



**Bilkent University
Engineering Faculty
Department of Computer Engineering**

**CS 342
Project 1
Report**

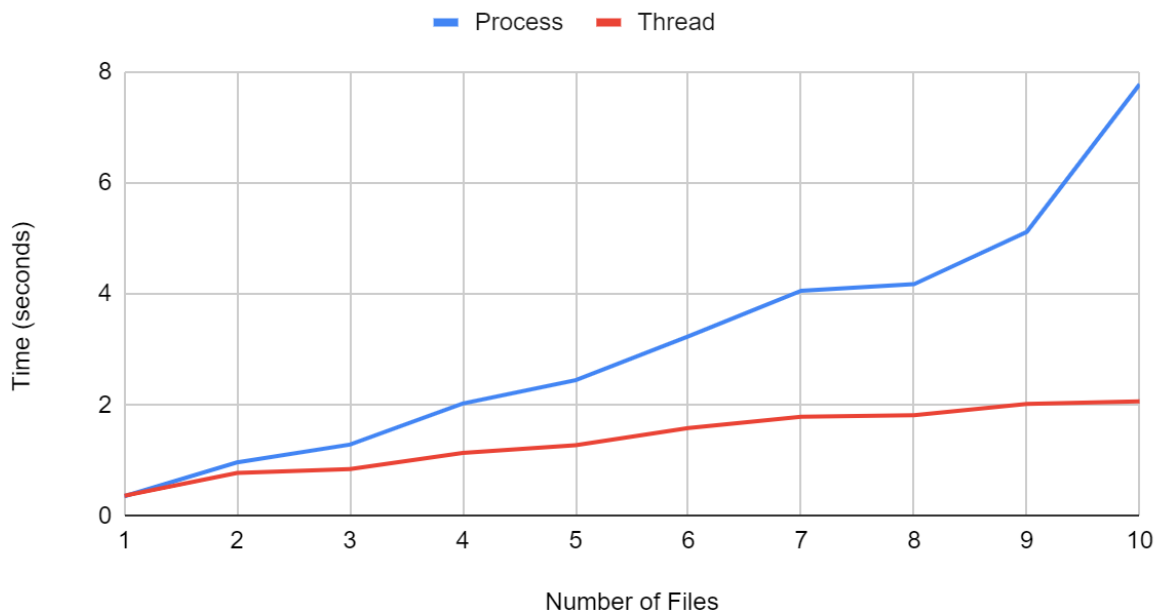
**Aziz Ozan Azizoğlu
21401701**

In order to create files that are filled with random numbers, I implemented a Python script. It's mentioned in the project description that there can be millions of numbers in the files and these could be very large numbers, therefore I decided to choose the range of the values as $[1 - 2^{28}]$ and number of integers that can be in a file as [1M-6M]. The script creates 10 ".txt" files for me to complete my experiments. In the experiments I chose 2^{28} as the largest number that can appear in the files, thus the range function checks. The script I have implemented is given below:

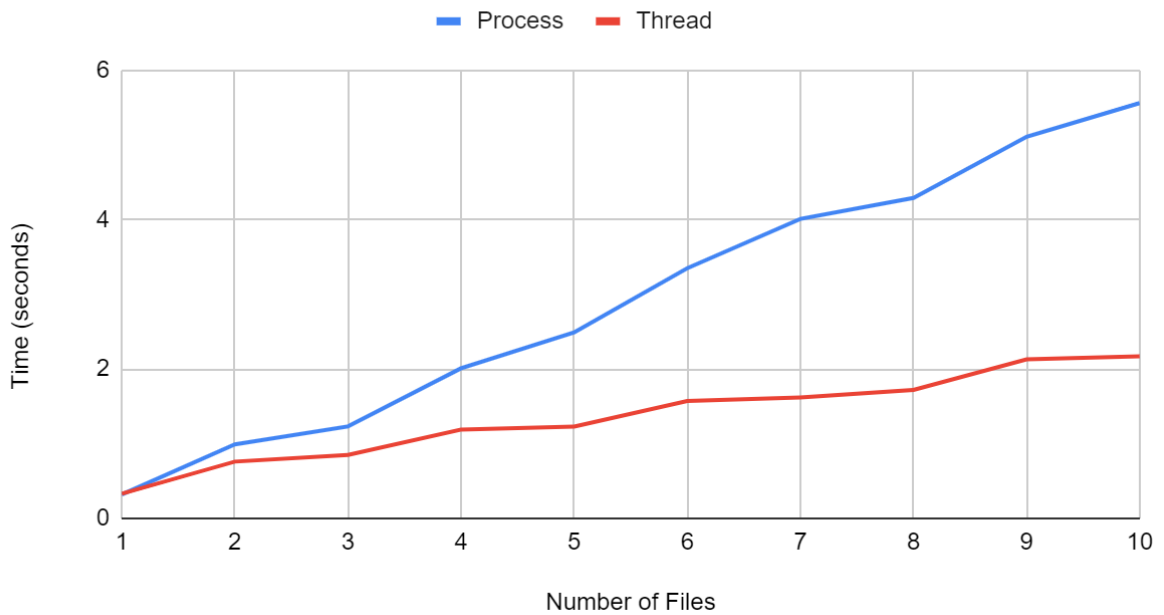
```
from random import seed
from random import random
from random import randint
minimum = 1
maximum = 2 ** 28
rmin = 1_000_000
rmax = 6_000_000
seed(1)
files = ["file1.txt", "file2.txt", "file3.txt", "file4.txt",
"file5.txt", "file6.txt", "file7.txt", "file8.txt", "file9.txt",
"file10.txt"]
for i in range(10):
    f = open(files[i], "w")
    num = randint(rmin, rmax)
    for _ in range(num):
        value = randint(minimum, maximum)
        result = str(value) + "\n"
        f.write(result)
    f.close()
```

