Stage Summary for ArceOS-Hypervisor

2024/5/26

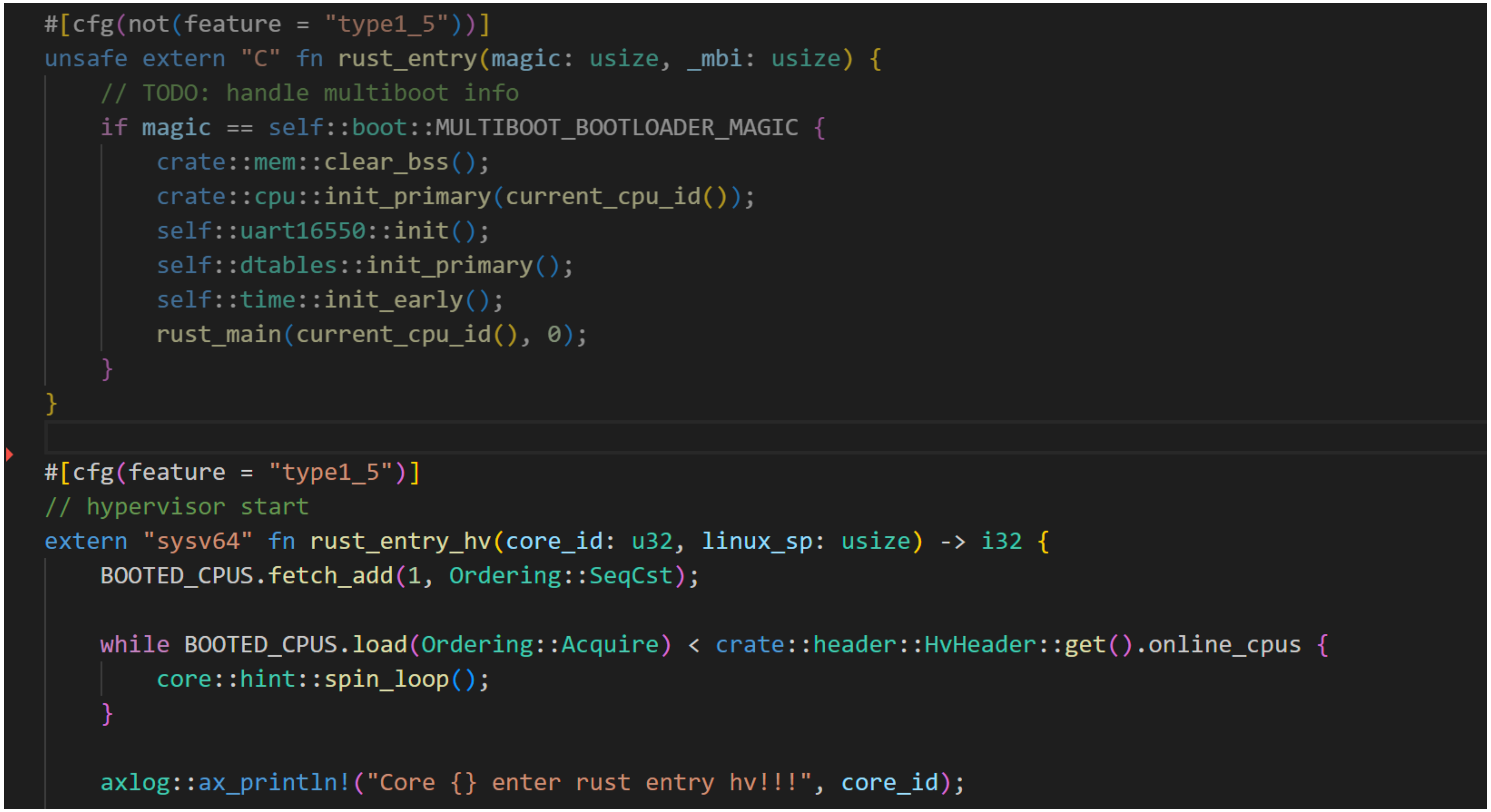
Problems & Challenges

- axhal imports a new (half-new) platform
- hypervisor inside ArceOS or hypervisor as a App
- git submodule
- Resource & Management
- Error Handling & Transmission

axhal imports a new (half-new) platform

- Different Entry
  - ArceOS: Boot from bare-metal hardware
  - ArceOS-HV(Type1.5): Boot from Linux

- How to achieve
  - `modules/axhal/src/platform/pc_x86`
    i. features
      - ugly code style



