

Dateisystemzugriff: Datei im FAT32
Root-Directory auslesen

Datenträgerinhalt

- MBR
 - ...
 - Partitionstabelle
 - ...
- ...
- erste Partition
 - FAT32 Dateisystem
 - Metadaten
 - FAT -> u32 Liste; Einträge stehen für Cluster in den
 - Nutzdaten
- ...

Modellierung

- BlockDevice (Trait)
 - abstrahiert Datenzugriff
 - read_blocks(..), number_of_blocks(..), block_size(..)
- MbrDeviceDriver
 - Zugriff: BlockDevice
 - liefert: erste Partition
- Partition
 - ist: BlockDevice
 - hat: Dateisystemtyp
- Fat32DeviceDriver
 - Zugriff: BlockDevice
 - liefert: Datei (Vec)

Fat32DeviceDriver Dateizugriff

1. Metadaten liefern: erster Cluster vom Root-Directory
2. Einträge dort enthalten: `is_file`, `name_extension`, `first_cluster`
3. Damit: FAT Clusterkette durchlaufen und
4. entsprechende Cluster in den Nutzdaten konkatenieren (Vec)

Verwendung Codebeispiel

```
let mbr_device_driver = MbrDeviceDriver::new(&block_device);
let partition = mbr_device_driver.get_first_partition();
if partition.get_partition_type() != 0x0B {
    panic!("not FAT32");
}
let fat32_device_driver = Fat32DeviceDriver::new(partition);
let file_vec = fat32_device_driver.read_file_to_vec("tst.txt");
if file_vec.is_some() {
    let file = String::from_utf8(file_vec.unwrap()).unwrap();
    println!("{:?}", file);
} else {
    println!("file not found");
}
```