

# Upstreaming Tyr: a DRM GPU Kernel Driver in Rust

**Deborah Brouwer**

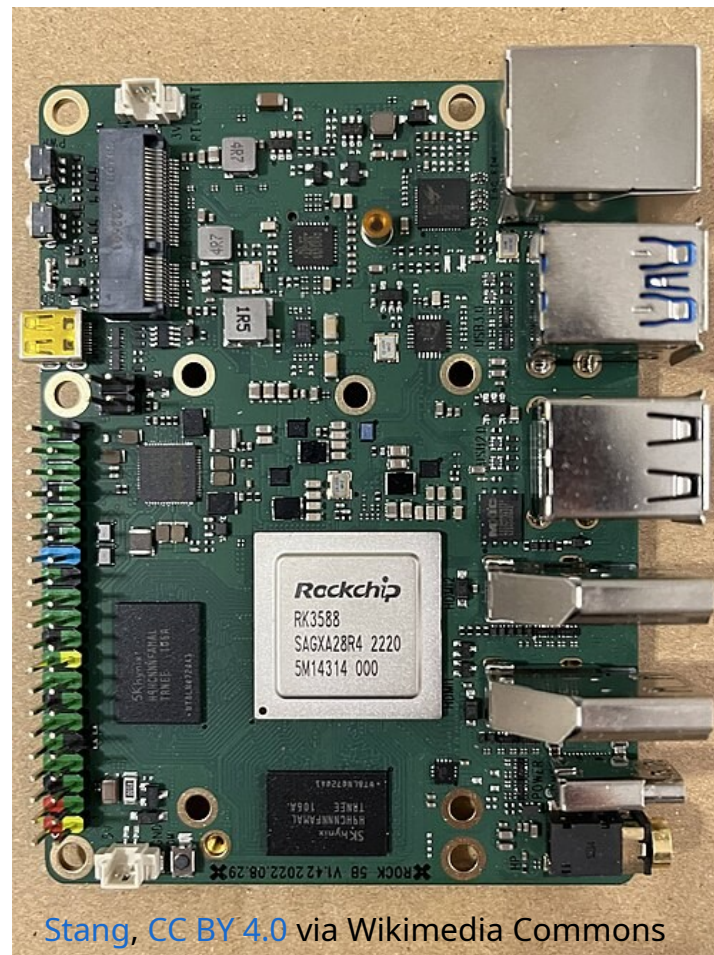


COLLABORA

**Open First**

# What is Tyr?

- Kernel driver for certain Arm Mali GPUs
- Valhall architecture, 3<sup>rd</sup> gen, major v.10
- GPU Microcontroller (MCU)
- Command Stream Frontend (CSF) interface
- Radxa ROCK 5B, RK3588 SOC, Mali GPU model G610
- Goal: Port existing C driver, Panthor to Rust
- Benefit: focus on Rust implementation



