Process Synchronization-3

CS3600 Spring 2022

Review

- wait() - p()
- signal() – v()

- Semaphore for precedence
- Semaphore for resource allocation

- Bounded buffer Problem

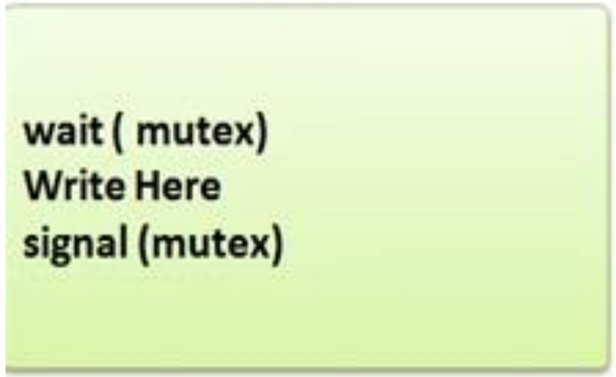
Readers-Writers Problem

- Suppose that a database is to be shared among several concurrent processes. Some of these processes may want only to read the database, whereas others may want to update the database. We distinguish between these two types of processes by referring to the former as readers and to the latter as writers.

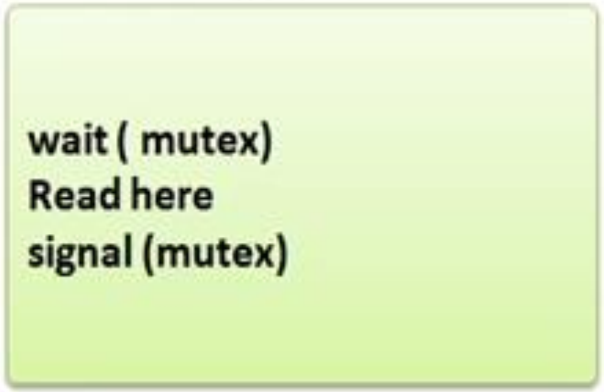
Writer	Reader
Writer can modify	Only Reads
Only one writer allowed	Multiple readers allowed
When writing reading is not allowed	When reading writing is not allowed

Basic solution R/W problem

Mutex =1

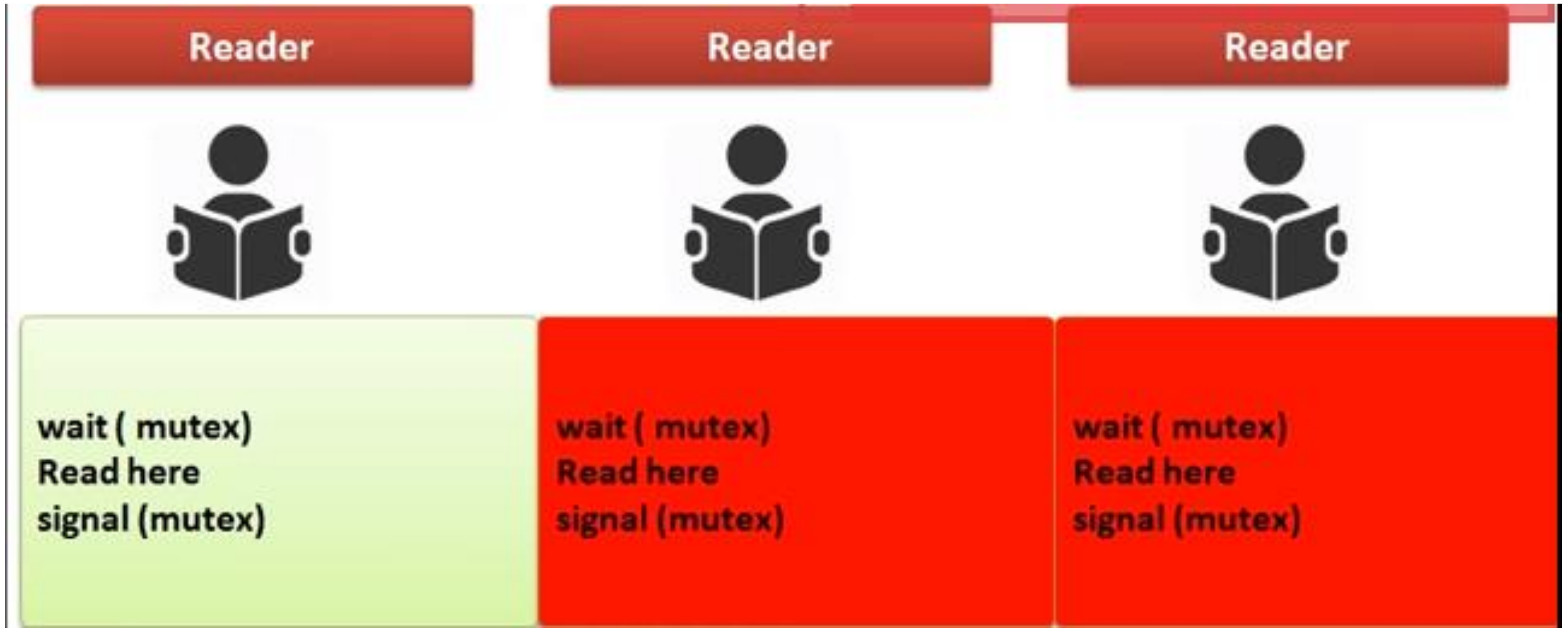


```
wait ( mutex)
Write Here
signal (mutex)
```



```
wait ( mutex)
Read here
signal (mutex)
```

