

## 1 FUSE-Dateisystem (210 Punkte)

In den Materialien finden Sie ein vordefiniertes Programmgerüst für ein FUSE-Dateisystem; es enthält einige wichtige Funktionen, die allerdings noch von Ihnen implementiert werden müssen. Das Dateisystem lässt sich mit `make` kompilieren und anschließend mit `./dummyfs $mnt` ausführen, wobei `$mnt` für ein beliebiges Verzeichnis steht, das bereits existieren muss.

Informieren Sie sich über die Bedeutung der einzelnen Funktionen mit Hilfe der entsprechenden Manpages und der FUSE-Dokumentation. Die FUSE-API-Dokumentation und weitere Informationen finden Sie unter:

<https://github.com/libfuse/libfuse>

Implementieren Sie die Funktionen, die notwendig sind, um das checkpoint-Programm aus der vorherigen Übung fehlerfrei ausführen zu können. Der komplette Dateisysteminhalt soll sich dabei im Arbeitsspeicher befinden und nach dem Unmounten verloren gehen.

Beschränken Sie sich auf das Unterstützen einer einzigen Datei (`matrix.out`) mit einer maximalen Größe von 5 MiB; das Anlegen, Schreiben, Lesen etc. von anderen Dateien und Verzeichnissen soll dabei einen entsprechenden Fehlerwert zurückliefern. Sie können hierfür der Einfachheit halber statische Datenstrukturen nutzen. (Tipp: Sie können einfach einen Puffer mit 5 MiB Größe allozieren, dessen Größe sich über die die gesamte Programmlaufzeit nicht ändert.)

Denken Sie dabei an die spätere Erweiterbarkeit Ihres Dateisystems, da Sie auf den folgenden Aufgabenblättern das Verhalten Ihres Dateisystems erweitern werden. Im nächsten Schritt soll die Funktionalität des RAM-Dateisystems erweitert werden. Anschließend werden Sie ein persistentes Dateisystem entwerfen und implementieren.

Vergleichen Sie die Werte für den Durchsatz sowie die IOPS mit dem Programm aus der letzten Übung; betrachten Sie dabei jeweils die Werte des Originalprogramms, des Originalprogramms mit `fsync` und Ihrer optimierten Variante. Was fällt Ihnen auf? Welchen Einfluss hat die Benutzung vieler kleiner Operationen und von `fsync` auf die Leistung Ihres Dateisystems?

Stellen Sie sicher, dass Sie den Mountpunkt Ihres Dateisystems betreten (`cd $mnt`) und dort `ls -l` korrekt ausführen können. Außerdem soll es möglich sein, den angelegten Checkpoint wieder zu löschen (`rm`) und seine Größe zu ändern (`truncate`).

## 2 Struktur der Dateisystemoperationen (30 Punkte)

Beschreiben Sie ausführlich den internen Arbeitsablauf Ihres Dateisystems beim Erzeugen, Löschen und Auflisten einer Datei. Benennen Sie dabei jeweils chronologisch alle aufgerufenen Dateisystemoperationen und begründen Sie deren Notwendigkeit.

Aktivieren Sie dafür bei Ihrem FUSE-Dateisystem die Debugausgabe mit `-d`; es kann außerdem sinnvoll sein das Multithreading mit `-s` zu deaktivieren.

## **Abgabe**

Erstellen Sie ein Verzeichnis mit ihrem C-Programm `dummyfs.c`, dem entsprechenden Makefile und der Datei `antworten.txt`. Packen Sie ein komprimiertes Archiv aus dem sauberen Verzeichnis (ohne Binärdateien).

Senden Sie das Archiv an `hea-abgabe@wr.informatik.uni-hamburg.de`.