

# Time Test

CS214 Assignment 2.

Rebecca Rice, Yequ Sun

In order to do multiple tests quickly, we used the following bash command:

```
for((n=0;n<20;n++)); do (time ./compressT/R_LOLS <File name> <part number>) 2>> <log file name> ; done
```

This command runs the program 20 times and logs all the time data into a file.

First test: Both versions compress a very short file (abc.txt 20 bytes) with 2 parts. Both of them gets the job done quickly. The results are very close. It seems the workload is too light to see the difference.

Second test: Both versions compress a big text file (large.txt 84 KB) with 2 parts. This time, the process version is clearly faster.

Third test: Both versions compress the previous file with 10 parts. The process version wins again. But the gap is smaller and they both faster than the last time.

Conclusion: When the workload is heavy, multi-processing has better performance. The context-switches of multi-threading can slow the progress down. When the workload is light, they perform similarly since multi-processing costs more to initialize. Both of them run faster with more threads/processes when under heavy load.