CS 214: Systems Programming
Fall 2016
Assignment 2: Procs vs Threads (round 0)
Riddhish Pandya & Kendrew Shum
Timetests

In order to test our processes vs threads, we generated multiple large text files of around 10,000 characters and tried to generate anywhere from 5-100 compressed files. We did this to really see how each program runs on asymptotically larger inputs in terms of time so we can see a distinction between the two different implementations (processes vs. threads). We calculated all time measurements using time from command line (“$ time ./compressR_LOLS test.txt 100).

Each of the following tests were run 3 times are these were the average of the times.

Test01:
Input file: 10,000 alphabetic characters.

Number of output files: 5

```
Processes Time:
real  0m0.007s
user  0m0.005s
sys   0m0.011s
Threads Time:
real  0m0.006s
user  0m0.004s
sys   0m0.008s
```

Test02:
Input file: 10,000 alphabetic characters.

Number of output files: 20

```
Processes Time:
real  0m0.007s
user  0m0.001s
sys   0m0.018s
Threads Time:
real  0m0.009s
user  0m0.004s
sys   0m0.012s
```

Test03:
Input file: 10,000 alphabetic characters.

Number of output files: 50

```
Processes Time:
real  0m0.018s
user  0m0.006s
sys   0m0.049s
Threads Time:
real  0m0.012s
user  0m0.000s
sys   0m0.021s
```

Test04:
Input file: 10,000 alphabetic characters.

Number of output files: 100

Processes Time:

```
real	0m0.022s
user	0m0.006s
sys	0m0.068s
```

Threads Time:

```
real	0m0.024s
user	0m0.010s
sys	0m0.030s
```

Conclusion:
Based on the results from the tests that I ran, it is evident that threads run faster than processes. There is clearly better performance from running the thread program. This is probably because context switches are done quicker in threads than in processes, and in larger inputs, these context switches can make a huge difference in runtime.