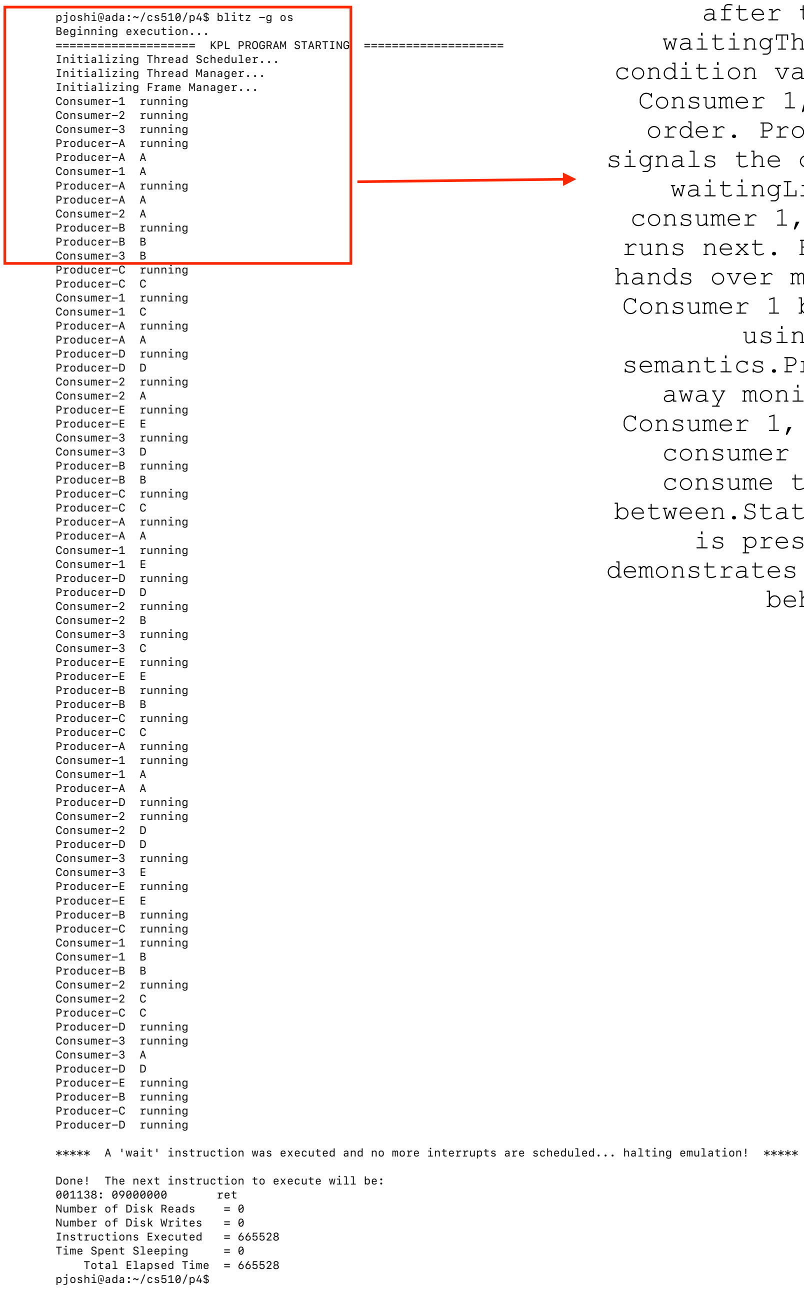

```
pjoshi@ada:~/cs510/p4$ blitz -g os
Beginning execution...
====================  KPL PROGRAM STARTING  ====================
Initializing Thread Scheduler...
Initializing Thread Manager...
Initializing Frame Manager...
Consumer-1  running
Consumer-2  running
Consumer-3  running
Producer-A  running
Producer-A  A
Consumer-1  A
Producer-A  running
Producer-A  A
Consumer-2  A
Producer-B  running
Producer-B  B
Consumer-3  B
Producer-C  running
Producer-C  C
Consumer-1  running
Consumer-1  C
Producer-A  running
Producer-A  A
Producer-D  running
Producer-D  D
Consumer-2  running
Consumer-2  A
Producer-E  running
Producer-E  E
Consumer-3  running
Consumer-3  D
Producer-B  running
Producer-B  B
Producer-C  running
Producer-C  C
Producer-A  running
Producer-A  A
Consumer-1  running
Consumer-1  E
Producer-D  running
Producer-D  D
Consumer-2  running
Consumer-2  B
Consumer-3  running
Consumer-3  C
Producer-E  running
Producer-E  E
Producer-B  running
Producer-B  B
Producer-C  running
Producer-C  C
Producer-A  running
Consumer-1  running
Consumer-1  A
Producer-A  A
Producer-D  running
Consumer-2  running
Consumer-2  D
Producer-D  D
Consumer-3  running
Consumer-3  E
Producer-E  running
Producer-E  E
Producer-B  running
Producer-C  running
Consumer-1  running
Consumer-1  B
Producer-B  B
Consumer-2  running
Consumer-2  C
Producer-C  C
Producer-D  running
Consumer-3  running
Consumer-3  A
Producer-D  D
Producer-E  running
Producer-B  running
Producer-C  running
Producer-D  running

*****  A 'wait' instruction was executed and no more interrupts are scheduled... halting emulation!  *****

Done!  The next instruction to execute will be:
001138: 09000000       ret
Number of Disk Reads    = 0
Number of Disk Writes   = 0
Instructions Executed   = 665528
Time Spent Sleeping     = 0
    Total Elapsed Time  = 665528
pjoshi@ada:~/cs510/p4$
```

After 3 consumers run one after the other, waitingThread list of condition variable contains Consumer 1, 2, 3 in that order. Producer A runs, signals the oldest thread on waitingList which is consumer 1, so consumer 1 runs next. Here Producer A hands over monitor mutex to Consumer 1 by signaling it using Hoare semantics.Producer A gives away monitor mutex to Consumer 1, hence no other consumer can come and consume the buffer in between.State of the buffer is preserved.This demonstrates Hoare Semantics behavior