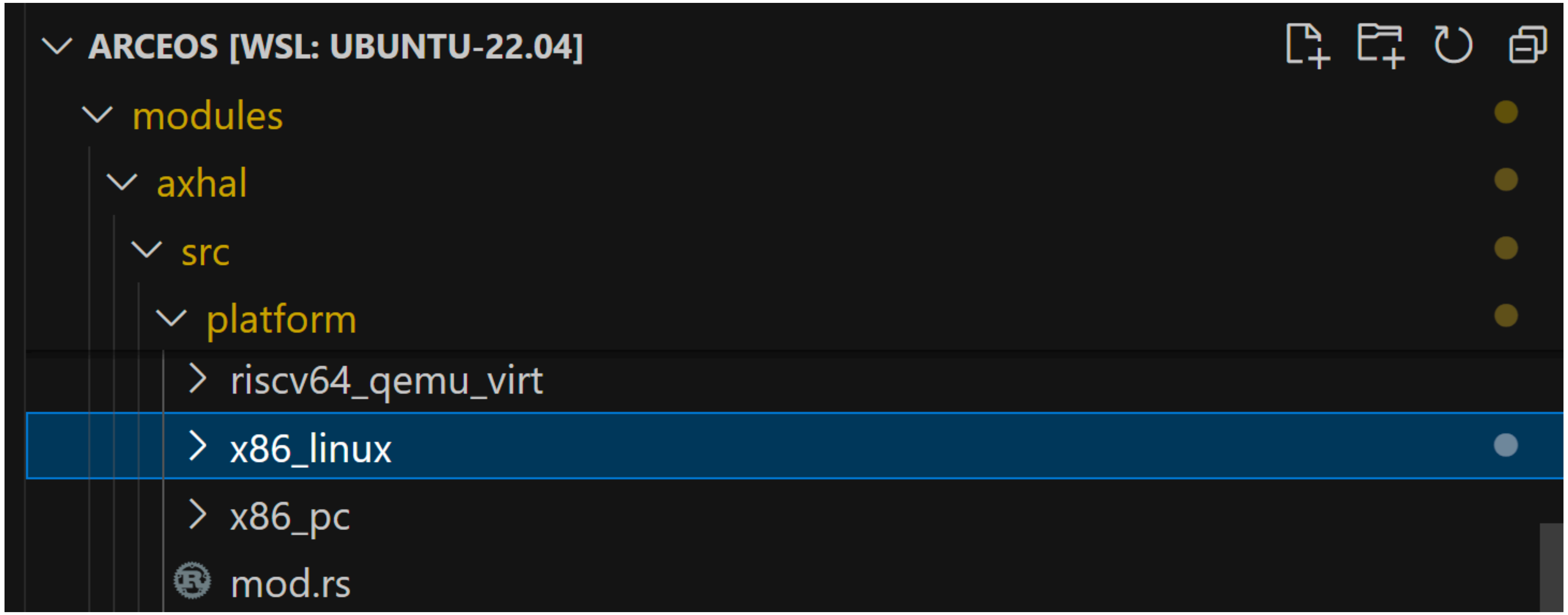
```rust
#[cfg(not(feature = "type1_5"))]
unsafe extern "C" fn rust_entry(magic: usize, _mbi: usize) {
    // TODO: handle multiboot info
    if magic == self::boot::MULTIBOOT_BOOTLOADER_MAGIC {
        crate::mem::clear_bss();
        crate::cpu::init_primary(current_cpu_id());
        self::uart16550::init();
        self::dtables::init_primary();
        self::time::init_early();
        rust_main(current_cpu_id(), 0);
    }
}

#[cfg(feature = "type1_5")]
// hypervisor start
extern "sysv64" fn rust_entry_hv(core_id: u32, linux_sp: usize) -> i32 {
    BOOTED_CPUS.fetch_add(1, Ordering::SeqCst);

    while BOOTED_CPUS.load(Ordering::Acquire) < crate::header::HvHeader::get().online_cpus {
        core::hint::spin_loop();
    }

    axlog::ax_println!("Core {} enter rust entry hv!!!", core_id);
```

- How to achieve
  - `modules/axhal/src/platform/pc_x86`
    i. features
    ii. A different platform
      - redundant codes



ARCEOS [WSL: UBUNTU-22.04]
modules
axhal
src
platform
riscv64_qemu_virt
x86_linux
x86_pc
mod.rs

hypervisor inside ArceOS or hypervisor as an App

- Fundamental Problem: Extending Unikernel to Hypervisor
- Current state: Ugly mixed architecture

hypervisor inside ArceOS or hypervisor as an App

- Current state: Ugly mixed architecture
- Current boot process of ArceOS-HV

1. `_start` & `switch_stack` (modules/axhal/src/platform/pc_x86/boot_type15.rs)
2. `rust_entry_hv` (modules/axhal/src/platform/pc_x86/mod.rs)
3. `rust_main` (modules/axruntime/src/lib.rs)
4. `main` (apps/hv/src/main.rs)
5. `config_boot_linux` (modules/axvm/src/vm.rs)
    i. `vm.run_type15_vcpu(hart_id, &linux_context)`

hypervisor inside ArceOS or hypervisor as an App

- Current state: Ugly mixed architecture
- Current boot process of ArceOS-HV
- Problem
  - Each core construct its own independent VM structure.
  - No way to support multiple vCPU on a phycial core and perform scheduling.
  - Lack of flexibility.

