SUSTech Southern University of Science and Technology

Teecert Labs

Revolutionizing System Support:

Supporting Firecracker Virtualization for Jinux Platform

CS321 Group Project – Presentation II

Boot into Rust

Teecert Labs

SUSTech Southern University of Science and Technology

```
$ ../firecracker/build/cargo_target/x86_64-unknown-linux-musl/debug/firecracker --no-api --show-level --config-file config.json
2023-12-06T22:50:02.095633926 [anonymous-instance:main:INFO] Running Firecracker v1.6.0-dev
2023-12-06T22:50:02.157250648 [anonymous-instance:main:INFO:src/vmm/src/device_manager/mmio.rs:386] Artificially kick devices.
2023-12-06T22:50:02.157300648 [anonymous-instance:main:INFO:src/firecracker/src/main.rs:574] Successfully started microvm that
was configured from one single json
[INFO]: Found usable region, start:0, end:9fc00
[INFO]: Found usable region, start:100000, end:d0000000
[INFO]: Found usable region, start:100000000, end:130000000
[INFO]: Initializing trapframe...
[DEBUG]: new gdt:[0, 0, 0, 0, 0, 0, 0, 0, ff, ff, 0, 0, 0, 9a, af, 0, ff, ff, 0, 0, 0, 93, cf, 0, ff, ff, 0, 0, 0, 9a, cf, 0, 6
7, 0, 0, e0, d3, 89, 0, 88, ff, ff, ff, ff, 0, 0, 0, 0, 0, 0, 0, 0, 0, 98, 20, 0, 0, 0, 0, 0, 0, 92, 0, 0, ff, ff, 0, 0, 0, fa,
 cf, 0, ff, ff, 0, 0, 0, f2, cf, 0, 0, 0, 0, 0, 0, f8, 20, 0], entry_count:11
[INFO]: GDT initialization completed
[INFO]: IDT initialization completed
[INFO]: Syscall related register initialization completed
[WARN]: ACPI info not found!
[INFO]: x2APIC ID:0, Version:14, Max LVT:5
[INFO]: [IOAPIC]: Not found ACPI talbes, using default address:fec00000
[INFO]: [IOAPIC]: IOAPIC id: 0. version:17. max redirection entry:24. interrupt base:0
```

SUSTech Southern University of Science and Technology

Continued

Timer Initialization

Teecert Labs

SUSTech Southern University of Science and Technology

APIC Timer

High precision
Unknown frequency

PIT Timer

Low precision
Known frequency

Timer Initialization

Teecert Labs

SUSTech Southern University of Science and Technology

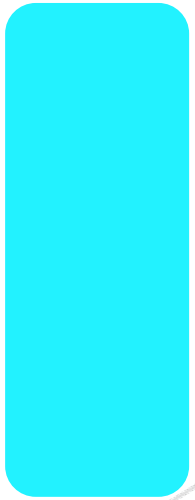
APIC Timer

High precision
Unknown frequency

x cycles

PIT Timer

Low precision
Known frequency



y cycles

Timer Initialization

Teecert Labs

SUSTech Southern University of Science and Technology

APIC Timer

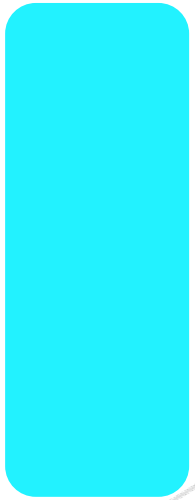
High precision
Unknown frequency

x cycles

Freq = ...

PIT Timer

Low precision
Known frequency



y cycles

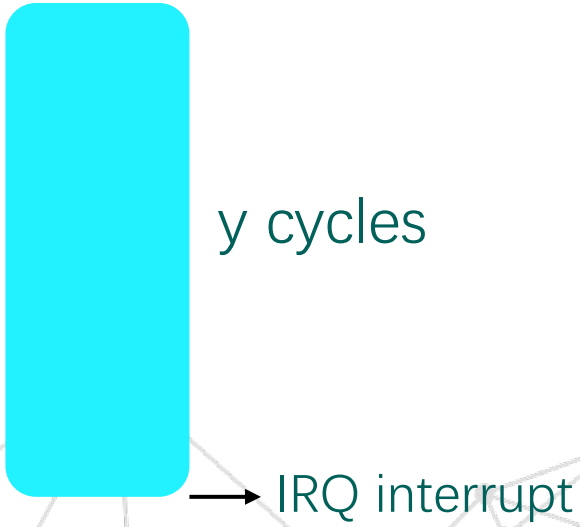
Timer Initialization

Teecert Labs SUSTech Southern University of Science and Technology

```
100
101  fn init_periodic_mode() {
102      let mut apic_lock = APIC_INSTANCE.get().unwrap().lock();
103      let mut irq = IrqLine::alloc_specific(super::TIMER_IRQ_NUM.load(
104      irq.on_active(init_function);
105      let mut io_apic = IO_APIC.get().unwrap().first().unwrap().lock()
106      debug_assert_eq!(io_apic.interrupt_base(), 0);
107      io_apic.enable(2, irq.clone()).unwrap();
108      drop(io_apic);
109      // divide by 64
110      apic_lock.set_timer_div_config(DivideConfig::Divide64);
111      apic_lock.set_timer_init_count(0xFFFF_FFFF);
112      drop(apic_lock);
113      super::pit::init();
```

