2. Coffman conditions for a deadlock

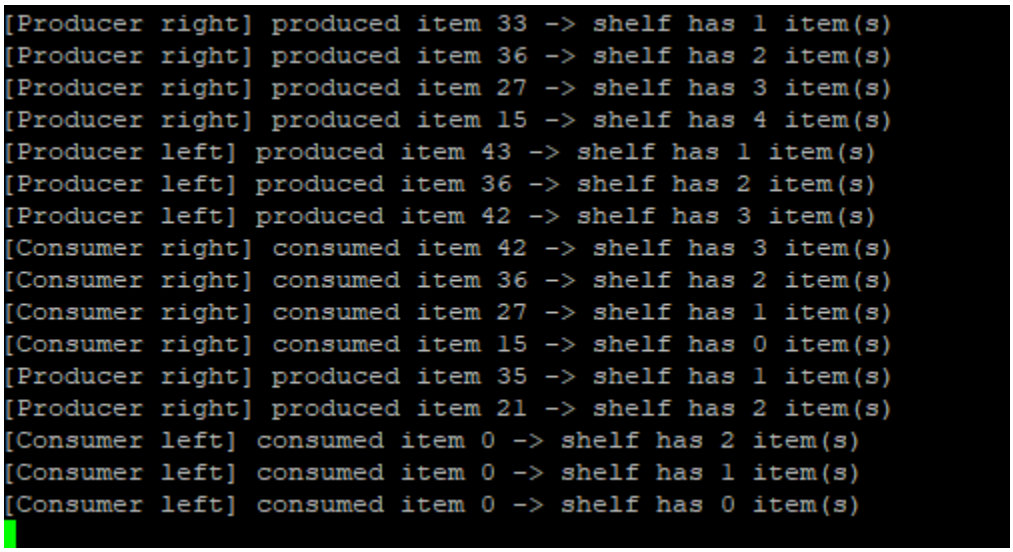
Circular wait: A chain is formed by the waits of the threads, in this case one thread wants Leftmutex and the other thread wants right. This is very similar to the hold and wait condition in our case because only 2 threads are in the loop

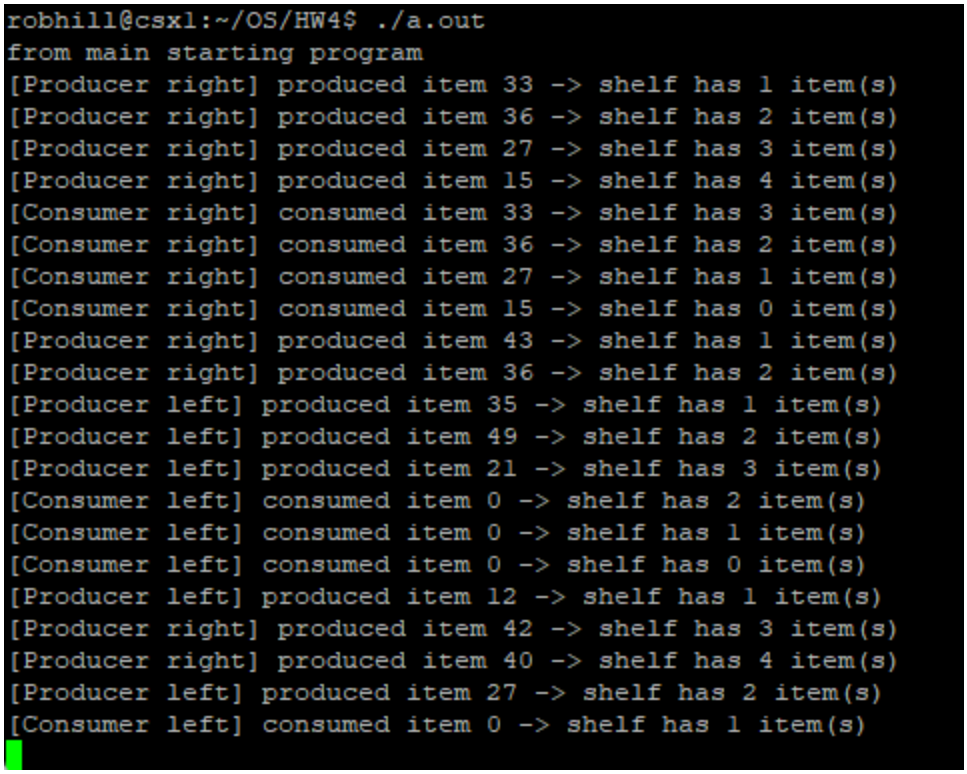
Mutual exclusion: This condition is provided by the mutex locks in pthreads, when they are held they exclude other processes or threads from working on that same section

No preemption: The provided OS can't take the mutex from either program to resolve the deadlock

Hold and wait: either producer is holding onto a mutex and asking for another one, instead of claiming both at the same time or in order. In this case, one thread is holding Left mutex and waiting for right mutex, and the other is holding Right and waiting for left

Examples of the left and right bakery deadlocking (not shown with monitor) The green box is showing that the program is stuck





Example of showing the monitor working correctly

```
Consumer 2] consumed item 14 -> left shelf has 3 item(s)
Consumer 2] consumed item 6 -> left shelf has 2 item(s)
Consumer 2] consumed item 43 -> left shelf has 1 item(s)
Consumer 2] consumed item 41 -> left shelf has 0 item(s)
Producer left] produced item 27 -> shelf has 1 item(s)
Producer left] produced item 1 -> shelf has 2 item(s)
Producer left] produced item 37 -> shelf has 3 item(s)
Producer left] produced item 28 -> shelf has 4 item(s)
Monitor] Everything working as expected
Consumer 1] consumed item 27 -> left shelf has 3 item(s)
Consumer 1] consumed item 1 -> left shelf has 2 item(s)
Consumer 1] consumed item 37 -> left shelf has 1 item(s)
Consumer 1] consumed item 28 -> left shelf has 0 item(s)
Producer left] produced item 25 -> shelf has 1 item(s)
Producer left] produced item 45 -> shelf has 2 item(s)
Producer left] produced item 29 -> shelf has 3 item(s)
Producer left] produced item 37 -> shelf has 4 item(s)
Consumer 1] consumed item 25 -> left shelf has 3 item(s)
Producer left] produced item 35 -> shelf has 4 item(s)
Monitor] Everything working as expected
Consumer 2] consumed item 45 -> left shelf has 3 item(s)
Consumer 2] consumed item 29 -> left shelf has 2 item(s)
Consumer 2] consumed item 37 -> left shelf has 1 item(s)
Consumer 2] consumed item 35 -> left shelf has 0 item(s)
Producer left] produced item 18 -> shelf has 1 item(s)
Consumer 2] consumed item 18 -> left shelf has 0 item(s)
Monitor] Everything working as expected
obhill@csxl:~/OS/HW4$
```

Example of showing deadlock being caught

```
robhill@csxl:~/OS/HW4$ gcc Q2Dmonitor.c
robhill@csxl:~/OS/HW4$ ./a.out
monitor function created
[Producer right] produced item 33 -> shelf has 1 item(s)
[Producer left] produced item 36 -> shelf has 1 item(s)
[Producer left] produced item 15 -> shelf has 2 item(s)
[Producer left] produced item 43 -> shelf has 3 item(s)
[Producer left] produced item 35 -> shelf has 4 item(s)
[Monitor] Everything working as expected
[Monitor] no activity for last 2 checks, possible stall detected
[Monitor] no activity for last 2 checks, possible stall detected
[Monitor] no activity for last 2 checks, possible stall detected
[Monitor] no activity for last 2 checks, possible stall detected
```