

Test explanation	Picture
<p>This test has 5 writers, each writing 5 items. We call these internal variables num_writers and num_items_to_write respectively. The reader calls fifo_rd num_writers*num_items_to_write times, so 25 times in this scenario.</p> <p>Since the program returned and all fifo_rd didn't hang, the test passed.</p>	 <pre> marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src\$./fifo -w 5 -n 5 Writer 1 stream completed Writer 2 stream completed Writer 3 stream completed Writer 4 stream completed Writer 5 stream completed Reader successfully called fifo_rd 25 times All children collected </pre>
<p>This is the same test as above but the reader prints the result of fifo_rd after every invocation.</p>	 <pre> marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src\$./fifo -w 5 -n 5 Writer 1 stream completed Writer 2 stream completed Reader read a 0 Reader read a 1 Reader read a 2 Reader read a 3 Reader read a 4 Reader read a 0 Reader read a 1 Reader read a 2 Reader read a 3 Reader read a 4 Writer 3 stream completed Reader read a 0 Reader read a 1 Reader read a 2 Reader read a 3 Reader read a 4 Reader read a 0 Reader read a 1 Reader read a 2 Reader read a 0 Reader read a 1 Reader read a 2 Reader read a 3 Reader read a 4 Reader read a 3 Reader read a 4 Reader successfully called fifo_rd 25 times Writer 5 stream completed Writer 4 stream completed All children collected marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src\$ </pre>

This is an example with a lot of writers, each writing 5 items.

```
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./fifo -w 62 -n 5
Writer 1 stream completed
Writer 2 stream completed
Writer 3 stream completed
Writer 4 stream completed
Writer 5 stream completed
Writer 6 stream completed
Writer 7 stream completed
Writer 8 stream completed
Writer 9 stream completed
Writer 11 stream completed
Writer 10 stream completed
Writer 12 stream completed
Writer 13 stream completed
Writer 14 stream completed
Writer 15 stream completed
Writer 16 stream completed
Writer 17 stream completed
Writer 18 stream completed
Writer 19 stream completed
Writer 20 stream completed
Writer 21 stream completed
Writer 23 stream completed
Writer 22 stream completed
Writer 24 stream completed
Writer 25 stream completed
Writer 27 stream completed
Writer 26 stream completed
Writer 28 stream completed
Writer 29 stream completed
Writer 30 stream completed
Writer 31 stream completed
Writer 33 stream completed
Writer 32 stream completed
Writer 34 stream completed
Writer 36 stream completed
Writer 35 stream completed
Writer 37 stream completed
Writer 38 stream completed
Writer 39 stream completed
Writer 40 stream completed
Writer 41 stream completed
Writer 42 stream completed
Writer 43 stream completed
Writer 44 stream completed
Writer 45 stream completed
Writer 46 stream completed
Writer 47 stream completed
Writer 48 stream completed
Writer 49 stream completed
Writer 50 stream completed
Writer 52 stream completed
Writer 51 stream completed
Writer 55 stream completed
Writer 54 stream completed
Writer 56 stream completed
Writer 57 stream completed
Writer 58 stream completed
Writer 53 stream completed
Writer 59 stream completed
Writer 60 stream completed
Writer 61 stream completed
Writer 62 stream completed
Reader successfully called fifo_rd 310 times
All children collected
```

This is an example with a few writers writing a lot of items.

```
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./fifo -w 5 -n 50000
Writer 1 stream completed
Writer 2 stream completed
Writer 4 stream completed
Writer 3 stream completed
Writer 5 stream completed
Reader successfully called fifo_rd 250000 times
All children collected
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./fifo -w 5 -n 50000
```

This is an example with a lot of writers writing a lot of items. (A lot of items means more than MYFIFO_BUFSIZ items)

```
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./fifo -w 30 -n 5000
Writer 4 stream completed
Writer 3 stream completed
Writer 2 stream completed
Writer 1 stream completed
Writer 8 stream completed
Writer 7 stream completed
Writer 6 stream completed
Writer 5 stream completed
Writer 14 stream completed
Writer 9 stream completed
Writer 13 stream completed
Writer 10 stream completed
Writer 11 stream completed
Writer 22 stream completed
Writer 24 stream completed
Writer 23 stream completed
Writer 15 stream completed
Writer 18 stream completed
Writer 16 stream completed
Writer 12 stream completed
Writer 19 stream completed
Writer 25 stream completed
Writer 17 stream completed
Writer 20 stream completed
Writer 26 stream completed
Writer 27 stream completed
Writer 21 stream completed
Writer 28 stream completed
Writer 29 stream completed
Writer 30 stream completed
Reader successfully called fifo_rd 150000 times
All children collected
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$
```

This is an example where the locking sequence was deliberately messed up

```
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./fifo -w 50 -n 1000
Reader read a 0
^C
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$
```

This is a semaphore stress test where N_PROC processes try to grab the semaphore.

This is a spinlock test where 10 processes count to 1 million each. The first execution is with locking. Afterward, the locks were removed and now the count is inconsistent.

```
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./spinlock
100000000
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ make spinlock
gcc -g -o spinlock test_spinlock.c spinlock.c tas64.S
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./spinlock
1555124
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./spinlock
1940366
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./spinlock
1790278
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$ ./spinlock
1734602
marco@xiaolong:~/cooper/fa_25/ece357/pset6/synchronization/src$
```