# Upstream Progress

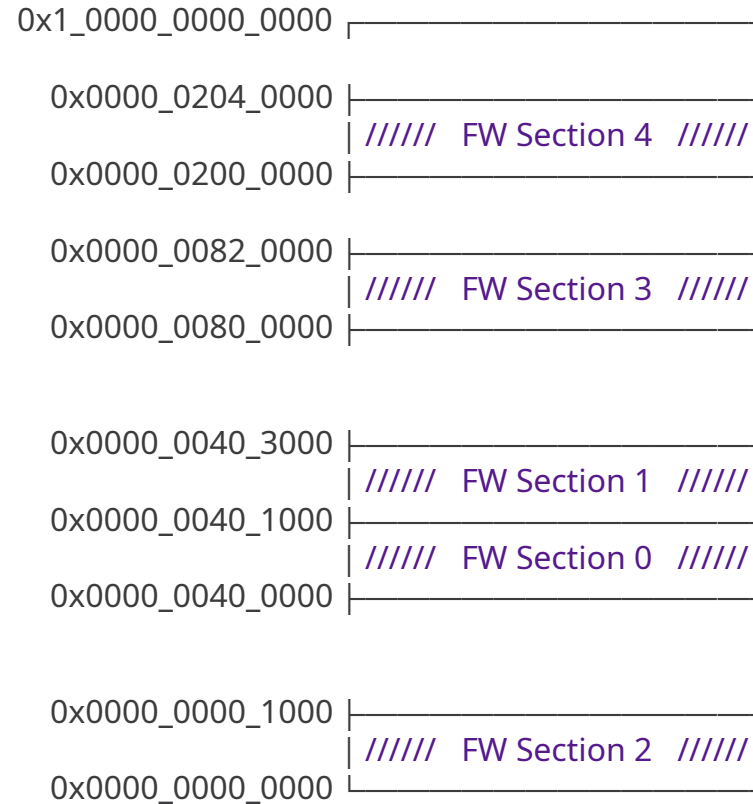
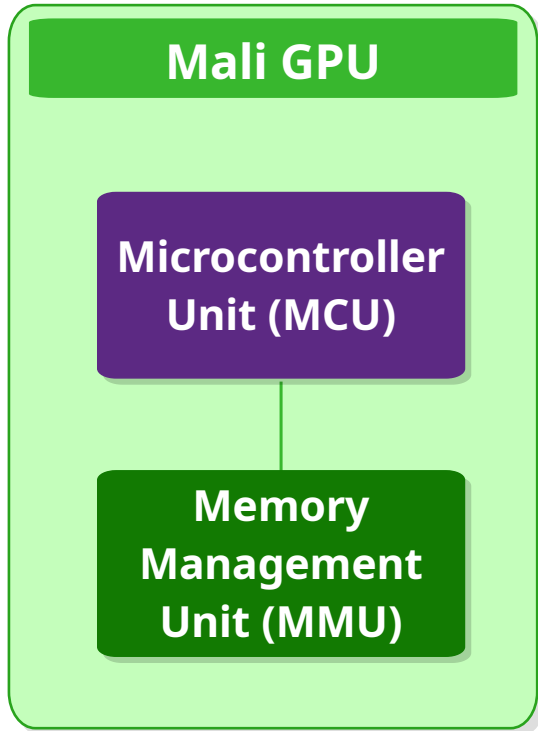


