

To benchmark both programs, you would need to add two libraries: `<time.h>` and `<stdint.h>` while changing the number of runs to 1,000,000. This is because if it only maintained the 5 runs, it would be difficult to measure the time differences. The library `time.h` allows for the measure of time duration, while `stdint.h` gives accurate parameters for the program. With this simple function;

```
uint64_t get_time_ns() {  
    struct timespec ts;  
    clock_gettime(CLOCK_MONOTONIC, &ts);  
    return (uint64_t)ts.tv_sec * 1000000000 + ts.tv_nsec;  
}
```

It allows for the measurement over the span of the program. This can be seen with “`uint64_t start = get_time_ns();`” and “`uint64_t end = get_time_ns();`”, which allows for the time to be calculated

To get the result of the function, you can print it with this

```
printf("Execution time: %.3f ms\n", (end - start) / 1e6);
```

This print function prints the result in the form of “Execution time: 65.782 ms,” which is the result for the thread-based execution, while the process-based execution printed “Execution time: 290.325 ms”. With this, it is clear that the thread-based execution is more time-efficient and performs better.

(if you wish to run, delete the printf functions or else it will overflow with 1m producers and consumers)