

---

## 6 Übungsaufgabe #6: Entwicklung eines Dateisystems

Im Rahmen von Aufgabe 6 soll ein Dateisystem im Linux-Kern entwickelt werden.

### 6.1 Grundlagen

Diese Aufgabe umfasst die Recherche notwendiger Techniken und Verfahren zur Implementierung eines Dateisystems. Einen ersten Einstieg können hierbei die Folien zu Systemprogrammierung II, letztes Kapitel([https://www4.cs.fau.de/Lehre/WS16/V\\_SP2/Vorlesung/fohlen.shtml](https://www4.cs.fau.de/Lehre/WS16/V_SP2/Vorlesung/fohlen.shtml)) geben. Jedoch ist das hinzuziehen weiterer Ressourcen, beispielsweise andere im Linux-Kernel vorhandene Dateisysteme und deren Dokumentation ratsam.

### 6.2 Funktionsumfang

Es soll ein Dateisystem im Kernel implementiert werden, das auf gegebenen Block-Devices das permanente Ablegen von Dateien in einer üblichen hierarchischen Verzeichnisstruktur ermöglicht. *Empfohlenerweise* sollte das vorgestellte Dateisystemlayout zusammen mit den gegebenen Hilfestellungen verwendet werden.

**Mindestanforderungen ans Dateisystem:** Das Dateisystem muss in der Lage sein

- Dateien zu lesen und zu schreiben
- übliche Dateinamen bis zu einer Länge von mindestens 63 Zeichen aus einem beliebigen Zeichensatz (mit den Unix-üblichen Ausnahmen Nullbyte und Schrägstrich) zu verwenden.
- reguläre Dateien von variabler Länge
- Dateirechte (traditionelle 9-bit Unix-Rechte) sowie UID und GID zu verwalten und zu beachten.
- wie gewohnt gemountet zu werden, auch mehrmals, d.h. mehrere Dateisysteminstanzen desselben Typs müssen mountbar sein.
- Verzeichnishierarchien bis zu einer Tiefe von mindestens 10 Verzeichnissen zu verwalten. Pro Verzeichnis sollen mindestens 100 Objekte (Dateien, Verzeichnisse, spezielle Dateien) ablegbar sein. Die Gesamtzahl möglicher Objekte im Dateisystem soll nicht unter 1000 betragen.
- Daten zwischen Rechnern mit unterschiedlichen Wortbreiten und Bytereihenfolgen auszutauschen.
- im Mehrbenutzerbetrieb fehlerfrei zu funktionieren

### 6.3 Hilfestellungen

Alle wichtigen Ressourcen sind unter

`/proj/i4passt/fs/ws18`

abrufbar. Darunter findet sich

- ein Code-Skeleton das es erlaubt Dateisysteme zu mounten
- eine Spezifikation des vorgestellten Dateisystems
- ein Programm zum Erstellen von Images (`mkfs.passt`)
- ein Image das der Spezifikation folgt zum Testen

### 6.4 Weitere Hinweise

Die Kodierrichtlinien des Linux-Kernels müssen eingehalten werden!

Im Kernel gibt es bereits diverse implementierte Dateisysteme. Diese dürfen als Vorlage und Inspiration zum eigenen Dateisystem gesehen werden. Jedoch ist der Funktionsumfang oft weit größer und berücksichtigt deutlich mehr Randfälle und spezielle Situationen als für diese Aufgabe gefordert ist. Im Zweifelsfall ist eine einfache und übersichtliche Implementierung zu bevorzugen.

---

## Aufgaben:

- Lesen der Dokumentation, um die für das Dateisystem notwendigen Teile des Linux-Kerns zu verstehen und benutzen zu können.
- Dokumentation des Dateisystems
- Implementierung des Dateisystems
- Testen des Dateisystems nach den gegebenen Anforderungen und ggf. mittels geeigneter Werkzeuge, z.B. `bonnie`, `dd` o.Ä.

## 6.5 Abgabe: am 29.01.2018