hypervisor inside ArceOS or hypervisor as an App

- Expected Startup Procedure
    1. axhal
    2. axruntime
    3. `main` in `apps/hv`
    4. Keep all hypervisor functionalities within the app

git submodule

```
[submodule "crates/hypercraft"]
        path = crates/hypercraft
        url = git@github.com:arceos-hypervisor/hypercraft.git
```

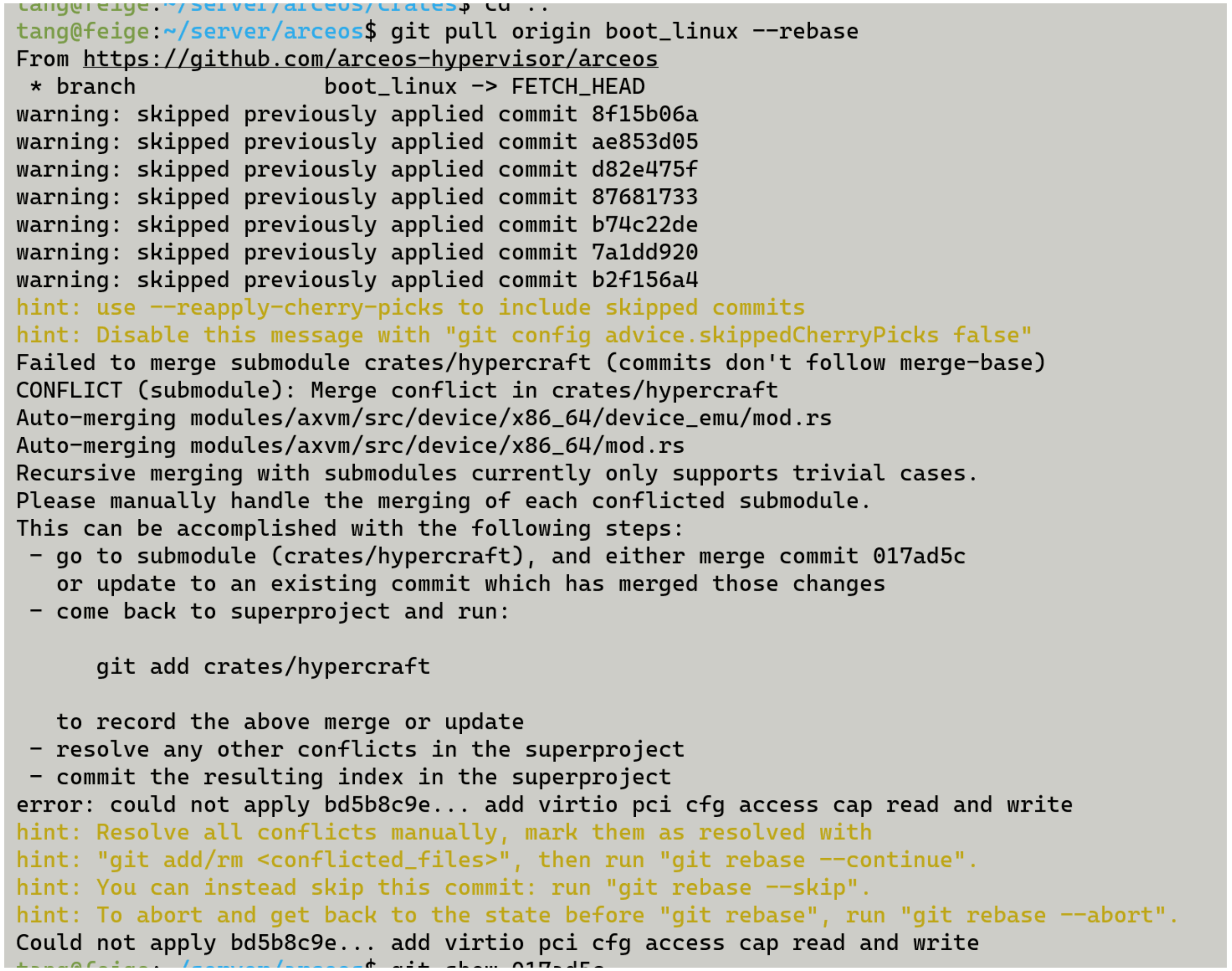
- No apparent obstacles during development.
- Obstacle exists in version control.