Basic Solution -Problem

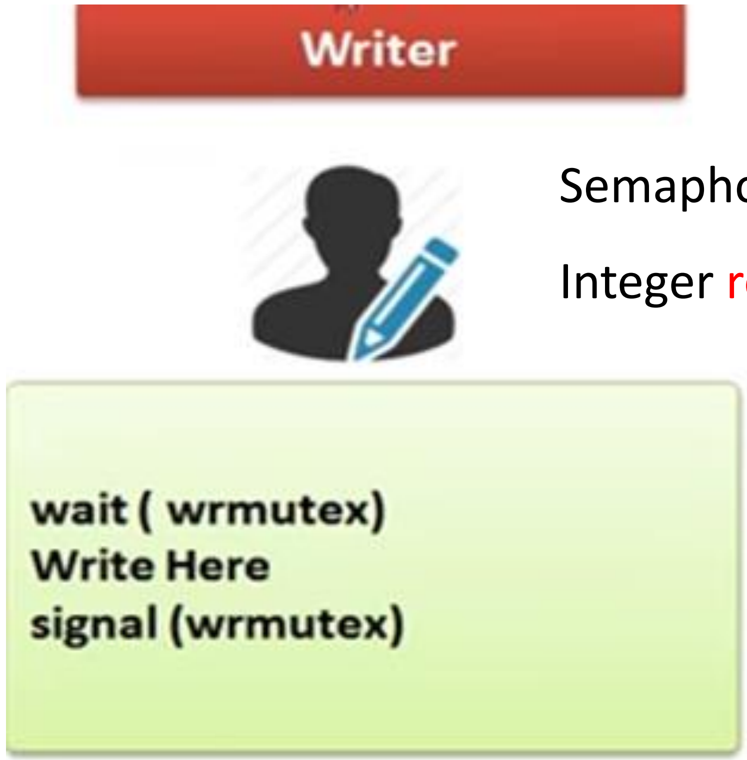


wait (mutex)
Read here
signal (mutex)

wait (mutex)
Read here
signal (mutex)

wait (mutex)
Read here
signal (mutex)

