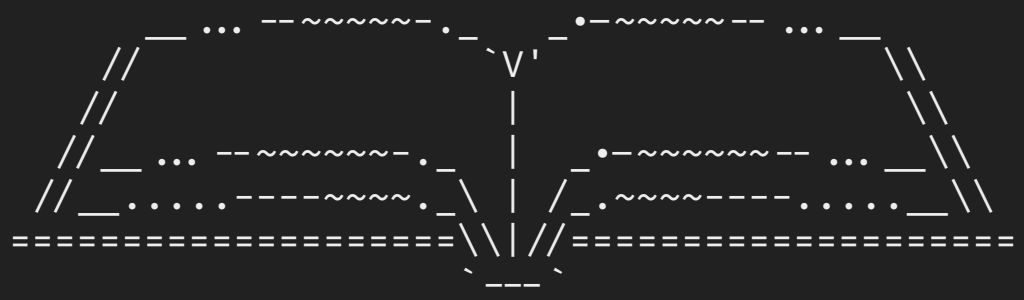
Ninjastorms:

# Memory Management

Marcel Garus, Rohan Sawahn, Jonas Wanke

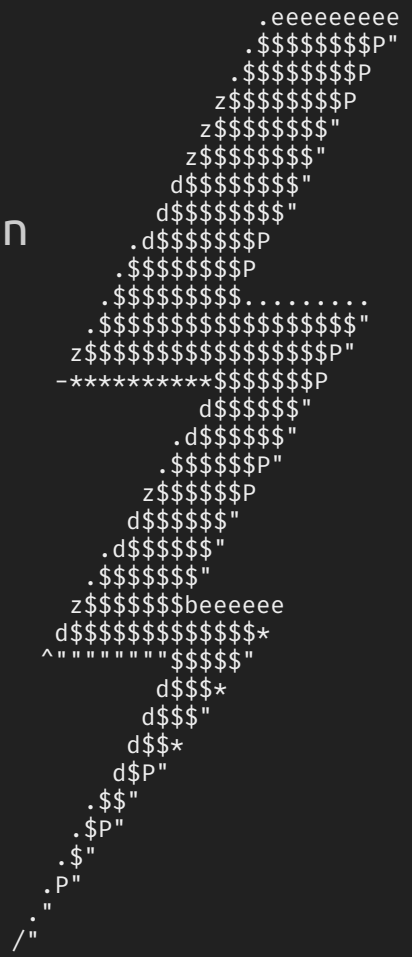
# Was bisher geschah …

# 1. MMU verstehen

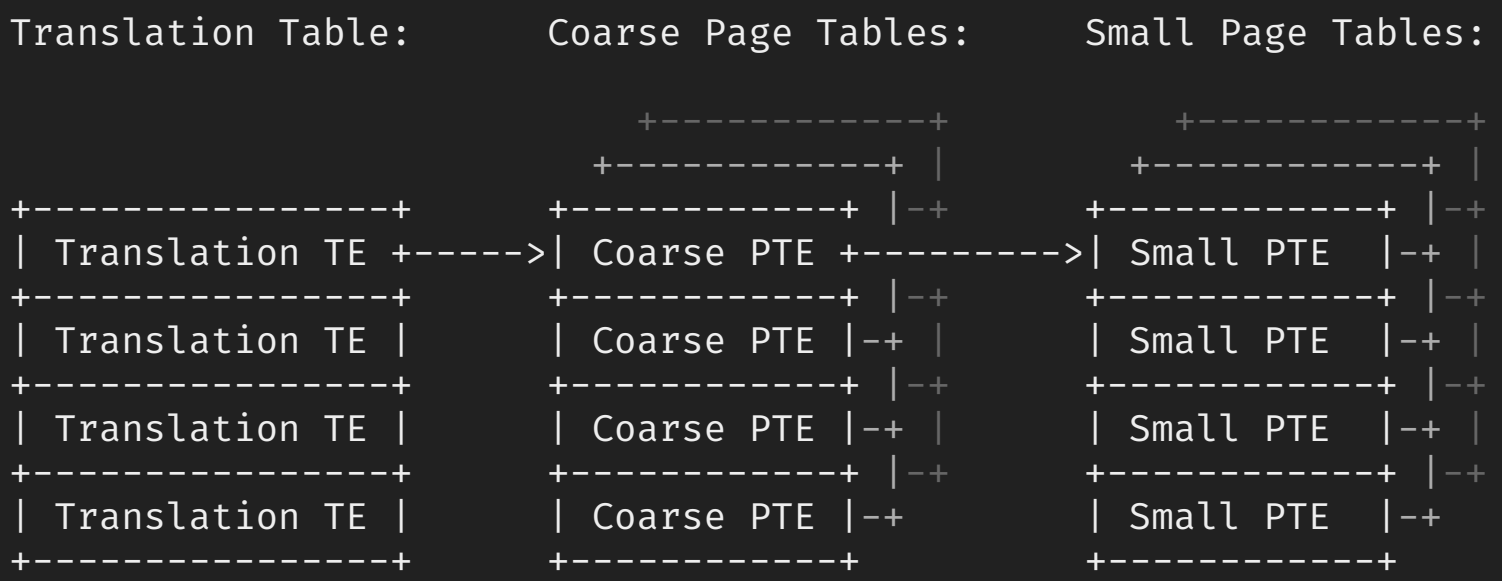


# 2. Interrupt Handling

Wurde dann vom Kernel-Team übernommen

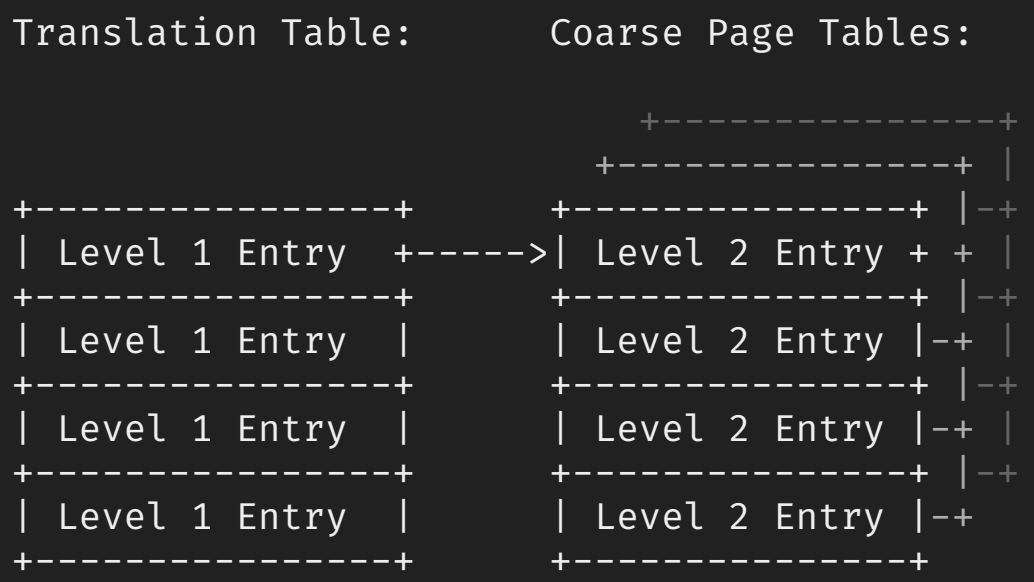


# 3. Pagetables implementieren



Irgendwas stimmt hier nicht …

# 4. Pagetables *korrekt* implementieren



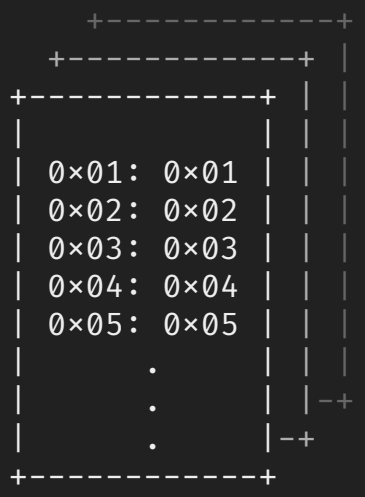
# 4. Pagetables *korrekt* implementieren

```c
typedef struct mem_lvl1_entry_t {

    uint32_t base_address: 22;
    uint8_t const_0: 1;
    uint8_t domain: 4;
    uint8_t const_1: 1;
    uint8_t const_00: 2;
    uint8_t descriptor: 2;

} mem_lvl1_entry_t;
```

```c
// Memory layout:
// | 31        10 | 9 | 8    5 | 4 | 3 2 | 1      0 |
// | Base address | 0 | Domain | 1 |  00 | Validity |

#define LVL1_ENTRY_DEFAULT 0b00000000000000000000000000010000
#define LVL1_ENTRY_CLEAR(entry) (entry = UINT32_ENDIANNESS_SWAP(LVL1_ENTRY_DEFAULT))