# GEM

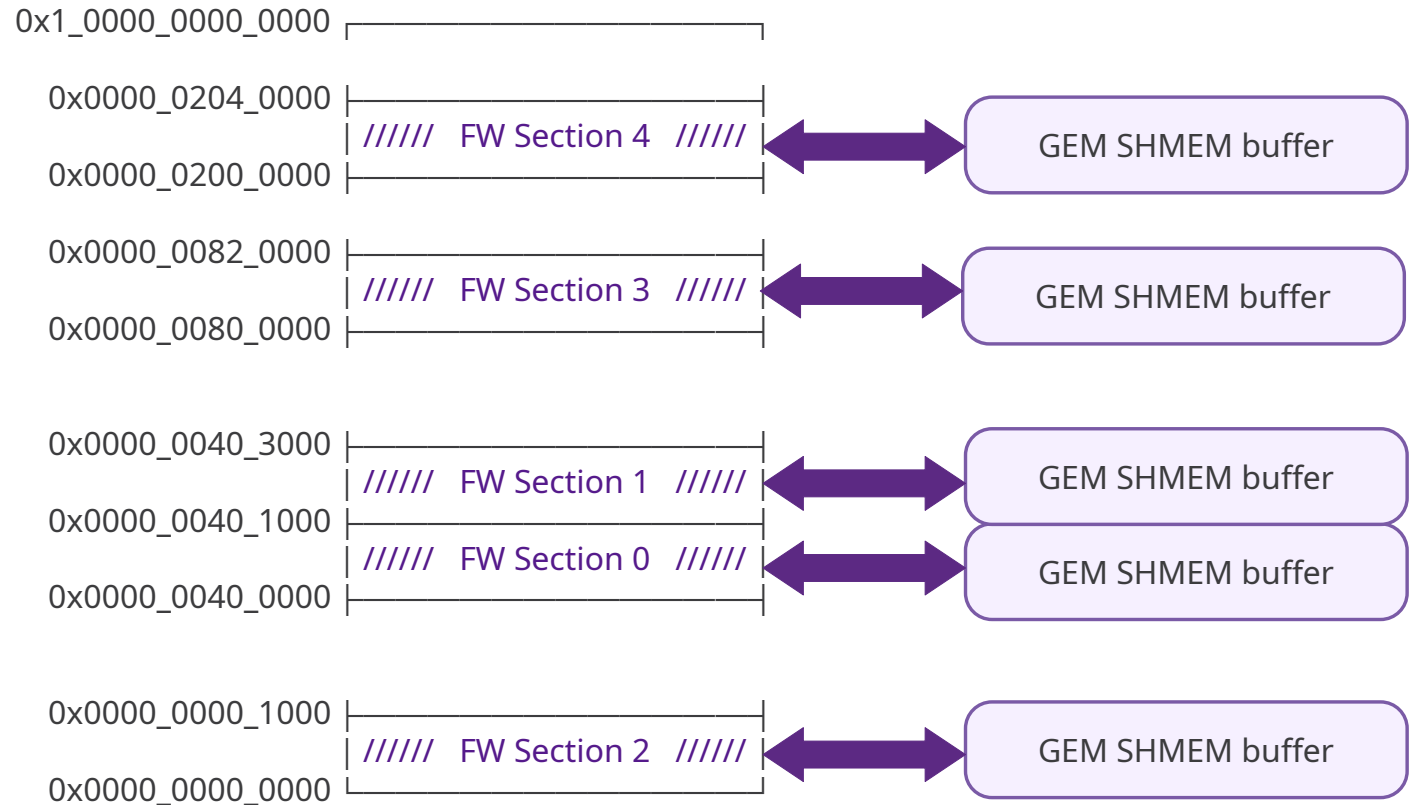
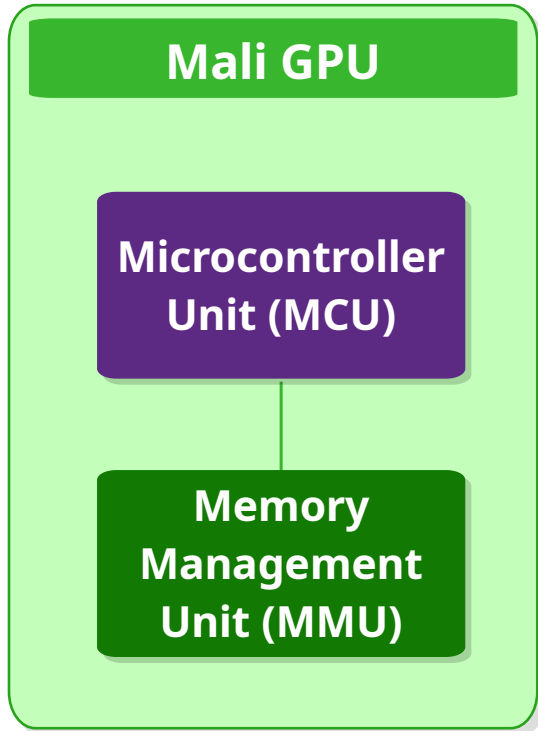


Moissanite (synthetic silicon carbide) 3 by James St. John, CC BY 2.0 Modified: removed background; adjusted colour.

# Virtual Memory



# Page Table Mapping



# Safe vs Unsafe



# Safety Justification

```
pub fn new(dev: &'a Device<Bound>, config: Config) -> Result<IoPageTable<'a, F>> {  
  // SAFETY:  
  // * The raw_cfg pointer is valid for the duration of this call.  
  // * The provided `FLUSH_OPS` contains valid function pointers that accept a null pointer  
  //   as cookie.  
  // * The caller ensures that the io pgtable does not outlive the device.  
  let ops = unsafe {  
    bindings::alloc_io_pgtable_ops(F::FORMAT, &mut raw_cfg, core::ptr::null_mut())  
  };
```

# Safety Requirement

```
/// # Safety
```

```
///
```

```
/// * No other io-pgtable operation may access the range `iova .. iova+pgsize*pgcount` while  
/// this `map_pages` operation executes.
```

```
///
```

```
/// * This page table must not contain any mapping that overlaps with the mapping created by  
/// this call.
```

```
///
```

```
/// * If this page table is live, then the caller must ensure that it's okay to access the  
/// physical address being mapped for the duration in which it is mapped.
```

```
#[inline]
```