PIT Timer

Low precision
Known frequency



Timer Initialization

arch > x86 > kvm > C i8254.c

```
12  struct kvm_pit *kvm_create_pit(struct kvm *kvm, u32 flags)
11  {
10      struct kvm_pit *pit;
 9      struct kvm_kpit_state *pit_state;
 8      struct pid *pid;
 7      pid_t pid_nr;
 6      int ret;
 5
 4      pit = kzalloc(sizeof(struct kvm_pit), GFP_KERNEL_ACCOUNT);
 3      if (!pit)
 2          return NULL;
 1
674     pit->irq_source_id = kvm_request_irq_source_id(kvm);
 1      if (pit->irq_source_id < 0)
 2          goto fail_request;
 3
 4      mutex_init(&pit->pit_state.lock);
```

First unused
That is, 0

Timer Initialization

Teecert Labs

SUSTech Southern University of Science and Technology

```
[WARN]: ACPI info not found!
[INFO]: x2APIC ID:0, Version:14, Max LVT:5
[INFO]: [IOAPIC]: Not found ACPI talbes, using default address:fec00000
[INFO]: [IOAPIC]: IOAPIC id: 0, version:17, max_redirection_entry:24, interrupt base:0
[INFO]: [Timer]: Enable APIC periodic mode.
[INFO]: APIC Timer ticks count:7b79, remain ticks: ffff02bf,Timer Freq:500 Hz
[WARN]: IOMMU initialization error:NoIommu
[INFO]: PIC init, master mask:ff slave_mask:ff
```

VirtIO Devices

Teecert Labs

SUSTech Southern University of Science and Technology

Jinux can not find VirtIO devices of Firecracker

Jinux search area:

```
22
23      pub fn init() {
24          // FIXME: The address 0xFEB0_0000 is obtained from an instance of microvm, and it may not work in other architecture.
25          iter_range(0xFEB0_0000..0xFEB0_4000);
26      }
27
```

VirtIO Devices

Teecert Labs

SUSTech Southern University of Science and Technology

Jinux search area:

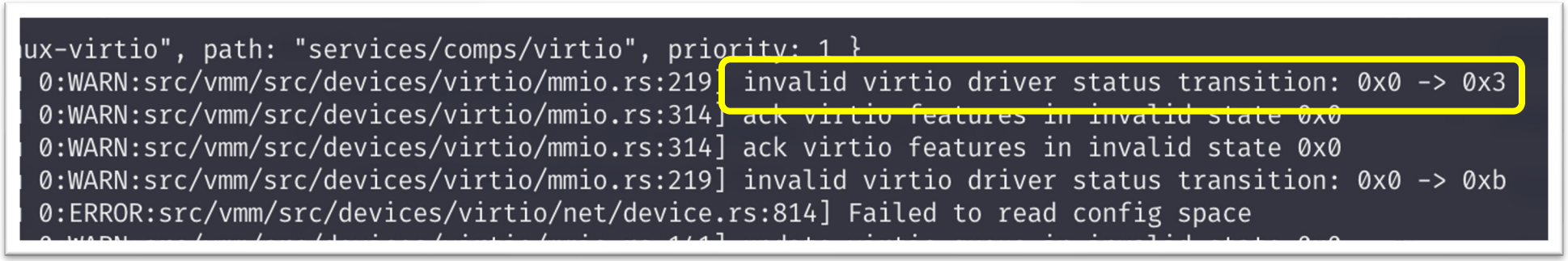
```
23    pub fn init() {
24        // FIXME: The address 0xFEB0_0000 is obtained from an instance of microvm, and it may not work in other architecture.
25        iter_range(0xFEB0_0000..0xFEB0_4000);
26    }
```

