

CSCE 311 Spring 2020

Process Coordination

Name: Key

Due: 2/20/20

- Consider a version of the bounded buffer problem in which there are one producer process (P_1) and two consumer processes (P_2 and P_3) all sharing the same buffer. Assume that the size of the buffer is $n = 4$, and that we start with a completely empty buffer. The structure of P_1 , P_2 , and P_3 as well as the semaphores and buffer is shown below:

```
/* structure of P1 */
do {
    .....
    produce an item in nextp
    .....
    wait(empty)
    wait(mutex)
    buffer[in] = nextp
    in = (in + 1) % n
    signal(mutex)
    signal(full)
}while(1)
```

```
/* structure of P2 & P3 */
do {
    wait(full)
    wait(mutex)
    nextp = buffer[out]
    out = (out + 1) % n
    signal(mutex)
    signal(empty)
    .....
    consume item in nextp
    .....
}while(1)
```

P3	item 0
P1	item 1
	item 2
	item 3
2	in
2	out
4	empty
-1	full
1	mutex

Assume a preemptive scheduler and that all processes start in the ready queue at the same time in the order from head to tail, P_1 , P_2 , and P_3 (P_3 at the head of the queue). Let the process priorities be: $P_3 = 2$, $P_2 = 2$, and $P_1 = 4$.

Draw the contents of the indices "in" and "out", as well as the state of the semaphores and the contents of the buffer **after 2 items have been consumed**. In the case of the buffers, simply notate each item with the name of the process that accessed it last.

P_3 : Full: -1 $\Rightarrow P_3$ blocks
 P_2 : Full: -2 $\Rightarrow P_2$ blocks
 P_1 : produce item, MT: 3, mux: 0, item 0: P_1 , in: 1, mux: 1, Full: -1
 \Rightarrow unblocks P_3 , P_3 preempts P_1
 P_3 : mux: 0, item 0: P_3 , out: 1, mux: 1, MT: 4, consumes item
 P_3 : Full: -2 $\Rightarrow P_3$ blocks & P_1 resumes
 P_1 : produce item, MT: 3, mux: 0, item 1: P_1 , in: 2, mux: 1, Full: -1
 \Rightarrow unblocks P_2 , P_2 preempts P_1
 P_2 : mux: 0, item 1: P_2 , out: 2, mux: 1, MT: 4, consumes item