

Pthread

Q1

- Approach:

As the code output shows for any “hello” word need to come “world” in follow. so we as producer and consumer with max buffer size as “1”. in this way for any item that producer creates, print “hello” as output and waits for 1 second, consumer remove it and print “world”.

We assign “MaxItems” as 5 so that producer create 5 items for consumer.

- Commands:

```
gcc -IC:/MinGW/include/ -pg q1.c -lpthread -o q1.exe  
q1.exe
```

- Comparison:

```
serial: 1000 microsecond  
parallel: 2000 microsecond
```

in almost all cases serial method was faster (at least half time) or had equal time in compare, it seems that added overhead for threading management and semaphore waits, make the process slower than serial manner.

Q2

- Approach:

The main PI calculation loop was divided to equal size based on number of threads and gave to them for processing.

The information about each thread process gave to them as bellow structure:

```
typedef struct argument{  
    int id;  
    int size;  
    double *sum;  
}argument;
```

- Commands:

```
gcc -IC:/MinGW/include/ -pg pi.c -lpthread -o pi.exe  
  
pi.exe <threads num>
```

- Comparison:

```
n = 10000000 , threads = 2  
pi (serial) = 3.141593          28000 microsecond  
pi (parallel) = 3.141593        14000 microsecond  
speed up : 2.00
```