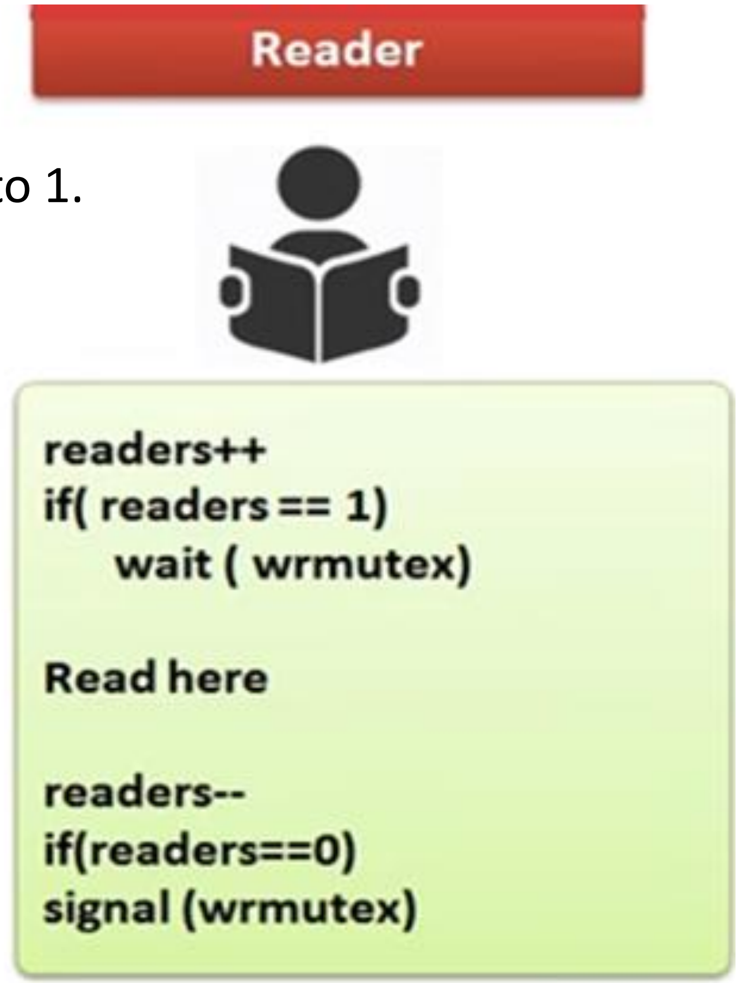
Multiple Reader solution



```
wait ( wrmutex)
Write Here
signal (wrmutex)
```

Semaphore wrmutex initialized to 1.

Integer readers initialized to 0.



```
readers++
if( readers == 1)
    wait ( wrmutex)

Read here

readers--
if(readers==0)
signal (wrmutex)
```

Semaphore wrmutex =1

Semaphore mutex=1

Integer readers = 0.

Writer

```
wait(wrmutex)

Write Here

signal(wrmutex)
```

Reader

```
wait (mutex)
readers++
if(readers==1)
    wait(wrmutex)
signal(mutex)

Read Here

wait (mutex)
readers--
if(readers==0)
    signal(wrmutex)
signal(mutex)
```

Monitors

- Monitor- Abstract Data Type (ADT)
  - The monitor construct ensures that only one process at a time is active within the monitor.
  - The programmer does not need to code this synchronization constraint explicitly.
  - For modelling some synchronization schemes programmer needs condition variables.

condition x; // Condition variables

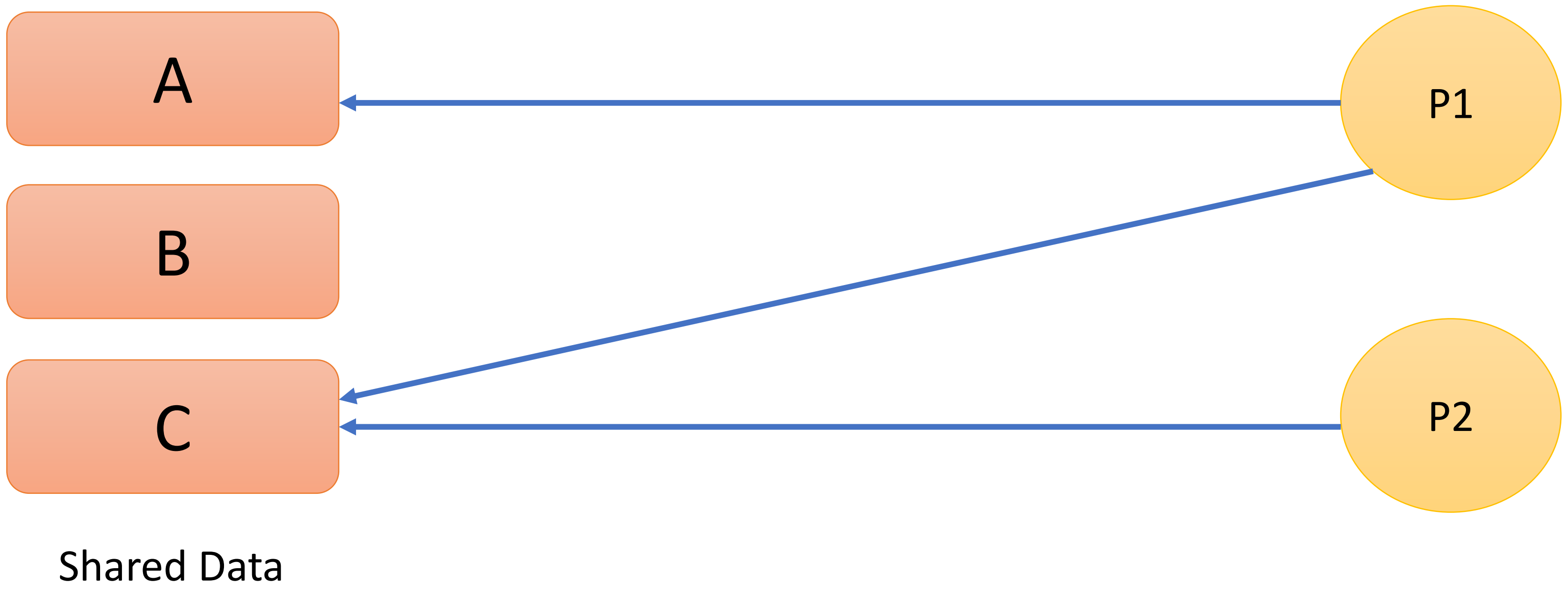
x.wait()

