



# Implementierung eines Dateisystems für Java-basierte eingebettete Systeme

(Zwischenstand Bachelorarbeit)

Dresden, 2012



# Gliederung

- Aufgabenstellung
- Motivation
- Hardwareumgebung
- Anforderungen
- Entwurf
- Auswahl des Dateisystems
- Zusammenfassung / Ausblick

# Aufgabenstellung

- SHAP – Java Bytecode ausführender Prozessor
- SATA-Controller
- Festplatte
- Gesucht: Dateisystemtreiber

# Motivation

- Wenig Speicher (RAM: 1 MiB)
- Autonomer Agent ist nicht immer im Netzwerk
- → persistente Datenspeicherung
- → Festplatte und Dateisystem

# Umgebung und Einschränkungen

- SHAP auf einem FPGA
- 1 MiB RAM für Programm und Daten
- Maximal 16.000 Elemente im Array
- `byte[]` nimmt selben Speicherplatz ein wie `int[]`
- Speichereffizienter SATA-Controller müsste in `int[]` schreiben
- `java.io.File` und Anhang bevorzugen `byte[]` für Lese- und Schreiboperationen
- Verarbeitungsbandbreite beim Umkopieren von `int[]` nach `byte[]`: 2 MB/s

# Anforderungen an das Dateisystem

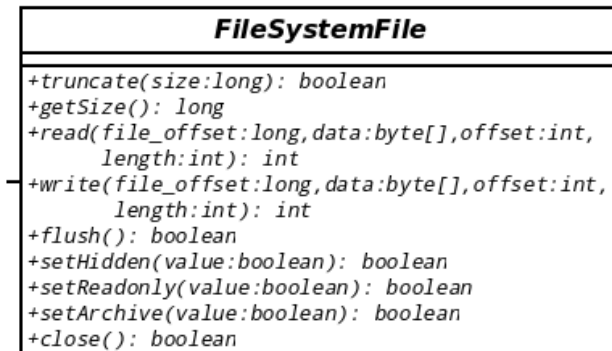
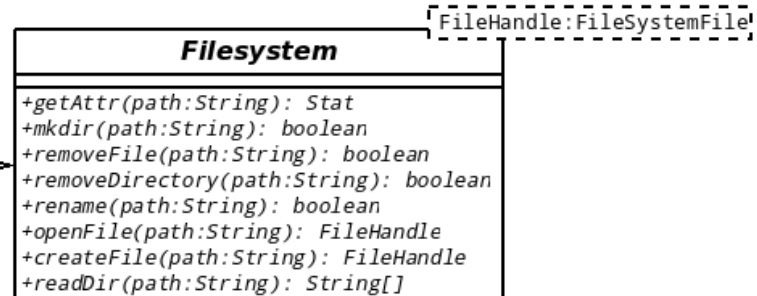
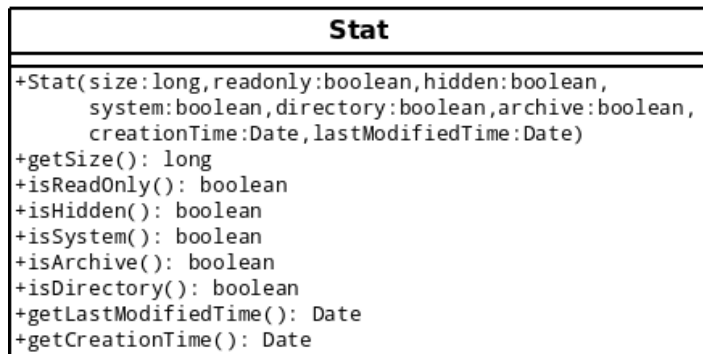
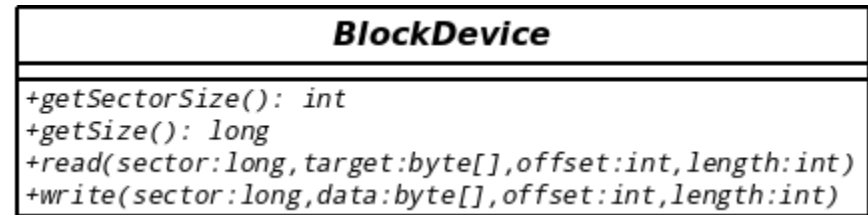
- Sollte mit handelsüblichen Festplatten umgehen
- Festplatte sollte von normalem PC aus schreib- und lesbar sein
- Ziel: `java.io.File` und Hilfsklassen mit selbem Interface ansprechbar
- → PC-Komfort auf einem Embedded System

