

# Embedded and Real Time Systems 2022

## Project 1

Triantafillidis Grigoris 8923

tgrigori@ece.auth.gr

code: [One Drive](#)

For this project the producer-consumer problem was tested using POSIX Threads as the producers and the consumers. The goal was to find the number of consumers in order for the work functions to stay in the queue for the minimum amount of time.

Assuming the queue has a fixed length of 10 spaces, each producer produces 50.000 work functions and places them in the queue. In our case each work function computes the sines for 70 angles. When a work function is placed in the queue, a timestamp is stored. When a consumer gets the work function off the queue, the wait time is calculated and stored in a *times[]* array (wait time in microseconds).

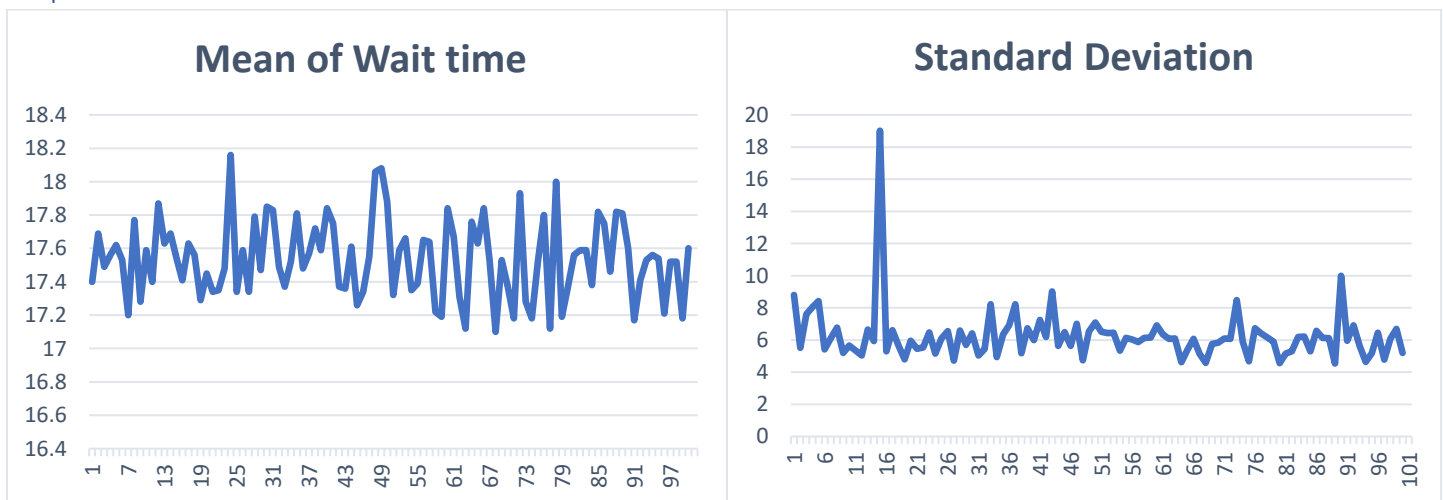
When the total number of work functions is reached ( $numProducers * workFunctionsPerProducer$ ), statistics are calculated from the *times[]* array (range, mean, variance, standard deviation).

The program was tested for consumers from 1 to 100 and 10, 15, 20, 25 producers on PC.

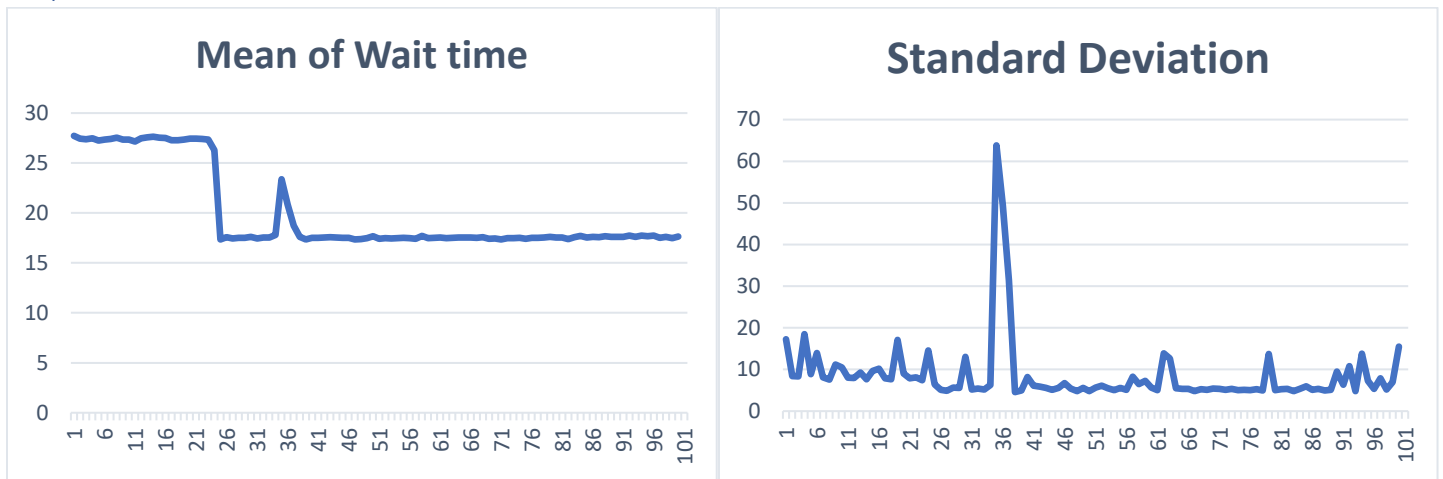
The *x-axis* is **numberOfConsumers** and the *y-axis* is **time in microseconds**.