#define LVL1_ENTRY_GET_BASE_ADDRESS(entry) ENTRY_GET(entry, 10, 22)
#define LVL1_ENTRY_SET_BASE_ADDRESS(entry, value) ENTRY_SET(entry, value, 10, 22)

#define LVL1_ENTRY_GET_DOMAIN(entry) ENTRY_GET(entry, 5, 4)
#define LVL1_ENTRY_SET_DOMAIN(entry, value) ENTRY_SET(entry, value, 5, 4)
```

# 5. Identity-Mapping für den Kernel

- Bereich des Speichers, in dem
  Virtuelle Adresse = Physikalische Adresse
- Wird für den Code, der die MMU aktiviert, benötigt



# 6. MMU aktivieren

This is where things stop working


```
// Actually enable the MMU.
asm (
  "MRC p15, 0, R1, c1, C0, 0\n" // Read control register
  "ORR R1, #0x1\n"              // Sets M bit from read Control Register to 1
  "MCR p15, 0,R1,C1, C0,0\n"    // Write control register and enables MMU
  "NOP\n"
```

# What’s next?

- Page Tables zum Laufen kriegen
- Demand Paging
  - Free-Page List beim Startup initialisieren
  - Pagefaults implementieren
- Access Control implementieren
- Shared Memory

# Bildquellen

- https://www.asciiart.eu
- https://i.imgur.com/nLnX7BX.jpg