

An introduction to Nushift

David Pollack

@davepollack@mastodon.social

Some motivation

Nushift

- An attempt to be an alternative to the web that has the advantages of shareability and accessibility of URLs, but uses different technologies to the web
- Apps are RISC-V binaries, interacting with an ABI/API defined by us
- Focus is on a small/strict API, and apps doing their own visual layout, with support for an accessibility tree
- Hypervisor is written in Rust, apps (at the moment) in Zig

Nushift



Hello World App



Hello World App



Hello, world!

Key hypervisor/app
technologies

SHM

- “Shared memory”
- An SHM capability is a number of contiguous memory pages
- An SHM capability’s memory can be mapped into either the hypervisor or an app’s logical thread, but not both at the same time. However, mapping/unmapping should be cheap
- Powers deferred calls, app-allocated stack and other app memory, etc.

Deferred calls

- Philosophy: As many syscalls as possible should be asynchronous (credit: Managarm OS)
- A deferred syscall accepts an `input_shm_cap_id` and `output_shm_cap_id`, and returns a `task_id`
- The `BlockOnDeferredTasks` syscall is used to wait for tasks to complete, and it destroys its input `task_ids`, and makes the `input_shm_cap_id(s)` and `output_shm_cap_id(s)` available again

```

fn mainImpl() (writing.FBSWriteError || os_nushift.SyscallError || GfxOutput.Error || qoi.DecodeError || Allocator.Error)!usize {
    var title_outputs_tasks = blk: {
        var title_task = try TitleTask.init();
        errdefer title_task.deinit();

        var gfx_get_outputs_task = try GfxGetOutputsTask.init();
        errdefer gfx_get_outputs_task.deinit();

        break :blk .{ title_task, gfx_get_outputs_task };
    };

    const title_task_id = try title_outputs_tasks[0].titlePublish();
    const gfx_get_outputs_task_id = try title_outputs_tasks[1].gfxGetOutputs();

    // If an error occurs between publishing and the end of
    // block_on_deferred_tasks, you can't deinit the tasks because the resources
    // are in-flight. And we don't. But that does mean the task resources will
    // leak if that error occurs.

    try blockOnDeferredTasks(&.{ title_task_id, gfx_get_outputs_task_id });

    // Rest of main omitted...
}

```


Future of Nushift

- Networking

- It should be as easy to start a secure server serving Nushift programs as it is to start an SSH server.
- I am currently researching the Noise protocol framework by Trevor Perrin + QUIC (nQUIC) and PQNoise

- RISC-V/WebAssembly/x86-64/aarch64

- My original vision was devs would cross-compile to x86-64 and aarch64, with the user agent requesting only the relevant one. Not RISC-V or wasm
- Initially couldn't overcome sandboxing issues
- I am currently researching a hardware virtualisation approach

Thanks!

<https://github.com/davepollack/nushift>

@davepollack@mastodon.social