

Parallel Computing HW0318

電機碩一 陳昕佑 61375017H

To fix the waste of one slot and fully utilize all N buffer slots, the most common solution is to explicitly track the item count in a separate shared memory variable instead of relying on $(in + 1) \% N == out$ to determine fullness.

The result of the original method would always have one slot left unused:

Full condition: $(*in + 1) \% N == *out$

Empty condition: $*in == *out$

```
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 1
Produce buffer[6]: 1
Next in:7, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 2
Produce buffer[7]: 2
Next in:0, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 3
Produce buffer[0]: 3
Next in:1, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 4
Produce buffer[1]: 4
Next in:2, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 5
Produce buffer[2]: 5
Next in:3, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 6
Produce buffer[3]: 6
Next in:4, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 7
Produce buffer[4]: 7
Next in:5, out:6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./producer-with-circular-queue 8
buffer is full!
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[6]: 1
in:5, next out:7
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[7]: 2
in:5, next out:0
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[0]: 3
in:5, next out:1
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[1]: 4
in:5, next out:2
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[2]: 5
in:5, next out:3
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[3]: 6
in:5, next out:4
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
consumed buffer[4]: 7
in:5, next out:5
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./consumer-with-circular-queue
buffer is empty!
```

Solution:

Add a *count* variable in shared memory to track the actual number of items in the queue, that allows to use all N slots and clearly distinguish full and empty states.

1. Create an additional shared memory segment for *count*.
2. Update producer to check $count < N$ before inserting.
3. Update consumer to check $count > 0$ before consuming.
4. Adjust $count++$ in producer, and $count--$ in consumer.

Full condition: $*count == N$

Empty condition: $*count == 0$

```
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 1
Produced buffer[5] = 1
Next in: 6, out: 5, count: 1
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 2
Produced buffer[6] = 2
Next in: 7, out: 5, count: 2
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 3
Produced buffer[7] = 3
Next in: 0, out: 5, count: 3
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 4
Produced buffer[0] = 4
Next in: 1, out: 5, count: 4
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 5
Produced buffer[1] = 5
Next in: 2, out: 5, count: 5
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 6
Produced buffer[2] = 6
Next in: 3, out: 5, count: 6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 7
Produced buffer[3] = 7
Next in: 4, out: 5, count: 7
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 8
Produced buffer[4] = 8
Next in: 5, out: 5, count: 8
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_producer-with-circular-queue 9
Buffer is full!
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[5] = 1
in: 5, next out: 6, count: 7
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[6] = 2
in: 5, next out: 7, count: 6
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[7] = 3
in: 5, next out: 0, count: 5
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[0] = 4
in: 5, next out: 1, count: 4
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[1] = 5
in: 5, next out: 2, count: 3
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[2] = 6
in: 5, next out: 3, count: 2
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[3] = 7
in: 5, next out: 4, count: 1
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Consumed buffer[4] = 8
in: 5, next out: 5, count: 0
(base) rueii@rueii-Alienware-Aurora-R6:~/Graduation/First_Year/Parallel-Computing/WEEK4$ ./HW0318_consumer-with-circular-queue
Buffer is empty!
```