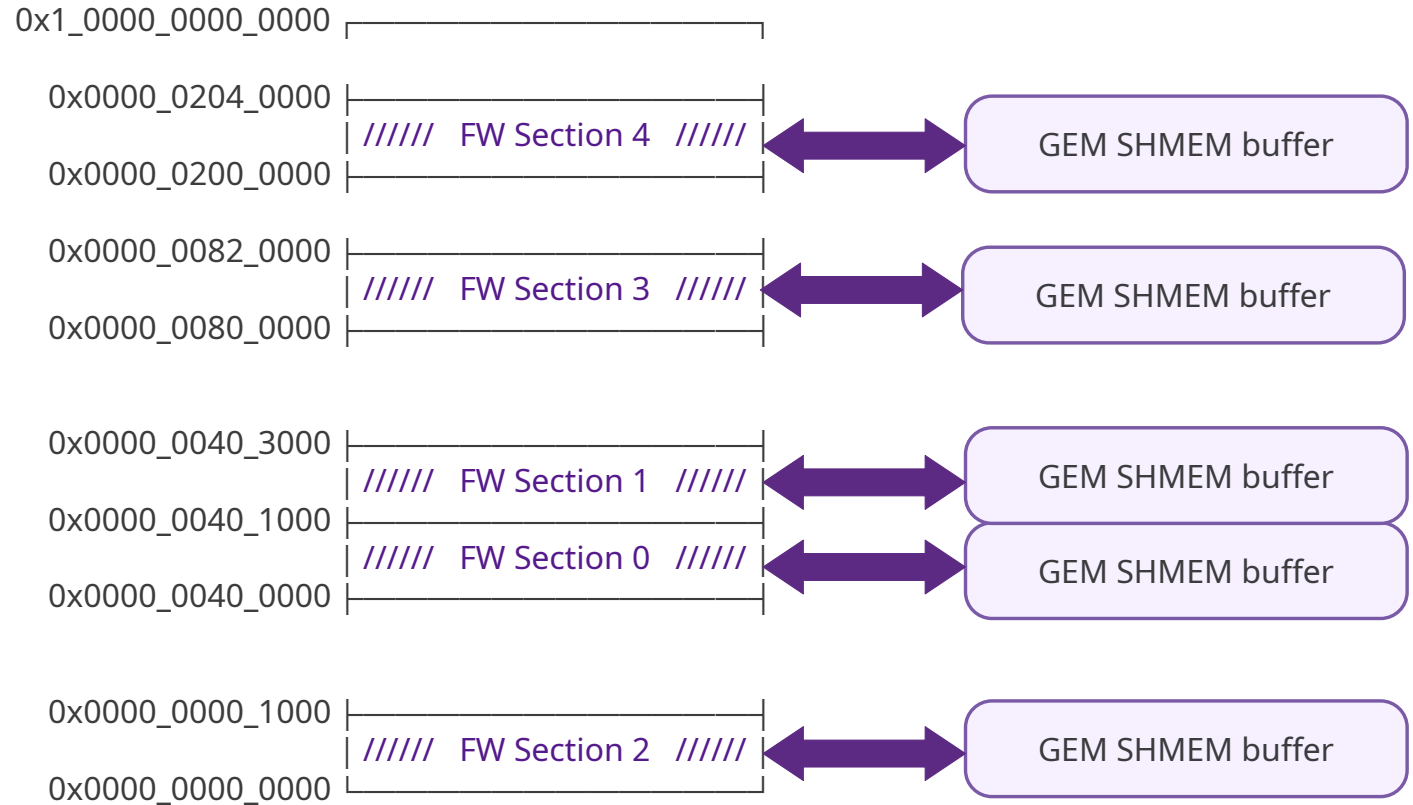
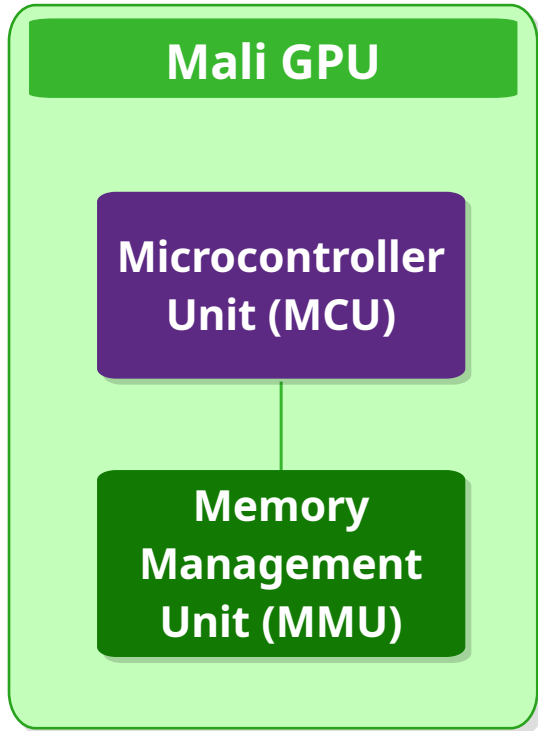
```
pub unsafe fn map_pages(
```



