Tu Tran
CIS 3207
Professor John Fiore
February 7th, 2017

Lab 1 Report

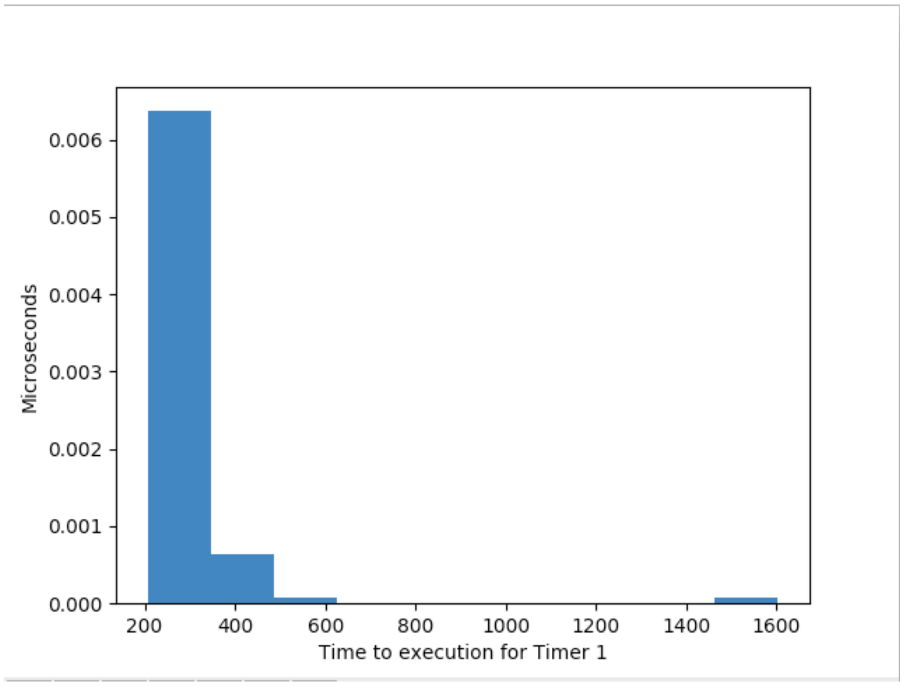
The purpose of this lab is to create an application and two timers to measure its startup time (i.e. the time from forking to execution.

For this lab, here's my steps of execution:

- Write the application program:
    - Open a new file by command-line argument
    - Loop 50 times
    - In each loop
        - Write 10 records, each of which is a sequence of 120 random characters
        - Read randomly from one of 10 records just written and compare to all of the 10 records
    - Close the file and remove it
- Write the two timers:
    - Timer 1:
        - Set up timer before forking.
        - In the parent branch, right before the parent "reaps" the child process, call the timer done and measure the duration by taking the difference between property *tv_usec* of the two *timeval* structures.
    - Timer 2:
        - Set up the first timer before forking for the first time.
        - After forking, in the parent branch of the first fork, launch the second timer before forking the second time.
        - In the parent branch **of the second fork,** call timer done before it reaps both children through *waitpid*.
        - Measure the two durations by taking two differences.
- Write a Python script:
    - Use the *subprocess* module to run both timers from Python for 100 times
    - In each loop, append the result into an array. There are two arrays for two timers as a result
    - Write a function to calculate the mean, mode and median
    - Use *matplotlib* module to draw the histogram

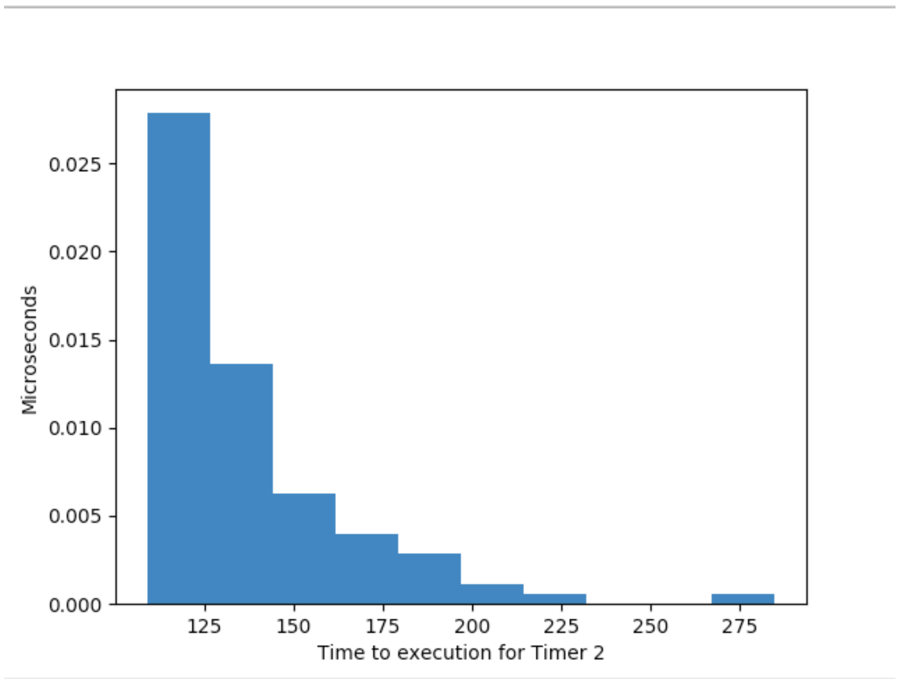
The result is on the next page.

Here’s the histogram for Timer 1:



As we can see, most of the times measured lie in the 200-250 range.

Here’s the histogram for the Timer 2:



In this case, mostly times measured are half of those in Timer 1.

1/ Here's the median, mean and mode for each Timer:

	Timer 1 (microsecond)	Timer 2 (microsecond)
Mean	263.35	137.5
Mode	232.0	123.0
Median	245.0	128.0

2/ Confidence level interval:

Timer 1: 263.35 ± 43 microseconds
Timer 2: 137.5 ± 32 microseconds

3/ The reason it shouldn't always take the same amount of time to perform the action of program startup is that there are multiple process running at the same time in the machine, and at diffent times the OS allocates different amounts of resources from the CPU. In the case the second application runs faster, because the OS already allocates resources for the first instance of application, when the second runs it uses some resources such as text and code.

In conclusion, this project shows us how at different times a program can take different amounts of time to run and what happens if multiple instances program are running at the same time.