

This report summarizes Assignment 5. Both parts were implemented and tested successfully on Linux (VMware Ubuntu, kernel $\geq 4.x$).

In Part A, the parent and child processes communicate via POSIX shared memory using semaphores to synchronize a fixed-size producer-consumer queue of doubles.

In Part B, the same sequence is transmitted through a pipe.

For both programs (lab5_a.c and lab5_b.c), the commands below were used: gcc -O2 -Wall -pthread -o lab5_a lab5_a.c, gcc -O2 -Wall -o lab5_b lab5_b.c, and execution ./lab5_a 6 2.5 or ./lab5_b 6 2.5.

The outputs matched the expected arithmetic sequence (79.7, 82.2, 84.7, 87.2, 89.7, 92.2), confirming correct behavior for both shared-memory and pipe-based IPC implementations.

```
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ touch lab5_a.c
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ touch lab5_b.c
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ gcc -o lab5_a lab5_a.c -pthread
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ gcc -o lab5_b lab5_b.c
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ gcc -o lab5_a lab5_a.c -pthread
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ gcc -o lab5_b lab5_b.c
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ ./lab5_a
Usage: ./lab5_a n d
  n: integer, > 1
  d: double (common difference)
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ ./lab5_b
Usage: ./lab5_b n d
  n: integer, > 1
  d: double (common difference)
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ ./lab5_a 6 2.5
Received: 79.700000
Received: 82.200000
Received: 84.700000
Received: 87.200000
Received: 89.700000
Received: 92.200000
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$ ./lab5_b 6 2.5
Received: 79.700000
Received: 82.200000
Received: 84.700000
Received: 87.200000
Received: 89.700000
Received: 92.200000
mycroft@mycroft-VMware-Virtual-Platform:~/hw5$
```