Results of the experiments of programs are shown below:

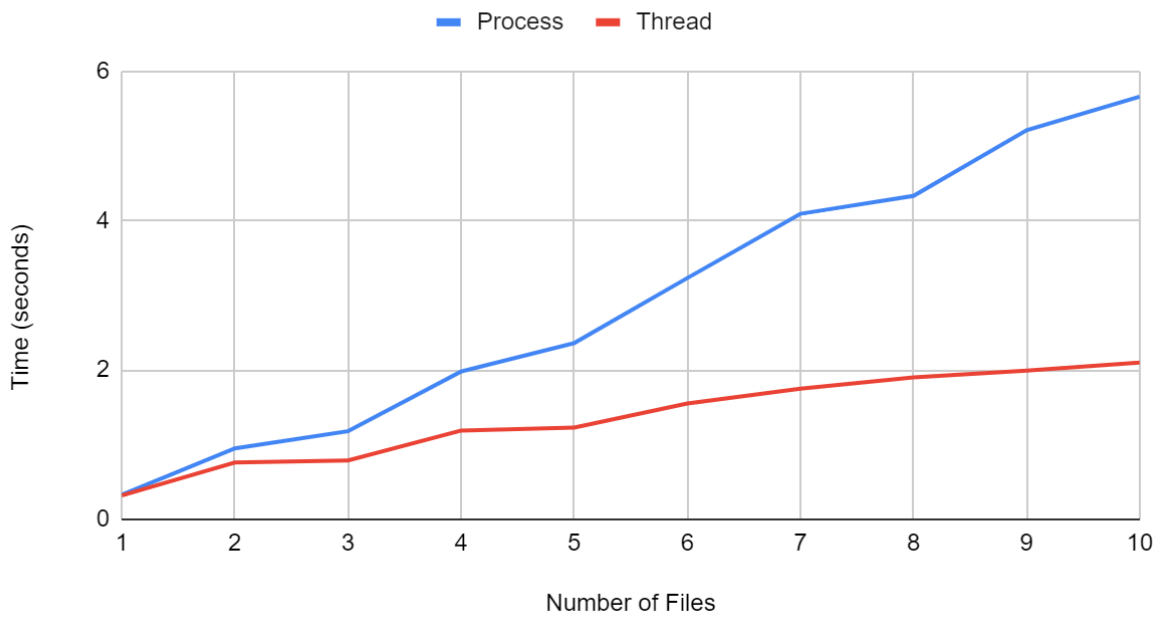
max



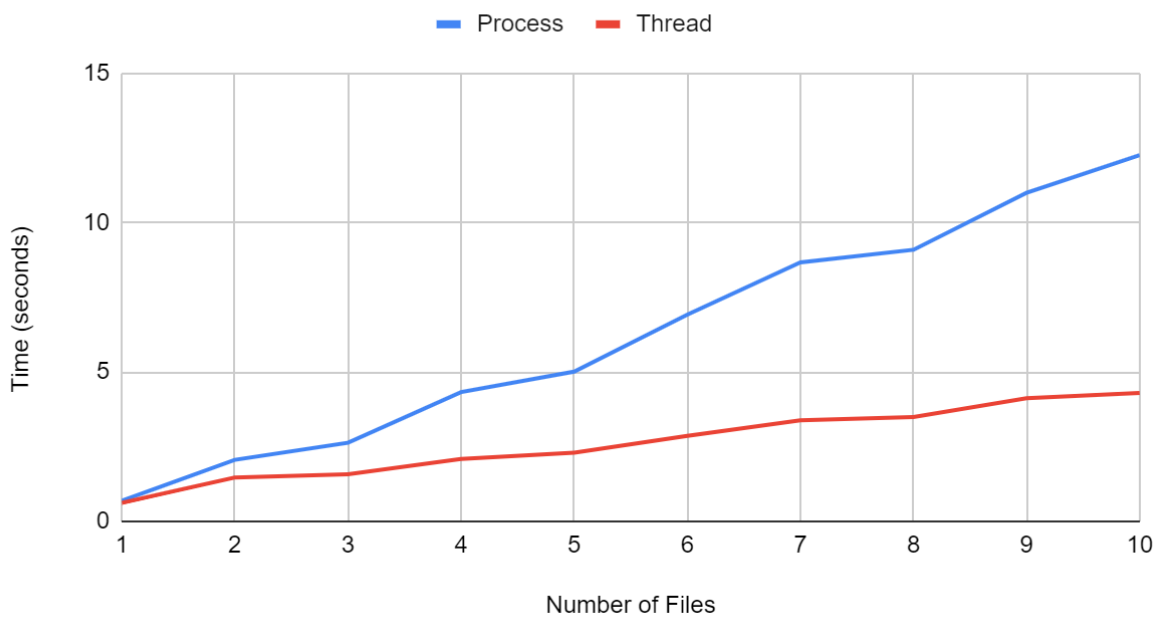
count



avg



range 1 268435456 1000



In conclusion, as we can see from these graphs, threads run much faster than processes. This was the expected result going into the experiments. Since processes do not share memory and instead are created with a copy of the parent process in the creation process, they both take more time to create and there is a need for interprocess communication. Threads on the other hand, do share the memory and thus do not need to communicate via a special method, e.g. pipes, threads can instead access to global variables . Also, there is the penalty for context switching in the case of processes which adds up to the running time.

Test Environment Specs:

Apple M1 chip with 8-core CPU, 8-core GPU, and 16-core Neural Engine

16GB unified memory

OS: Ubuntu 20.04 LTS ARM

VM: UTM