

Chapter 30

1. The file implements a producer and a consumer using conditional variables and locks.
2. Tried buffer sizes 1-3, behavior does not change.

```
luayan@LAPTOP-111L4GAE:~/chapter30$ ./main-two-cvs-while -l 3 -m 3 -p 1 -c 1 -v
NF      P0 C0
0 [*-- ] p0
0 [*-- ] c0
0 [*-- ] p1
1 [u 0 f-- ] p4
1 [u 0 f-- ] p5
1 [u 0 f-- ] p6
1 [u 0 f-- ] c1
1 [u 0 f-- ] p0
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
0 [*-- ] c0
0 [*-- ] p1
1 [u 1 f-- ] p4
1 [u 1 f-- ] p5
1 [u 1 f-- ] p6
1 [u 1 f-- ] c1
0 [*-- ] c4
0 [*-- ] p0
0 [*-- ] c5
0 [*-- ] c6
0 [*-- ] p1
0 [*-- ] c0
1 [f-- ] u 2 p4
1 [f-- ] u 2 p5
1 [f-- ] u 2 p6
1 [f-- ] u 2 c1
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
1 [uEOS f-- ] [main: added end-of-stream marker]
1 [uEOS f-- ] c0
1 [uEOS f-- ] c1
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
Consumer consumption:
C0 -> 3
```

The max value of num_full should be equal to buffer size.

3.

Windows PowerShell

```
PS C:\Users\yong\Desktop\chapter30> .\main-two-cvs-while.exe -l 3 -m 1 -p 1 -c 1 -v
NF      P0 C0
0 [*-- ] p0
0 [*-- ] p1
1 [* 0 ] p4
1 [* 0 ] p5
1 [* 0 ] p6
1 [* 0 ] p0
1 [* 0 ] p1
1 [* 0 ] p2
1 [* 0 ] c0
1 [* 0 ] c1
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
0 [*-- ] c0
0 [*-- ] p3
1 [* 1 ] p4
1 [* 1 ] p5
1 [* 1 ] p6
1 [* 1 ] p0
1 [* 1 ] c1
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
0 [*-- ] c0
0 [*-- ] p1
1 [* 2 ] p4
1 [* 2 ] p5
1 [* 2 ] p6
1 [* 2 ] c1
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
1 [*EOS ] c0
1 [*EOS ] [main: added end-of-stream marker]
1 [*EOS ] c1
0 [*-- ] c4
0 [*-- ] c5
0 [*-- ] c6
Consumer consumption:
C0 -> 3
```

luyan@LAPTOP-III4GAE: ~/chapter30

```
luyan@LAPTOP-III4GAE:~/chapter30$ ./main-two-cvs-while -l 3 -m 1 -p 1 -c 1 -v
NF P0 C0
0 [*--] p0
0 [*--] c0
0 [*--] p1
1 [* 0] p4
1 [* 0] p5
1 [* 0] p6
1 [* 0] c1
1 [* 0] p0
0 [*--] c4
0 [*--] c5
0 [*--] c6
0 [*--] p1
1 [* 1] p4
1 [* 1] c0
1 [* 1] p5
1 [* 1] p6
1 [* 1] c1
1 [* 1] p0
0 [*--] c4
0 [*--] c5
0 [*--] c6
0 [*--] p1
0 [*--] c0
1 [* 2] p4
1 [* 2] p5
1 [* 2] p6
1 [* 2] c1
0 [*--] c4
0 [*--] c5
0 [*--] c6
1 [*EOS] [main: added end-of-stream marker]
1 [*EOS] c0
1 [*EOS] c1
0 [*--] c4
0 [*--] c5
0 [*--] c6
Consumer consumption:
C0 -> 3
```

4.

./main-two-cvs-while -p 1 -c 3 -m 1 -C 0,0,0,1,0,0,0:0,0,0,1,0,0,0:0,0,0,1,0,0,0 -l 10 -v -t

```
Consumer consumption:
C0 -> 0
C1 -> 10
C2 -> 0
Total time: 12.03 seconds
```

5. No changes.

./main-two-cvs-while -p 1 -c 3 -m 3 -C 0,0,0,1,0,0,0:0,0,0,1,0,0,0:0,0,0,1,0,0,0 -l 10 -v -t

```
Consumer consumption:
C0 -> 0
C1 -> 0
C2 -> 10
Total time: 12.03 seconds
```

6.

```
Consumer consumption:
C0 -> 0
C1 -> 10
C2 -> 0
Total time: 13.03 seconds
```

7. The total time does not change.

```
Consumer consumption:  
C0 -> 0  
C1 -> 10  
C2 -> 0  
  
Total time: 13.03 seconds
```

8. No problem with one producer and one consumer.

9. Single cv might cause all threads (producer and consumers) all end up sleeping, when there are multiple consumers. (P10)

10. With a single producer and consumer, the code works. But with 2 consumers there might be problems. (P8)