x.signal()

Implementing a Monitor

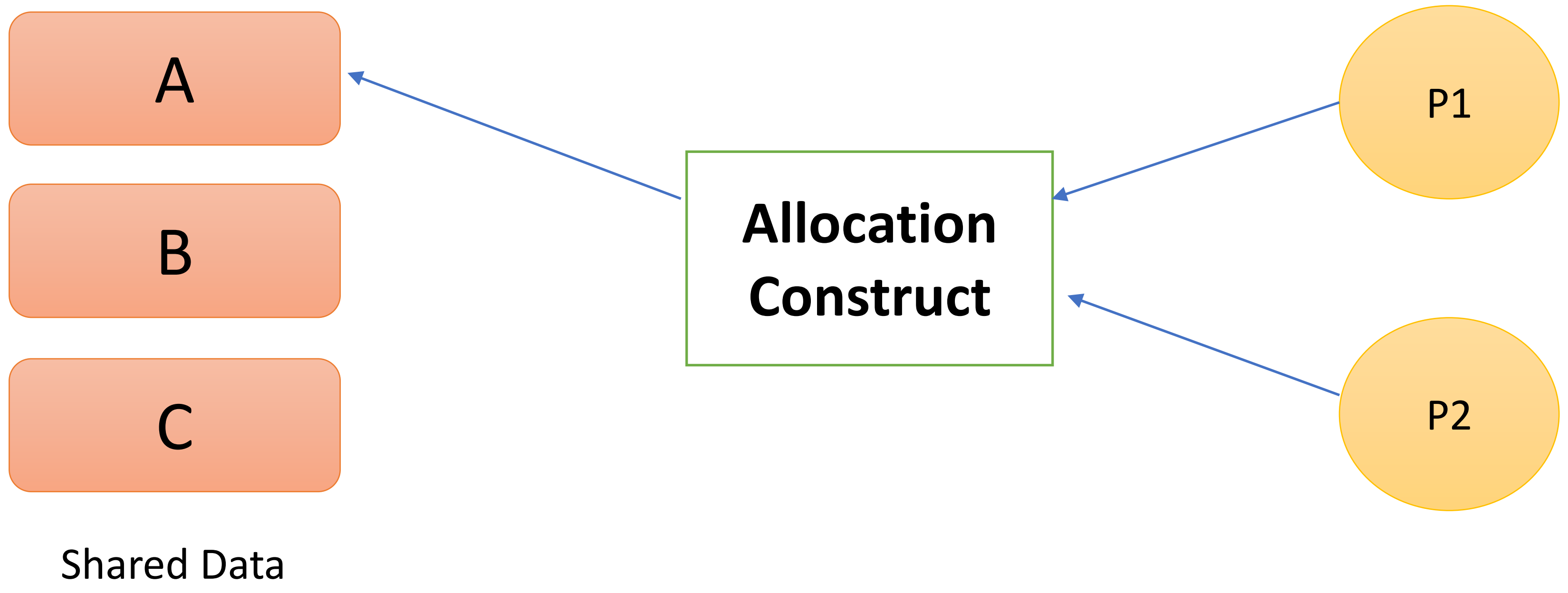
- For each monitor, a binary semaphore mutex (initialized to 1) is provided to ensure mutual exclusion.
- A process must execute wait(mutex) before entering the monitor and must execute signal(mutex) after leaving the monitor.
- A ***condition variable*** (a named queue) is used so the processes can wait for some condition to become true.

