Exercise Sheet 5 for Lecture Parallel Storage Systems

Deadline: 2026-05-31 and 2026-06-14, 23:59

Prof. Dr. Michael Kuhn (michael.kuhn@ovgu.de)
Michael Blesel (michael.blesel@ovgu.de)
Parallel Computing and I/O • Institute for Intelligent Cooperating Systems
Faculty of Computer Science • Otto von Guericke University Magdeburg
https://parcio.ovgu.de

---

1. Memory FUSE File System (480 Points)

We no longer want to limit our file system to only support one file. Instead, we want to allow a dynamic file system structure.

Extend your file system so that it is possible to create arbitrary files and directories. It should be possible to create nested directories with no depth limit. The number of sub-objects in a directory shall be unlimited as well.

The file system shall be limited to a maximum size of 4 GiB. This includes the regular data and the metadata. Individual files shall have a size limit of 10 MiB. The memory shall not be allocated completely at the beginning but grow on demand.
Hint: The `realloc` function allows shrinking and growing allocations.

You are free to use already existing data structures (trees, hash tables, etc.) from libraries such as for example GLib[1]. Keep in mind that the data structures will have to be made persistent on a storage device for the next exercise sheet. To allow for parallel access you also have to think about a working locking mechanism.

Make sure your file system at a minimum works with the following commands:
`touch, rm, ls, mkdir, rm -Rf, read, write, truncate, mv`

2. Memory File System Design (120 Points)

Until the first deadline, you have to submit a design document (2–3 pages), in which you document the basic structure of your file system. There shall be a graphic about the internal structure of your file system that makes clear how your implementation works. Please explain the design decisions you made.

[1] https://docs.gtk.org/glib/

3. Performance Measurements (90 Points)

Analyze your file system with the benchmarks from the materials. To compile it, you will have to load the `glib` module:

```
$ . /opt/spack/pss-2026/env.sh
$ module load glib
```

Create suitable visualizations of the results and document your conclusions.

The following shows how the benchmark can be executed, where `$mnt` stands for the mount point of your file system:

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

Further options can be displayed with `--help`. Measure the performance of your implementation with 1–12 threads (`--threads`) for thread-local directories and also for a shared directory (`--shared`). Choose a suitable amount of objects (`--objects`) so that the benchmark runs for several minutes.
Hint: The object count is given per thread and defaults to 1,000,000.

Submission

We will count your last commit on the `main` branch of your repository before the exercise deadline as your submission. In the root directory of the repository, we expect a `PSS-2026-Exercise-05-Materials` directory with the following contents:

- A file `group.md` with your group members (one per line) in the following format:

  ```
  Erika Musterfrau <erika.musterfrau@example.com>

  Max Mustermann <max.mustermann@example.com>
  ```

- Task 1: The source code for your file system `memoryfs.c` and the corresponding Makefile. If you chose a different language than C, please include all source code and a way to build the application easily. The implementation has to be submitted until the second deadline.
- Task 2: An extensive design document detailing the inner workings of your file system (including at least one architecture graphic) called `design.pdf`. The design document has to be submitted once until the first deadline and then again for the second deadline (with possible improvements/changes).
- Task 3: A document containing your visualized performance measurements and explanations called `benchmark.pdf`. The evaluation has to be submitted until the second deadline.