original artists, worlds more different one from the other than those which revolve in infinite space, worlds which, centuries after the extinction of the fire from which their light first emanated, whether it is called Rembrandt or Vermeer, send us still each one its special radiance."

  — Marcel Proust

- "Each artist distinctly projects the user's perceptions of the physical world. With great artists, the mechanism even lasts beyond the spatial and temporal boundaries of the artist's own life [22,45].

  -- Baris Kasikci

Technical writing explained by great thinkers

- *Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.*

- **I would have written a shorter letter, but I did not have the time.**

— Blaise Pascal

Recursion in papers

- Good papers have a recursive structure:
- Paper has an abstract
- Section has an introductory paragraph = (abstract of section)
- Paragraph has a **topic sentence** = abstract of paragraph
  - First sentence of the paragraph !

Do's and Don'ts

- Technical writing
  - Zero points for "creative writing" - passive form, gratuitous use of synonyms, hyperbolae, subjectivity, emotion
- Always be specific
  - "a cache can significantly affect performance" →
    "a buffer cache of 80GB improves the performance of two workloads by at least 1.25x"
- Follow each general statement with an example.