

# Time Tests

I compiled two binaries: `outR` and `outT` representing the process and thread implementations, respectively.

I also created two text files, `100k.txt` and `50.txt` which are text files containing 100,000 and 50 randomly generated characters, respectively.

To record the amount of time elapsed, I am using the UNIX `time` utility provided on OSX.

## Test 1

`time ./outR 100k.txt 1` takes a total of 0.013s to complete

`time ./outT 100k.txt 1` takes a total of 0.007s to complete

## Test 2

`time ./outR 100k.txt 50` takes a total of 0.040s to complete

`time ./outT 100k.txt 50` takes a total of 0.012s to complete

## Test 3

`time ./outR 100k.txt 500` takes a total of 0.288s to complete

`time ./outT 100k.txt 500` takes a total of 0.102s to complete

## Test 4

`time ./outR 50.txt 1` takes a total of 0.009s to complete

`time ./outT 50.txt 1` takes a total of 0.013s to complete

## Test 5

`time ./outR 50.txt 25` takes a total of 0.038s to complete

`time ./outT 50.txt 25` takes a total of 0.027s to complete

## Test 6

`time ./outR 50.txt 50` takes a total of 0.075s to complete

`time ./outT 50.txt 50` takes a total of 0.025s to complete

## Conclusion

After running these tests it appears that using threads is - on average - faster than using worker processes when compressing files.