Helios

A small, practical microkernel

Drew DeVault
SourceHut

January 25, 2023

Why write a new kernel?

- Kernel hacking is really fun
- Prove if Hare is useful for this purpose
- Can we do better than seL4?
- Can we do better than, dare I suggest, Linux?

What is Helios?

Helios is a microkernel, largely inspired by seL4. It is written in Hare and runs on x86_64 and aarch64; RISC-V is planned.

- $\approx$ 8,500 lines of portable code
- $\approx$ 3,000 lines non-portable per architecture
- GPL 3.0

Note: Line counts do not include the bootloaders

A brief introduction to Hare

Hare is a systems programming language designed to be simple, stable, and robust. Hare uses a static type system, manual memory management, and a minimal runtime. It is well-suited to writing operating systems, system tools, compilers, networking software, and other low-level, high performance tasks.

A brief introduction to Hare

- General purpose systems programming language
- 3 years in development
- 18,000 line compiler (C11)
- 12,000 line backend (C99)
- x86_64, aarch64, riscv64

What does Hare look like?

```
export @noreturn fn kmain(ctx: arch::bootctx) void = {
        log::printfln("Booting Helios kernel");

        const pages = init::pages(&ctx);
        let heap = init::heap_init(&ctx, pages);
        let task = init::task_init(&heap, ctx.argv);
        init::load(&task, &ctx.mods[0]);
        init::heap_finalize(&task, &heap, &ctx);
        init::devmem_init(&task);
        init::finalize(&task);

        log::printfln("Entering userspace");
        sched::init();
        sched::enteruser(task.task);
};
```

Design basics

Helios is a microkernel with capability-based security.

- Simple, small, and flexible kernel design: 14 syscalls
- More secure than monolithic designs like Linux
- IPC via endpoints/notifications, or shared memory

Helios project status

What works?

- Capability-based security
- IPC (similar to seL4)
- Preemptive scheduling (single core, no SMP)
- Hardware I/O (ports or mmio), IRQs
- EFI (aarch64) or multiboot (x86_64)

Does it work?

This slide deck is being presented from a Raspberry Pi 4 running Helios :D

- Kernel ported to aarch64 in about 42 days(!)
- GPU & serial drivers in userspace
- Slide deck on an initramfs-like tarball
- No hacks, no SoC-specific builds, uses EFI + device tree

What's next?

The kernel is mostly "done". Still needs:

- Polish
- About 100 // TODOs
- SMP support
- riscv64 port
- More bootloader options
- Better docs

Big picture

Get in loser, we're going to userspace

- Mercury: Driver framework
- Venus: Driver collection
- Gaia: Userspace interface
- Luna: POSIX compatibility layer
- **Ares**: Complete operating system

Acknowledgements

Shoutout to early Hare kernel attempts from Ember Sawady and Alexey Yerin!

Big thanks to the Hare community as well: almost 80 contributors!

The #osdev community on Libera Chat is GOAT

We stole a bunch of ideas from seL4 too

Want to learn more?

Full-length Helios talk at 13:00 tomorrow in H.1308 (Rolin)
Hare BoF session at 15:00 today in UB2.147

Kernel hacking is fun! Hare is fun! Let's all have fun together!

`https://ares-os.org`
`https://sr.ht/~sircmpwn/helios`
`https://harelang.org`
IRC: #helios on Libera Chat