

In dieser Übung werden Sie Ihr implementiertes FUSE-Dateisystem weiterentwickeln.

## 1 Memory-FUSE-Dateisystem (360 Punkte)

Wir wollen uns nun nicht mehr auf das Unterstützen einer einzelnen Datei beschränken, sondern eine dynamische Dateisystemstruktur ermöglichen.

Erweitern Sie ihr Dateisystem so, dass es möglich ist, beliebige Dateien und Verzeichnisse anzulegen; dabei soll es auch möglich sein, Unterverzeichnisse beliebiger Schachtelungstiefe anzulegen. Die Anzahl an Unterobjekten in einem Verzeichnis soll nicht beschränkt sein.

Das Dateisystem soll allerdings die Speicherbelegung auf maximal 4 GiB beschränken; dazu zählen sowohl die eigentlichen Daten als auch alle Metadaten. Eine einzelne Datei soll dabei eine Größe von maximal 10 MiB erreichen können; der Speicherplatz soll nicht zu Beginn komplett allokiert werden, sondern nur bei Bedarf anwachsen (Tipp: `realloc`).

Dazu dürfen Sie auf bereits fertige Datenstrukturen (Bäume, Hashtabellen etc.) aus anderen Bibliotheken wie zum Beispiel der GLib<sup>1</sup> zurückgreifen. Um den parallelen Zugriff zu gewährleisten, muss außerdem auf einen funktionierenden Locking-Mechanismus geachtet werden.

Zum 04.05.2016 soll ein Designdokument (2-3 Seiten) abgegeben werden, in dem Sie dokumentieren, wie Ihr Dateisystem grundlegend strukturiert ist. Eine Grafik zur internen Datenstruktur sollte verdeutlichen, wie Ihr Code aufgebaut ist. Bitte erläutern Sie Ihre Entscheidungen!

## 2 Leistungsmessung (90 Punkte)

Analysieren Sie ihr Dateisystem mittels des in den Materialien befindlichen Benchmarks. Erstellen Sie geeignete Diagramme und notieren Sie Ihre Schlussfolgerungen.

Der Benchmark lässt sich wie folgt aufrufen, wobei `$mnt` für den Einhängpunkt Ihres Dateisystems steht:

```
$ ./metadata --path=$mnt --iterations=3
```

Weitere Optionen können Sie sich mit `--help` anzeigen lassen. Messen Sie die Leistung Ihrer Implementierung mit 1–12 Threads (`--threads`) sowohl für thread-lokale Verzeichnisse als auch für ein geteiltes Verzeichnis (`--shared`). Wählen Sie dafür jeweils eine geeignete Anzahl an Objekten (`--objects`) damit der Benchmark einige Minuten läuft. (Hinweis: Die Anzahl der Objekte wird pro Thread angegeben und ist standardmäßig auf 1.000.000 gesetzt.)

## Abgabe

Erstellen Sie ein Verzeichnis mit ihrem C-Programm `memoryfs.c`, dem dazugehörigen Makefile und den Dateien `design.pdf` und `benchmark.pdf`. Das Designdokument soll dabei sowohl

---

<sup>1</sup><https://developer.gnome.org/glib/>

nach einer Woche als auch zum eigentlichen Abgabetermin (mit eventuellen Verbesserungen/Anpassungen) abgegeben werden. Packen Sie ein komprimiertes Archiv aus dem sauberen Verzeichnis (ohne Binärdateien).

Senden Sie das Archiv per E-Mail an [hea-abgabe@wr.informatik.uni-hamburg.de](mailto:hea-abgabe@wr.informatik.uni-hamburg.de).