

Course Work 2 Report

Status:

I believe my solution has achieved tier three of this project. When running the code using make new & make, it compiles with no warnings or errors. When testing the code with the provided files, the output from test.log is the same as test.out.

Sequential & 2-thread Runtimes:

Tier 1:

```
[2870251m@ssh1 Test]$ time ./strace-analyser test.log > tier1
real    0m0.056s
user    0m0.049s
sys     0m0.003s
```

Tier 2:

```
[2870251m@ssh1 SPCW2]$ time ./strace-analyser test.log 1 > tier2
real    0m0.148s
user    0m0.164s
sys     0m0.068s
[2870251m@ssh1 SPCW2]$ |
```

Runtime with Multiple Threads:

| Threads | 1 | 2 | 4 | 8 |
|-----------------------|----------|----------|----------|----------|
| Execution Time/Run 1 | 0m0.152s | 0m0.171s | 0m0.265s | 0m0.390s |
| Execution Time/Run 2 | 0m0.149s | 0m0.170s | 0m0.229s | 0m0.330s |
| Execution Time/Run 3 | 0m0.154s | 0m0.160s | 0m0.230s | 0m0.301s |
| Median Execution Time | 0m0.152s | 0m0.170s | 0m0.230s | 0m0.330s |

Discussion:

The runtime results shown above shows that having 1 thread has the fastest execution time with a median of 0m0.152s. When we increase the number of threads it makes the execution time slower, with the median of 2 threads being 0m0.170s, the median of 4 threads being 0m0.230s, and the median of 8 threads being 0m0.330s.

This shows that as the number of threads increased the run time gets longer. This is because the program is relatively small and simple and thus introducing more threads complicates the process, hence the longer runtimes.