Operating Systems 198:416
Prof. Srinivas Narayana
AC1771 Advith Chegu
TSC95 Tajvir Chahal

Total number of blocks used when running benchmark:

For simple_test.c: 23 blocks were used in total

For test_cases.c: 23 blocks were used in total

Time to run the benchmark

For test_cases.c:
TEST 1: File create Success
TEST 2: File write Success
TEST 3: File close Success
TEST 4: File read Success
TEST 5: Directory create success
TEST 7: Sub-directory create success
Benchmark completed
The program took **0.003031 seconds** to run.

For simple_test.c:
TEST 1: File create Success
TEST 2: File write Success
TEST 3: File close Success
TEST 4: File read Success
TEST 5: Directory create success
TEST 6: Sub-directory create success
Benchmark completed
The program took **0.001903 seconds** to run.

How the code was implemented:

The method that the code was implemented was how it was recommended to be done in the writeup. First, we began by implementing rufs_mkfs, rufs_init, and rufs_destroy.

Following this we implemented the helper functions before the rest of the rufs functions. We then made the bitmap and inode operation functions. We finished by completing the directory operations and began testing our methods with the benchmarks.

Additional steps to compile the code: N/A

Difficulties encountered:

During the beginning of the project we were initially overwhelmed with the amount of functions that needed to be completed, so it took a period for us to create a plan on how to develop the project. We specifically encountered issues with developing the rufs_create() function and how to deal with subdirectories in the project. Through periods of debugging and planning, we were able to resolve our issues.

Collaboration and References

- *Lecture and Recitation Notes* → used as reference for implementation and logic
- *FUSE library API documentation* → used throughout implementation of the project to refer to when using FUSE API
- *Other FUSE tutorials in project 4 pdf* → similarly used as reference for FUSE API