



User-defined abstractions using exokernels

Daniel Frédéric, 2022

Summary



- What is an exokernel ?
- Motivations
- Implementation details
- Demo
- Limitations
- Current Objectives

What is an exokernel ?



- A way to solve abstraction specification issues
- Two main exokernels, Cheetah (Berkeley) and Aegis (MIT)
- The abstraction is left to the userland
- The kernel safely schedules hardware accesses and ownership
- Not a general purpose kernel as-is, softwares should be trusted

Motivations



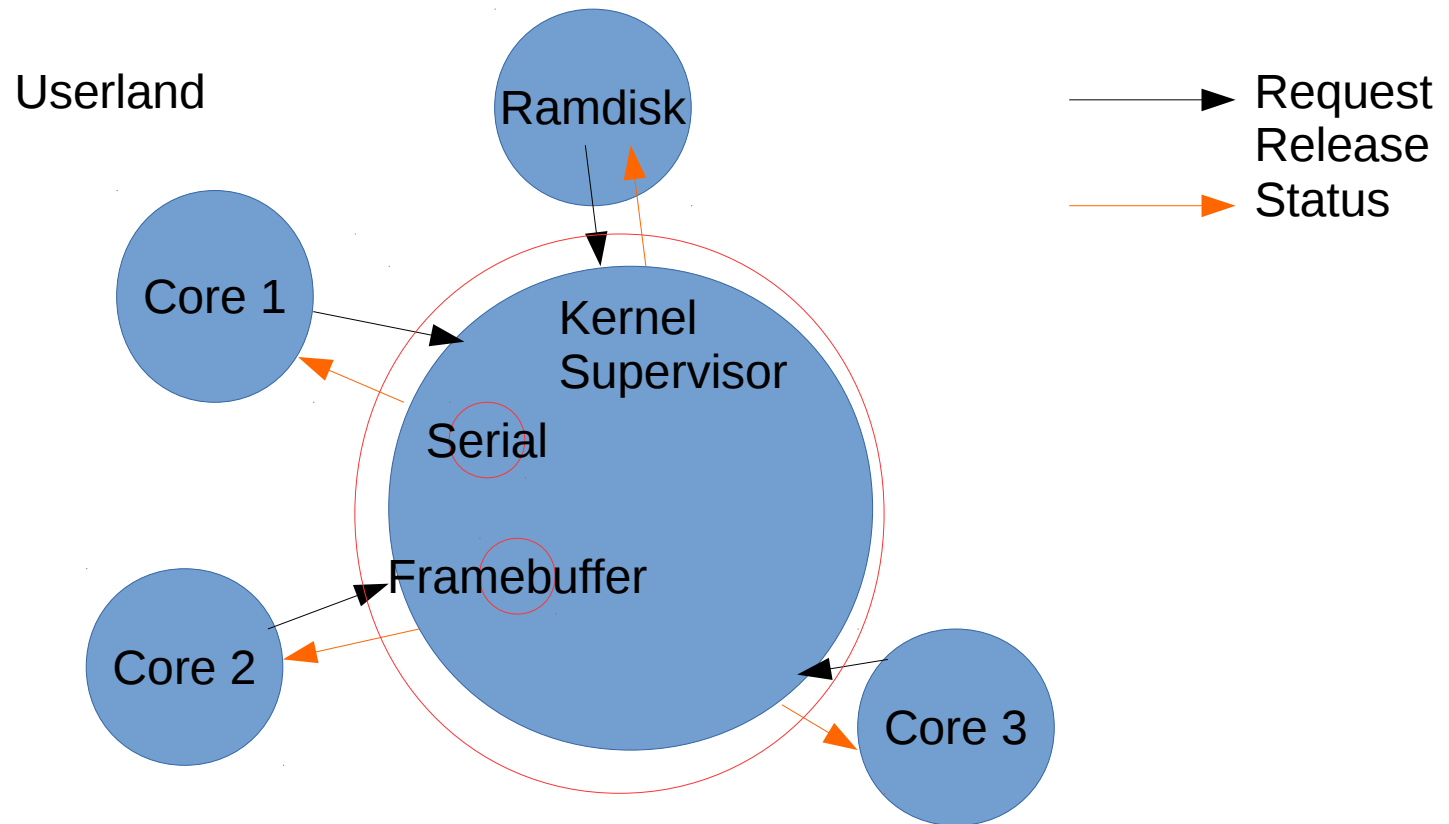
- A huge performance gain :
 - Cheetah had a 16 factor of throughput for an http server
 - Getpid (2) ran ten times more quickly in Aegis
- User-defined abstractions are interesting, because of low-level optimizations available while delivering high-level primitives in libOS
- Laboratories didn't released their implementations on github, research papers are fuzzy, not reproducible, so it's interesting to have a serious implementation with more than 15 github stars.

