

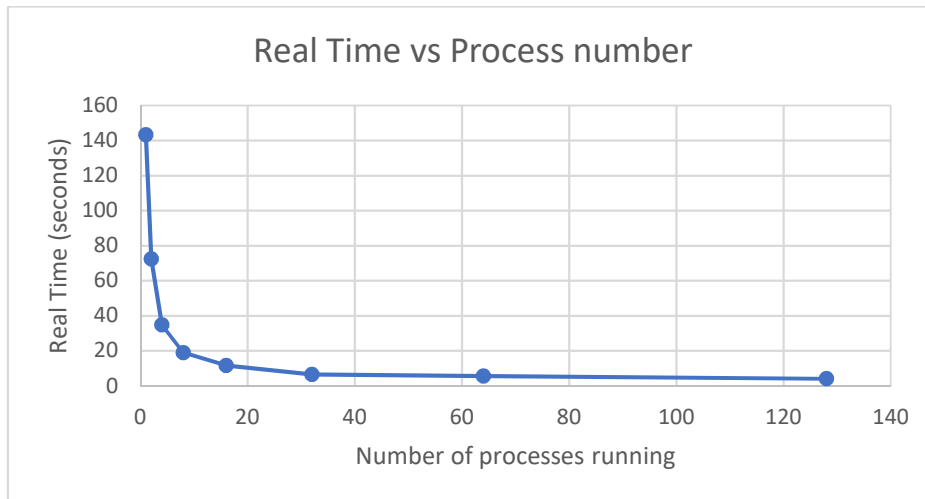
Home Work 2

Gai Ashkenazy – 204459127

Part 1:

The code is included in the zip file

Part 2:



3.

4. We see the improvement in the graph even when using a signal core because we are running several processes in parallel. Instead of waiting for the server's response in each call we gave each process another task while others are waiting for the response. This causes the processes to work in parallel which reduces the time. Also when running to each URL a process the time is still more than 2 minutes because we are doing other time expensive tasks like querying the server to check it, fork, and system calls.

5. checking only the structure of the URL is an action that does not require a system call, so there is no waiting time for the process. This means that the graph will look completely different, because if we will use several processes, which takes time because initiating processes is a system call, the graph will likely increase as we have more processes.

6. Using threads will improve the running time, threads share memory, which makes switching and creating them much faster than processes. Hence the running time will decrease.

Part 3:

A. The code will print numbers between 1-101. We can not know the exact order of the number, only that one of the two '1' will be printed first and one of the '101' will be printed last

We do know that the total amount of numbers that we will print, each x will be printed 2^x times. so the total number of prints will be $\sum_1^{101} 2^x$.

B. Two of following lines will be printed first (not necessarily in this order):

PID PID1: Pending for signal.

PID PID2: Pending for signal.

After that the following lines will be printed:

PID1: Caught User signal from another process.

PID2: Caught User signal from another process.

PID1: Caught Interrupt, exiting now.

the fourth kill command will not affect the output because PID1 is already dead.

And then:

PID2: Caught Interrupt, exiting now.

Part 4:

1. False, when a parent of a process terminates the child process keeps running and they become children of the init process. (init PID is 1).

2. False, processes that are related and both unrelated can communicate through named pipes.

3. False, using t.run will run the tread, so there will not be multiple treads, and using t.start will create a new tread and run it, few treads will run simultaneously.