# Das java.io.File-Interface

- Benutzerorientiert
- Benötigt „eigentlich“ Betriebssystem
- → Dateisystemunabhängige Zwischenschicht
- FUSE-Interface erfüllt diese Anforderung

# Schichten

- Partitionsmanager
- BlockDevice
- FUSE
- Nutzerschicht





## Anmerkungen zum Entwurf

- Partitionsmanager wird initialisiert
- Aus einer Partition wird das Dateisystem erstellt
- Optional: Darüberschalten eines Laufwerksbuchstaben-Wrappers oder VFS
- Dieses Dateisystem wird File als / zugewiesen
- Anschließend kann man `ite.java.io.File` ganz normal benutzen

# Auswahl des Dateisystems

- **Kriterien:**
  - Operiert auf Festplatten
  - Geringe Komplexität
  - Beachtung der Einschränkungen der Dateisystem- und Dateigröße
  - Dateirechte
  - Verbreitung

Dateisystem	Verbreitung	Komplexität	Dateirechte
HFS	MAC	mittel	ja
NTFS	Win, Linux, (MAC)	hoch	ja
ext2	(Win), Linux, (MAC)	gering	ja
FAT	Win, Linux, MAC	gering	nein

## Details zu FAT

- FAT = File Allocation Table
- Feld von „Next“-Zeigern einer verketteten Liste
- Einfach verkettete Cluster
- Sonderfälle: Frei, Beschädigter Cluster, End of File, End of Clusterchain
- Ein Cluster besteht aus einem oder mehreren Sektoren: Sektorgrößen von 512 Byte bis 32 KiB
- Aufbau: MBR/Reserved, FATs, [Wurzelverzeichnis mit fester Länge], Daten

# Dateien und Ordner in FAT

- Verzeichnis: Liste von 32 Byte großen Einträgen
- Jedem echten Eintrag gehen die LFN-Einträge voraus
- Jeder Eintrag enthält Dateinamenteile und Attribute
- Jeder Datei- und Verzeichniseintrag hat Dateigröße und Zeiger auf den ersten Cluster
- Datei/Verzeichnis ist verkettete Liste von Clustern
- Verzeichnisse haben Einträge „.“ und „..“

# Pro - Contra

- **Negativ**

- Clustergrößen 32 KiB ↔ SHAP-Arrays
- Vielfalt der FAT-Implementierungen
- Komplexität aufeinander aufbauender Erweiterungen und deren Abwärtskompatibilität

- **Positiv**

- Gute Dokumentation von Microsoft
- Leicht verständliches Prinzip von FAT
- Gute Dateisystemtools für Testumgebung
- Gute Abstraktionsmöglichkeit der Dateisystemproblematik als Klassen

# Fragen

- Fragen
- Anregungen
- Vorschläge

# Quellen

- Filesystem in Userspace.  
<http://fuse.sourceforge.net/>
- Microsoft EFI FAT32 File System Specification.  
<http://msdn.microsoft.com/en-us/windows/hardware/gg463080.aspx>
- Kapitel Dateisysteme. In: TANENBAUM, Andrew S.: Moderne Betriebssysteme. Pearson Studium, 2002, S. 407ff



**»Wissen schafft Brücken.«**



Dateisystem	Windows	Linux	MAC
FAT/VFAT	x	x	x
HFS/HFS+			x
ext2	(x)	x	(x)
ext3/ext4		x	(x)
NTFS	x	x	(x)

Tabelle 3.1: Dateisystemunterstützung etablierter Betriebssysteme

Eigenschaft	FAT/VFAT[Tan02]	NTFS[Sta03]	ext2[ext12]
Maximale Dateisystemgröße	4 TiB	16 EiB	4 TiB
Maximale Dateigröße	2 GiB	16 EiB	2 GiB
Journaling	nicht vorhanden	vorhanden	nicht vorhanden
Dateirechte	nicht vorhanden	vorhanden	vorhanden
Implementierungskomplexität in Codezeilen	7190	29587	9561

Tabelle 3.2: Dateisysteme

Dateisystem	FAT-Größe in Bits	Maximale Clustergröße	Maximale Dateisystemgröße
FAT12	12	512 B	2 MiB
FAT12 revidiert	12	4 KiB	16 MiB
FAT16	16	32 KiB	2 GiB
FAT32	32	32 KiB	2 TiB

Tabelle 2.1: Dateisystemgrößen von FAT