16:198:519:01 - OPERATING SYS THEORY
Project1 Part1 - Summary Report
Raghvesh Prasad (rp1399)

Overview

Added a new dummy custom syscall *app_helper* to the *mmap.c*

- app_helper, reads the buffer from user mode using ***copy_from_user()*** i.e. copies the content of user buffer to the kernel buffer.
- app_helper, after reading the buffer then sets the entire buffer content to 1.
- app_helper, finally copies the updated kernel buffer to the user buffer using ***copy_to_user***.

Created a custom user level tester which invokes the *app_helper*

- The tester, takes two inputs from the command line i.e. buffer_size and No. of Iterations.
- The tester, then creates a new buffer of the passed size and initializes it to 4.
- The tester, runs a loop for the no. of iterations passed to it invoking the ***app_helper***.
- The tester, checks if the content of the buffer is updated to 1.
- Prints out the total latency and Average latency per call in nano seconds.

Git Repo URL: CS519_Project1

Experiment Steps

- Worked with kernel version linux-5.15.0.
- Installed all the scripts, compiled the kernel and restarted the OS.
- ***app_helper*** added to mmap.c, entry added to syscall_64.tbl and function signature added to syscalls.h.
- Tester run for buffer sizes : (256,512,1024,2048,4096,8192,16384,2097152) and iterations : (10,50,100,500,1000
- Tester called via the updated run_tests.sh which tackles the above two argument passing.
- diff between old and new kernel files recorded.
- Output presented as a graph in subsequent sections.

Results and Observations

The syscall was invoked succesfully, The buffer was read, updated and sent back to the user-mode tester without any hiccups.

```
[ 3268.915996] app_helper: System call invoked successfully
[ 3268.916000] app_helper: System call invoked successfully
[ 3268.916004] app_helper: System call invoked successfully
[ 3268.916008] app_helper: System call invoked successfully
[ 3268.916012] app_helper: System call invoked successfully
[ 3268.916016] app_helper: System call invoked successfully
[ 3268.916020] app_helper: System call invoked successfully
[ 3268.916025] app_helper: System call invoked successfully
[ 3268.916029] app_helper: System call invoked successfully
[ 3268.916033] app_helper: System call invoked successfully
raghvesh@node-0:~$
```

Figure 1: Successfull call to app_helper; dmesg entries

Total latency over entire set of iterations and average latency per call were recorded.

Total latency over different buffer sizes for varying iterations:

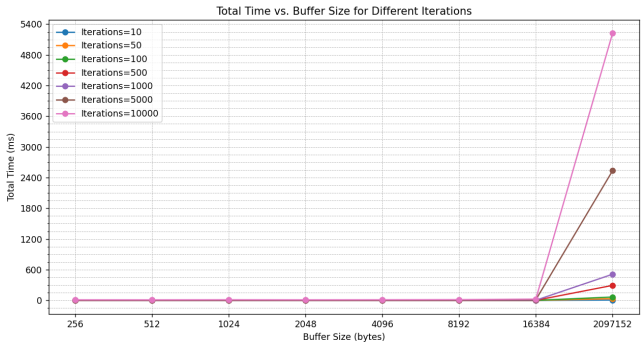

Figure 2: Total Latency with varying Buffer Sizes and Iterations

For better comparison, Displaying the same results on log scale:

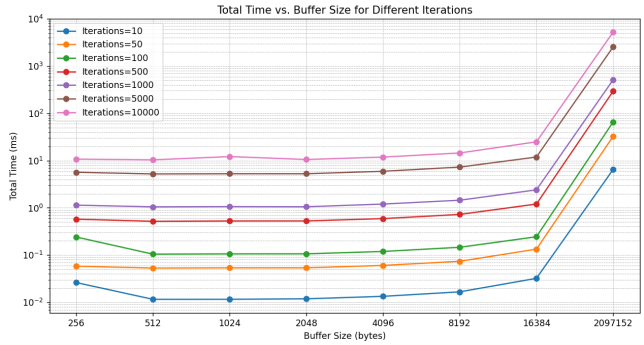

Figure 3: Total Latency with varying Buffer Sizes and Iterations

Average latency over different buffer sizes for varying iterations:

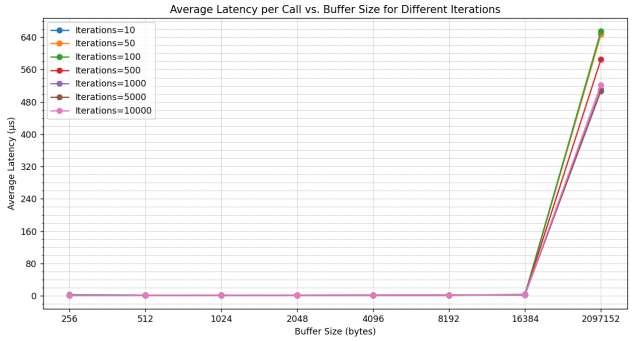

Figure 4: Average Latency with varying Buffer Sizes and Iterations

For better comparison, Displaying the same results on log scale:

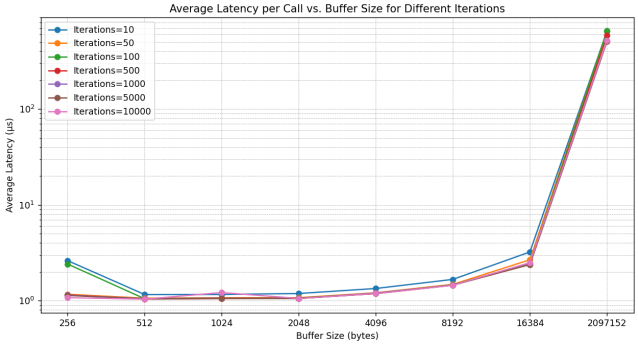

Figure 5: Average Latency with varying Buffer Sizes and Iterations

Observations The total latency increases sharply as the no. of iterations and buffer sizes increase without any great variation.

Total latency continously climbed as iterations and buffer sizes increased.

The average latency reduces as the syscall is invoked again and again hinting at caching of the Context and syscall by the CPU.

Variations There were not many variations observed in the runs.
However, the only small variation that is observed is with the average latency observed during the first call for 256 buffer size but that eventually reduces as the iterations increase.
Which may be due to the first time kernel call resulting in context being cached.