

ProducersConsumer using semaphore

```
void producer () {  
    while(1) {  
        item := produce()  
        sem_wait(not_full)  
        write(buffer, item)  
        sem_signal(not_empty)  
    }  
}
```

```
void consumer () {  
    while(1) {  
        sem_wait(not_empty)  
        item := read(buffer)  
        sem_signal(not_full)  
        consume(item)  
    }  
}
```

➡ What if we have multiple consumers and/or producers?

```
sem_init(&not_full, 0, n)
```

```
sem_init(&not_empty, 0, 1)
```

Producer Consumer **using a semaphore**

```
sem_init(&not_full, n)
sem_init(&not_empty, 0)
```

```
void producer () {
    while(1) {
        item := produce()
        sem_wait(&not_full)
        write(buffer, item)
        sem_signal(&not_empty)
    }
}
```

```
void consumer () {
    while(1) {
        sem_wait(&not_empty)
        item := read(buffer)
        sem_signal(&not_full)
        consume(item)
    }
}
```

➡ What if we have multiple consumers and/or producers?

(Bad) Producers consumers

```
sem_init(&not_full, 0, n)
sem_init(&not_empty, 0, 1)
sem_init(&mutex, 0, 1)
```

```
void producer () {
    while(1) {
        item := produce()
        sem_wait(&mutex)
        sem_wait(&not_full)
        write(buffer, item)
        sem_signal(&not_empty)
        sem_signal(&mutex)
    }
}
```

```
void consumer () {
    while(1) {
        sem_wait(&mutex)
        sem_wait(&not_empty)
        item := read(buffer)
        sem_signal(&not_full)
        sem_signal(&mutex)
        consume(item)
    }
}
```