# Unsafe Rust?

```
static int __arm_lpaemap(  
...  
    if (pte && !iopte_leaf(pte, lvl, data->iop.fmt)) {  
        cptep = iopte_deref(pte, data);  
    } else if (pte) {  
        /* We require an unmap first */  
        WARN_ON(!(cfg->quirks & IO_PGTABLE_QUIRK_NO_WARN));  
        return -EEXIST;  
    }  
}
```

# How do we know this is safe?



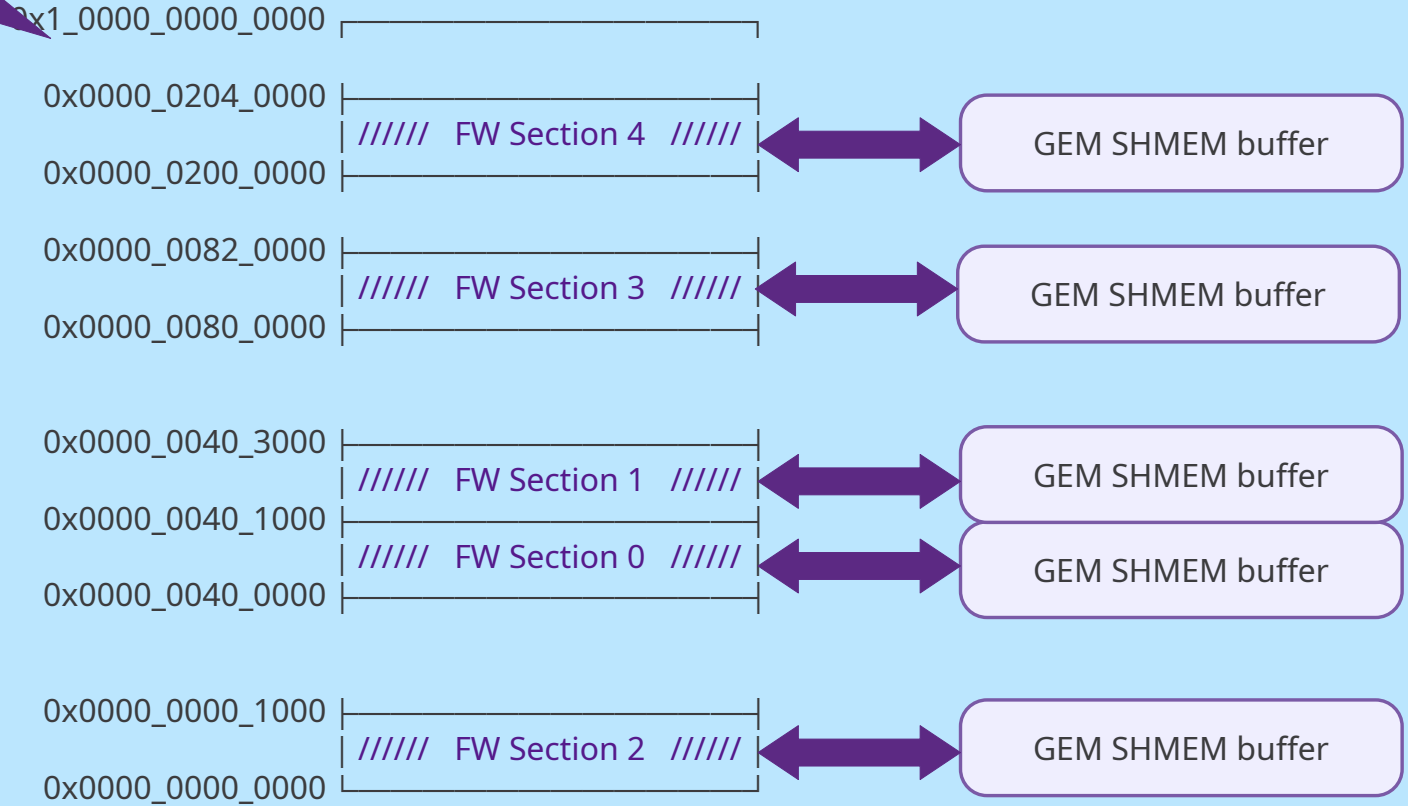
# GPUVM

**SAFE**

**Mali GPU**

**Microcontroller  
Unit (MCU)**