- Obstacle exists in version control

  > The parent repository stores the commit hash of each submodule, not the code of the submodule itself

  - push code
    - commit submodule (hypercraft)
    - commit parent repository (arceos)
  - merge & rebase code

- Obstacle exists in version control



```
tang@feige:~/server/arceos/crates$ cd ..
tang@feige:~/server/arceos$ git pull origin boot_linux --rebase
From https://github.com/arceos-hypervisor/arceos
 * branch              boot_linux -> FETCH_HEAD
warning: skipped previously applied commit 8f15b06a
warning: skipped previously applied commit ae853d05
warning: skipped previously applied commit d82e475f
warning: skipped previously applied commit 87681733
warning: skipped previously applied commit b74c22de
warning: skipped previously applied commit 7a1dd920
warning: skipped previously applied commit b2f156a4
hint: use --reapply-cherry-picks to include skipped commits
hint: Disable this message with "git config advice.skippedCherryPicks false"
Failed to merge submodule crates/hypercraft (commits don't follow merge-base)
CONFLICT (submodule): Merge conflict in crates/hypercraft
Auto-merging modules/axvm/src/device/x86_64/device_emu/mod.rs
Auto-merging modules/axvm/src/device/x86_64/mod.rs
Recursive merging with submodules currently only supports trivial cases.
Please manually handle the merging of each conflicted submodule.
This can be accomplished with the following steps:
 - go to submodule (crates/hypercraft), and either merge commit 017ad5c
   or update to an existing commit which has merged those changes
 - come back to superproject and run:

      git add crates/hypercraft

   to record the above merge or update
 - resolve any other conflicts in the superproject
 - commit the resulting index in the superproject
error: could not apply bd5b8c9e... add virtio pci cfg access cap read and write
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply bd5b8c9e... add virtio pci cfg access cap read and write
```

- Obstacle exists in version control
  - push code
  - merge&rebase code
    - merge submodule(hypercraft itself)
    - merge&rebase arceos
      - conflict in submodule

        ```
        git add crates/hypercraft
        git rebase --continue
        ```

Resource & Management

Root Case: ArceOS-HV was based an independent project, hypercraft.

Resource & Management

Root Case: ArceOS-HV was based an independent project, hypercraft.

- Expected architecture:
  - hypercraft: architectural-related virtualization functionality
  - ArceOS-HV: construct a hypervisor utilizing the foundational functionalities exposed by hypercraft.
- Overall speaking:
  - vCPU implemented by hypercraft
  - VM implemented by ArceOS-HV

Resource & Management

Root Case: ArceOS-HV was based an independent project, hypercraft.

- Chaos architecture:
  - Guest VM resources like `GuestPhysMemorySet` and `GuestPageTable` were are by `axvm` module.
  - vcpu and vm structure are exposed by hypercraft, maneged by `axvm`.
  - `vm.run_vcpu()` is called inside `axvm` module, `run_vcpu` method is exposed by hypercraft's VM structure.
  - `apps/hv` does nothing but called `config_boot_linux` inside `axvm` module.

Resource & Management

- Chaos architecture
- Problem:
  - Catastrophic resource management logic.
  - No way to operate VM resource inside hypercraft.
  - Each modification requires to change codes from both ArceOS-HV and hypercraft.
- Example
  - The implementation of instruction decoding

Resource & Management

- Good architecture:
  - `PerCpuDevices` and `PerVMDevices` are exposed by hypercraft as `Trait`, implemented inside `axvm`.
    - decoupling emulated device implementation from hypercraft.
    - Allowing ArceOS-HV's customization of emulated device.

Resource & Management

Core Problem

- How to decouple the implementation of virtualization architecture-related functionalities from resource management and runtime flow control ?

Error Handling & Transmission

- error types
    - axerrno: `AxError` and `AxResult`
    - hypercraft: `HyperError`
- The combination of the use of these error types seems akward.
- Loss where exactly did this error happen during bottom-up error propagation.
- @Su Mingxian suggests we can use anyhow crate for error handling.

Stage Summary

- Boot from Linux with the help of Jailhouse kernel module.
- Boot NimbOS and ArceOS as guest VM.
- Boot secondary Linux (slightly modified kernel) with ramdisk file system as guest VM.
- Boot on QEMU and real x86 hardwares.
- Some docs

To be implemented

- Refactor (**modularity**)
- Migrating to ARM and RISC-V (**modularity**)
- More emulated device (**modularity**)
    - virtual local APIC for supporting multiple vCPU on the same pCPU
    - virtio devices for more functional guest VM
- **Intel VTD** for irq remapping and device memory remapping
- The compatibility with vanilla Linux