

# Deadlock



**Deadlock** when one thread tries to access a resource that a second holds, and vice-versa

- They can never make progress

```
void thread1 () {  
    ...  
    sem_wait(sem1)  
    sem_wait(sem2)  
    /* critical section */  
    sem_signal(sem2)  
    sem_signal(sem1)  
    ...  
}
```

```
void thread2 () {  
    ...  
    sem_wait(sem2)  
    sem_wait(sem1)  
    /* critical section */  
    sem_signal(sem1)  
    sem_signal(sem2)  
    ...  
}
```

# (Good) 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(&not_full)
        sem_wait(&mutex)
        write(buffer, item)
        sem_signal(&mutex)
        sem_signal(&not_empty)
    }
}
```

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