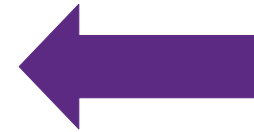
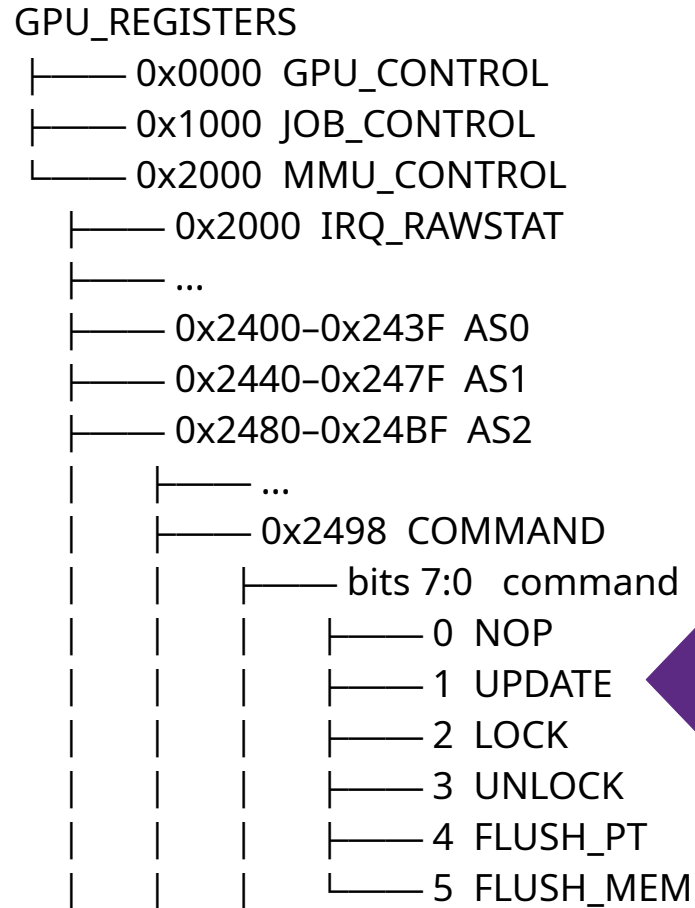
**Memory  
Management  
Unit (MMU)**



COLLABORA

**Open First**

# Writing to a GPU Register Address



# Panthor

```
#define MMU_AS_BASE      0x2400
#define MMU_AS(as)      ((as) << MMU_AS_SHIFT)
#define MMU_AS_SHIFT     6
#define AS_COMMAND(as)  (MMU_AS(as) + 0x18)
#define AS_COMMAND_UPDATE 1
```

```
gpu_write(ptdev->mmu->iomem, AS_COMMAND(as_nr), AS_COMMAND_UPDATE);
```