Monitor

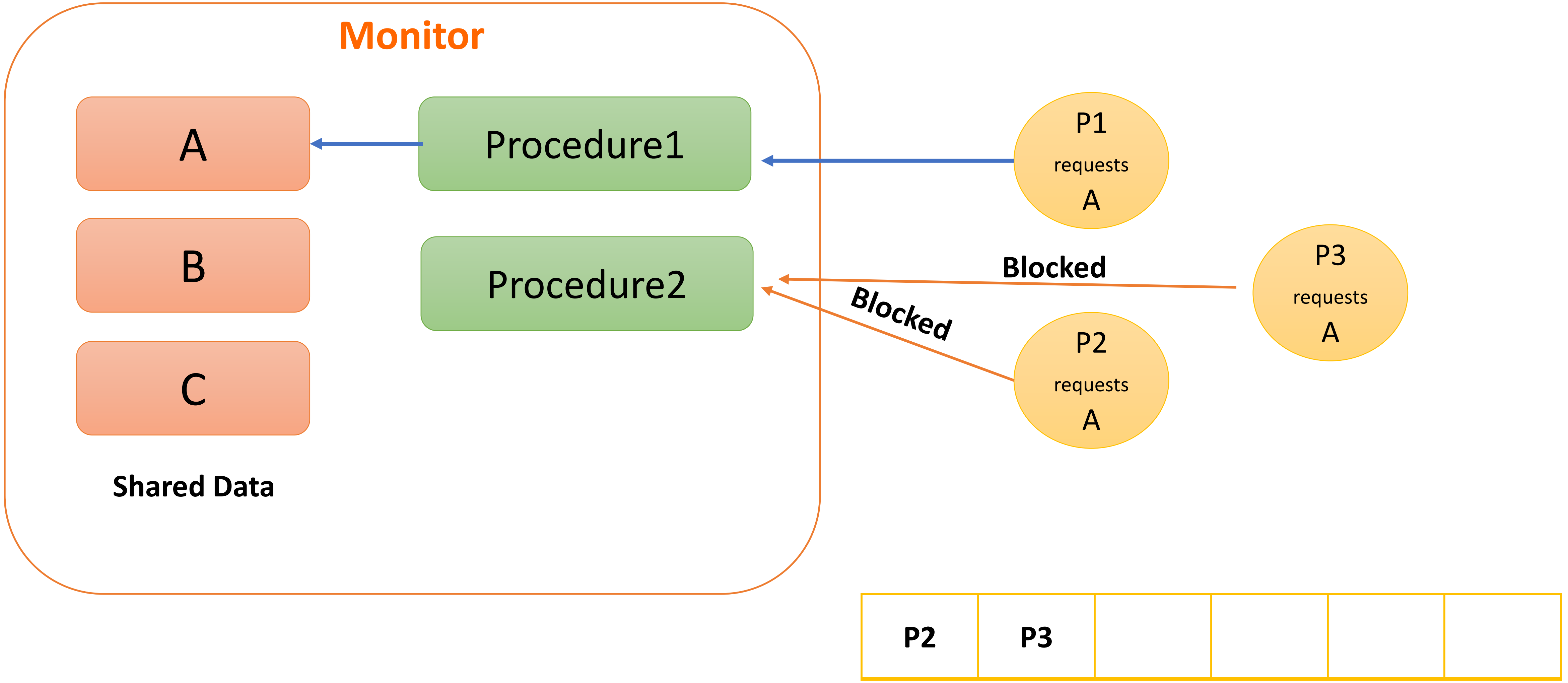


When both access the same resource –there can be a **race condition**

Monitor Cont.