Implementation details

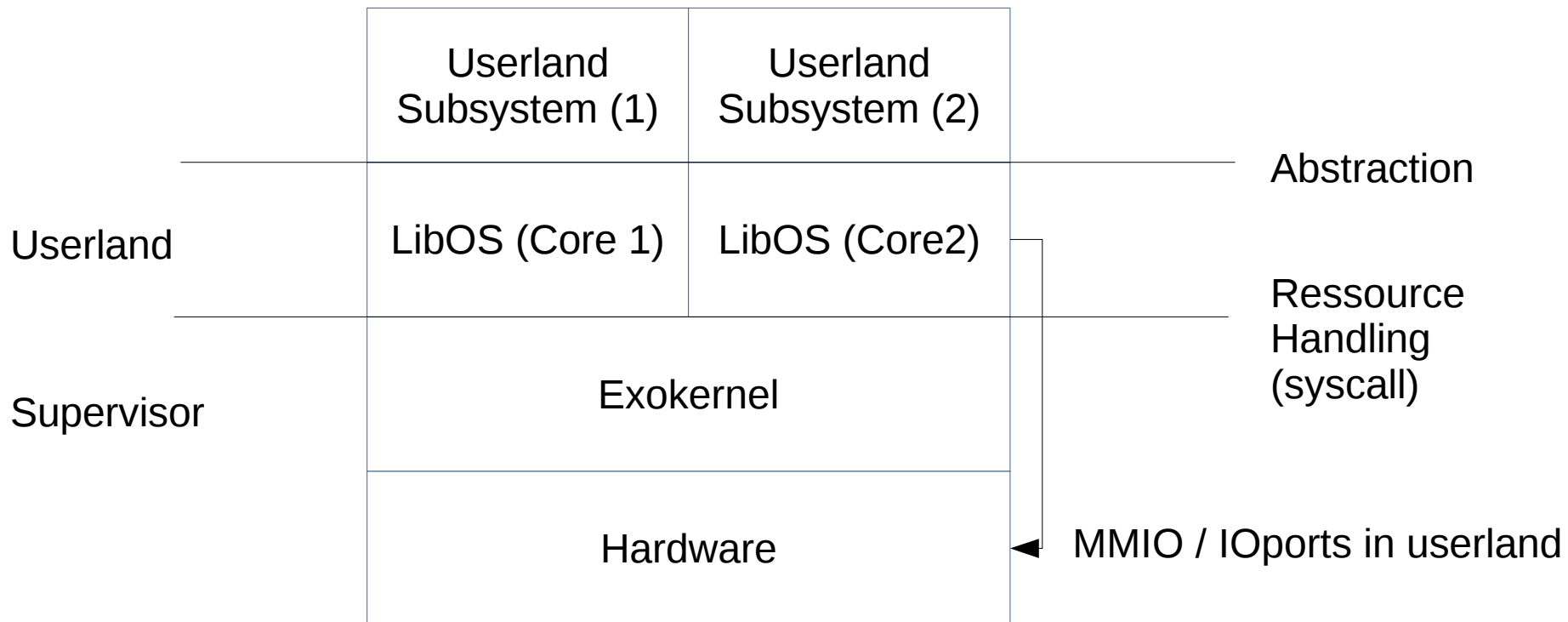


- Current project name : Raccoon-exok (because raccoons are the best!)
- Capability-based. A capability in a user-defined hardware feature
- Two syscalls atm, three maybe in the future :
 - `get_capability`, an abstraction over a hardware feature
 - `release_capability`, release the abstraction
- There is atomicity to manage concurrent accesses !

System architecture



System Architecture



Build demo & source
code



Current limitations



- Need LibOS support !!
- Security is only cooperative, all security layers are on a higher level of abstraction

Current objectives



- Port some softwares and create appropriate libOSes
- Benchmark against newer systems
- Trying to get other kernels running with minimal modifications
- Other architectures support ?

Repository (for now)



<https://github.com/DaemonOnUnix/Raccoon-exok>

Questions ?