Firecracker VirtIO area:

```
56      const FIRST_ADDR_PAST_32BITS: u64 = 1 << 32;
57      /// Size of MMIO gap at top of 32-bit address space.
58      pub const MEM_32BIT_GAP_SIZE: u64 = 768 << 20;
59      /// The start of the memory area reserved for MMIO devices.
60      pub const MMIO_MEM_START: u64 = FIRST_ADDR_PAST_32BITS - MEM_32BIT_GAP_SIZE;
61      /// The size of the memory area reserved for MMIO devices.
62      pub const MMIO_MEM_SIZE: u64 = MEM_32BIT_GAP_SIZE;
```

i.e. `0xD0000000` to `0xFFFFF000`

VirtIO Devices



VirtIO Devices

Teecert Labs SUSTech Southern University of Science and Technology

```
ux-virtio", path: "services/comps/virtio", priority: 1 }
 0:WARN:src/vmm/src/devices/virtio/mmio.rs:219] invalid virtio driver status transition: 0x0 -> 0x3
 0:WARN:src/vmm/src/devices/virtio/mmio.rs:314] ack virtio features in invalid state 0x0
 0:WARN:src/vmm/src/devices/virtio/mmio.rs:314] ack virtio features in invalid state 0x0
 0:WARN:src/vmm/src/devices/virtio/mmio.rs:219] invalid virtio driver status transition: 0x0 -> 0xb
 0:ERROR:src/vmm/src/devices/virtio/net/device.rs:814] Failed to read config space
```

One step [EMPTY] → ACKNOWLEDGE + DRIVER

```
30          transport::init();
31          while let Some(mut transport) = pop_device_transport() {
32              // Reset device
33              transport.set_device_status(DeviceStatus::empty()).unwrap();
34              // Set to acknowledge
35              transport
36                  .set_device_status(DeviceStatus::ACKNOWLEDGE | DeviceStatus::DRIVER)
37                  .unwrap();
38              // negotiate features
39              negotiate_features(&mut transport);
```

VirtIO Devices

Teecert Labs

SUSTech Southern University of Science and Technology

One step [EMPTY] → ACKNOWLEDGE + DRIVER

```
transport::init();
while let Some(mut transport) = pop_device_transport() {
    // Reset device
    transport.set_device_status(DeviceStatus::empty()).unwrap();
    // Set to acknowledge
    transport
        .set_device_statu
        .unwrap();
    // negotiate features
    negotiate_features(&m
```

Firecracker: [EMPTY] → ACKNOWLEDGE → ACKNOWLEDGE + DRIVER

```
175             // match changed bits
176             match !self.device_status & status {
177                 ACKNOWLEDGE if self.device_status == INIT => {
178                     self.device_status = status;
179                 }
180                 DRIVER if self.device_status == ACKNOWLEDGE => {
181                     self.device_status = status;
182                 }
183                 FEATURES_OK if self.device_status == (ACKNOWLEDGE | DRIVER) => {
184                     self.device_status = status;
185                 }
186                 DRIVER_OK if self.device_status == (ACKNOWLEDGE | DRIVER | FEATURES
187                     self.device_status = status;
188                     let device_activated = self.locked_device().is_activated();
```

VirtIO Devices

Teecert Labs SUSTech Southern University of Science and Technology

One step [EMPTY] → ACKNOWLEDGE + DRIVER

```
transport::init();
while let Some(mut transport) = pop_device_transport() {
    // Reset device
    transport.set_device_status(DeviceStatus::empty()).unwrap();
    // Set to acknowledge
    transport
        .set_device_s
        .unwrap();
    // negotiate feat
    negotiate_feature
```

Firecracker: [EMPTY] → ACKNOWLEDGE → ACKNOWLEDGE + DRIVER

```
175             // match changed bits
176             match !self.device_status & status {
177                 ACKNOWLEDGE if self.device_status == INIT => {
178                     self.device_status = status;
179                 }
```

```
EDGE => {

CKNOWLEDGE | DRIVER) => {

NOWLEDGE | DRIVER | FEATURES

device().is_activated();
```

```
diff --git a/services/comps/virtio/src/lib.rs b/services/comps/virtio/src/lib.rs
index aee89fc..0716e61 100644
--- a/services/comps/virtio/src/lib.rs
+++ b/services/comps/virtio/src/lib.rs
@@ -32,6 +32,10 @@ fn virtio_component_init() -> Result<(), ComponentInitError> {
         // Reset device
         transport.set_device_status(DeviceStatus::empty()).unwrap();
         // Set to acknowledge
+        // firecracker/src/vmm/src/devices/virtio/mmio.rs:173
+        transport
+            .set_device_status(DeviceStatus::ACKNOWLEDGE)
+            .unwrap();
         transport
             .set_device_status(DeviceStatus::ACKNOWLEDGE | DeviceStatus::DRIVER)
             .unwrap();
```

Summary

1. Implement Linux64 Boot Protocol
2. Fix APIC Timer Initialization
3. Correct VirtIO Device Discovery
4. Amend VirtIO Device Initialization
5. ... Other Code Modifications

Hopefully merged into the Jinux mainline

Teecert Labs

SUSTech Southern University of Science and Technology

Thanks