# Tyr

```
const MAX_AS: usize = 16;  
const STRIDE: usize = 0x40;
```

```
pub(crate) COMMAND(u32)[MAX_AS, stride = STRIDE] @ 0x2418 {  
    7:0    command ?=> MmuCommand;  
}
```

```
pub(crate) enum MmuCommand {  
    Nop = 0,  
    Update = 1,  
    Lock = 2,  
    Unlock = 3,  
    FlushPt = 4,  
    FlushMem = 5,  
}
```

```
COMMAND::try_at(index).ok_or(EINVAL)?  
io.write(command_reg, COMMAND::zeroed().with_command(MmuCommand::Update));
```

```
impl TryFrom<Bounded<u32, 8>> for MmuCommand {  
    type Error = Error;  
    fn try_from(val: Bounded<u32, 8>) -> Result<Self, Self::Error> {  
        match val.get() {  
            0 => Ok(MmuCommand::Nop),  
            1 => Ok(MmuCommand::Update),  
            2 => Ok(MmuCommand::Lock),  
            3 => Ok(MmuCommand::Unlock),  
            4 => Ok(MmuCommand::FlushPt),  
            5 => Ok(MmuCommand::FlushMem),  
            _ => Err(EINVAL),  
        }  
    }  
}  
  
impl From<MmuCommand> for Bounded<u32, 8> {  
    fn from(cmd: MmuCommand) -> Self {  
        (cmd as u8).into()  
    }  
}
```



# Take Away...

- We have come a long way to boot the MCU.
- We are still not done:
  - irqs and CSF interface
  - userspace ioctls
  - power management
  - job scheduling
- Being an end user helps those Rust APIs to develop.
- Once Rust stabilizes, it will be much faster and easier to write a kernel driver.



"The Binding of Fenrir" by Trish Baer, Public Domain Dedication (CC0)



COLLABORA

Open First

# References

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tyr         | <a href="https://support.arm.com/compute-ip/mali-g610#Technical-Specifications">https://support.arm.com/compute-ip/mali-g610#Technical-Specifications</a><br><b>44e7e7f7cffb drm/tyr: add Microcontroller Unit (MCU) booting</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Firmware    | <a href="https://gitlab.com/kernel-firmware/linux-firmware/-/tree/main/arm/mali">https://gitlab.com/kernel-firmware/linux-firmware/-/tree/main/arm/mali</a><br><a href="https://rust.docs.kernel.org/kernel/firmware/index.html">https://rust.docs.kernel.org/kernel/firmware/index.html</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| GEM         | <a href="https://docs.kernel.org/gpu/drm-mm.html">https://docs.kernel.org/gpu/drm-mm.html</a><br><b>c284d3e42338 rust: drm: gem: Add GEM object abstraction</b><br><b>80df573af9ef rust: drm: gem: shmem: Add DRM shmem helper abstraction</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Page Tables | <b>2e2f6b0ef855 rust: iommu: add io_pgtable abstraction</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Unsafe Rust | <a href="https://lore.kernel.org/all/20251112-io-pgtable-v3-1-b00c2e6b951a@google.com">https://lore.kernel.org/all/20251112-io-pgtable-v3-1-b00c2e6b951a@google.com</a><br><a href="https://origin.kernel.org/doc/html/next/rust/coding-guidelines.html">https://origin.kernel.org/doc/html/next/rust/coding-guidelines.html</a><br><a href="https://doc.rust-lang.org/reference/unsafe-keyword.html">https://doc.rust-lang.org/reference/unsafe-keyword.html</a><br><a href="https://doc.rust-lang.org/nomicon/safe-unsafe-meaning.html">https://doc.rust-lang.org/nomicon/safe-unsafe-meaning.html</a><br><a href="https://cdn.kernel.org/doc/html/latest/process/coding-style.html">https://cdn.kernel.org/doc/html/latest/process/coding-style.html</a><br><a href="https://lore.kernel.org/rust-for-linux/20240717221133.459589-1-benno.lossin@proton.me/">https://lore.kernel.org/rust-for-linux/20240717221133.459589-1-benno.lossin@proton.me/</a> |
| GPUVM       | <b>82b78182eacf rust: drm: add base GPUVM immediate mode abstraction</b><br><b>e6303f323b1a drm: manager to keep track of GPUs VA mappings</b><br><a href="https://cdn.kernel.org/doc/html/latest/gpu/drm-mm.html#id38">https://cdn.kernel.org/doc/html/latest/gpu/drm-mm.html#id38</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |





**Thank you!**

**[deborah.brouwer@collabora.com](mailto:deborah.brouwer@collabora.com)**



COLLABORA

**Open First**