On intel core i7-9750H 2.6GHz with 12 logical processors

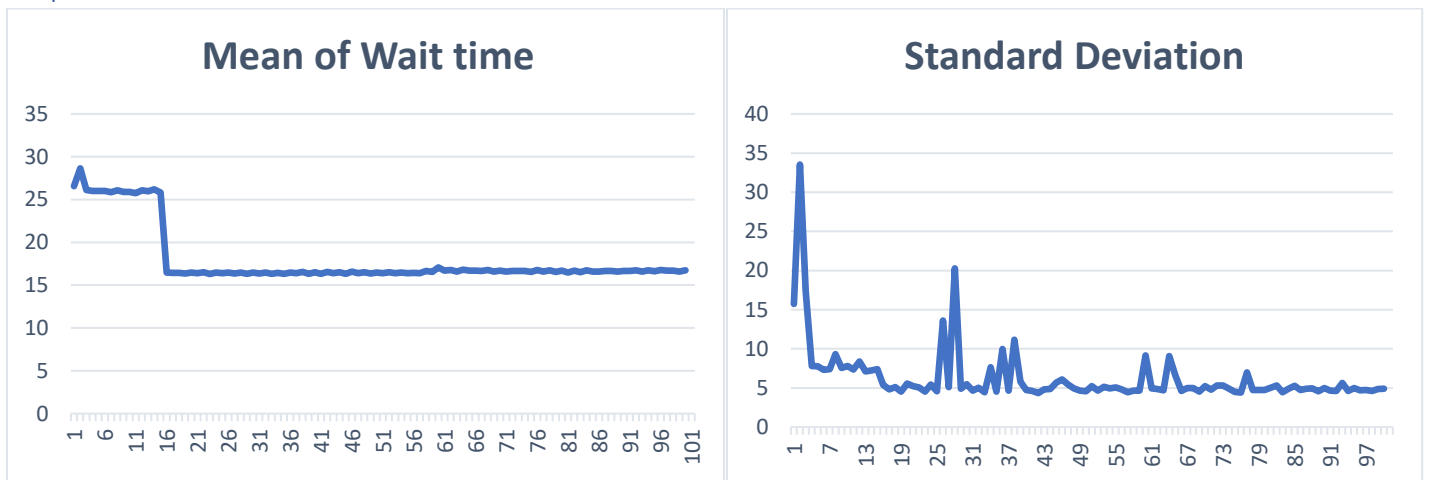
10 producers



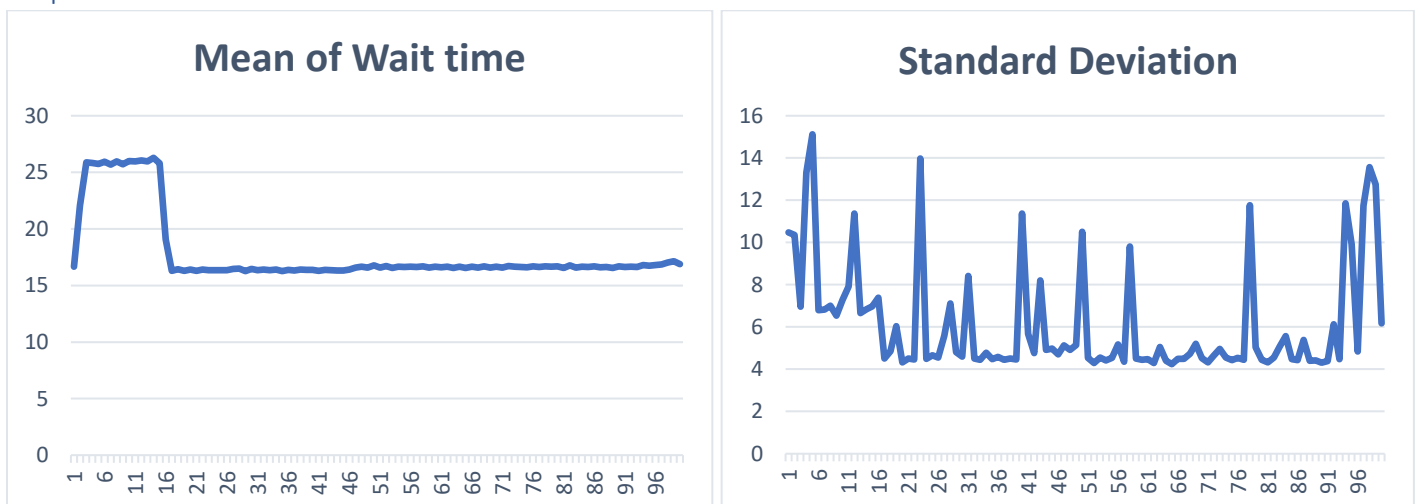
### 15 producers



### 20 producers



### 25 producers



As we can see, for 15, 20 and 25 producers the mean wait times stabilize at 16-20 consumers, and the wait time drops to around 16-17 microseconds. For 10 producers the times are at 17 – 18 microseconds.