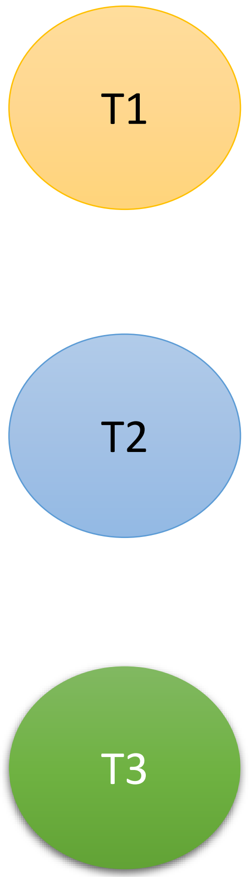


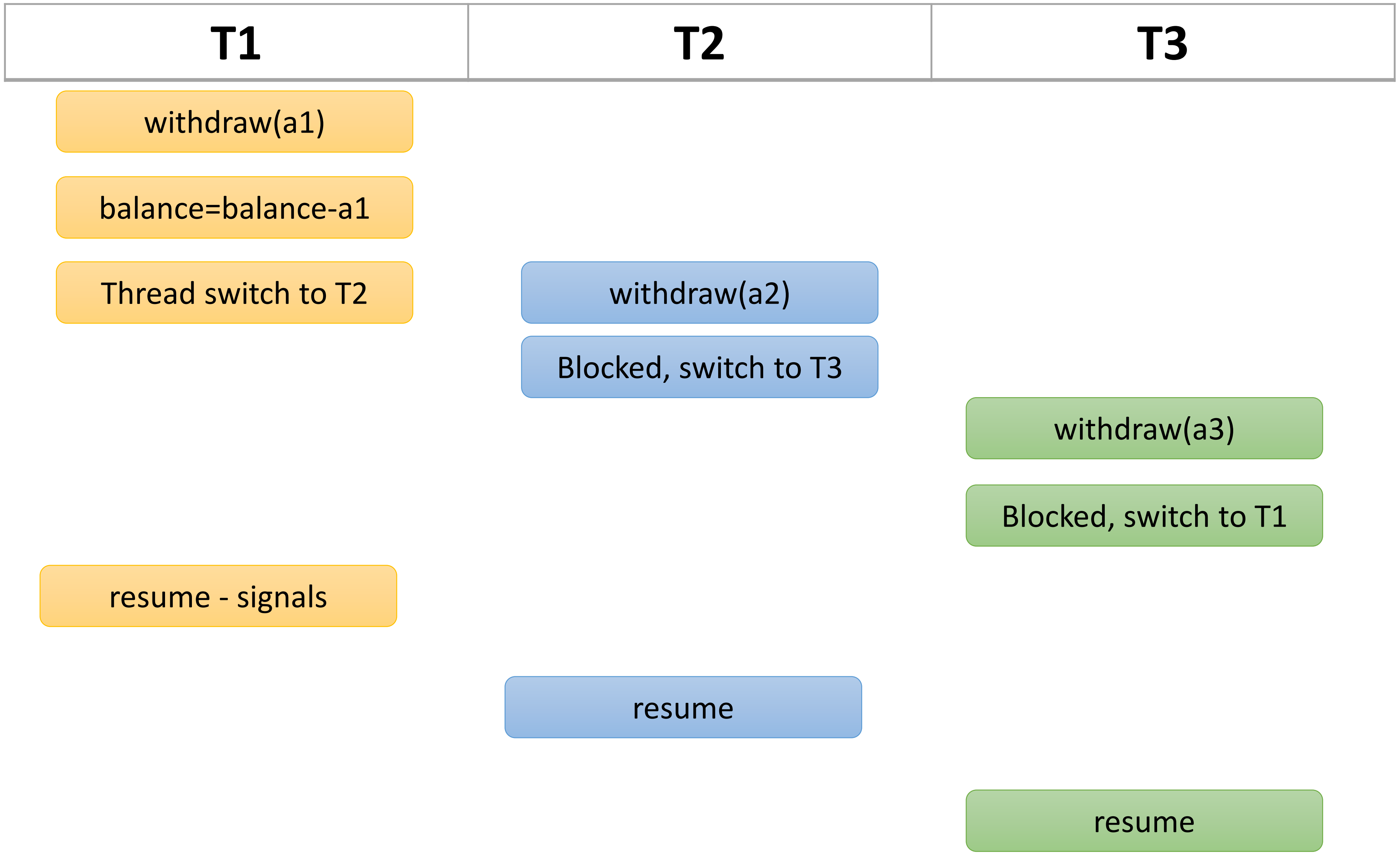
Monitor Cont.



Monitor- Example

```
Monitor account
{
  double balance;
  withdraw (amount)
    {
      balance =balance-amount;
      return balance;
    }
}
```



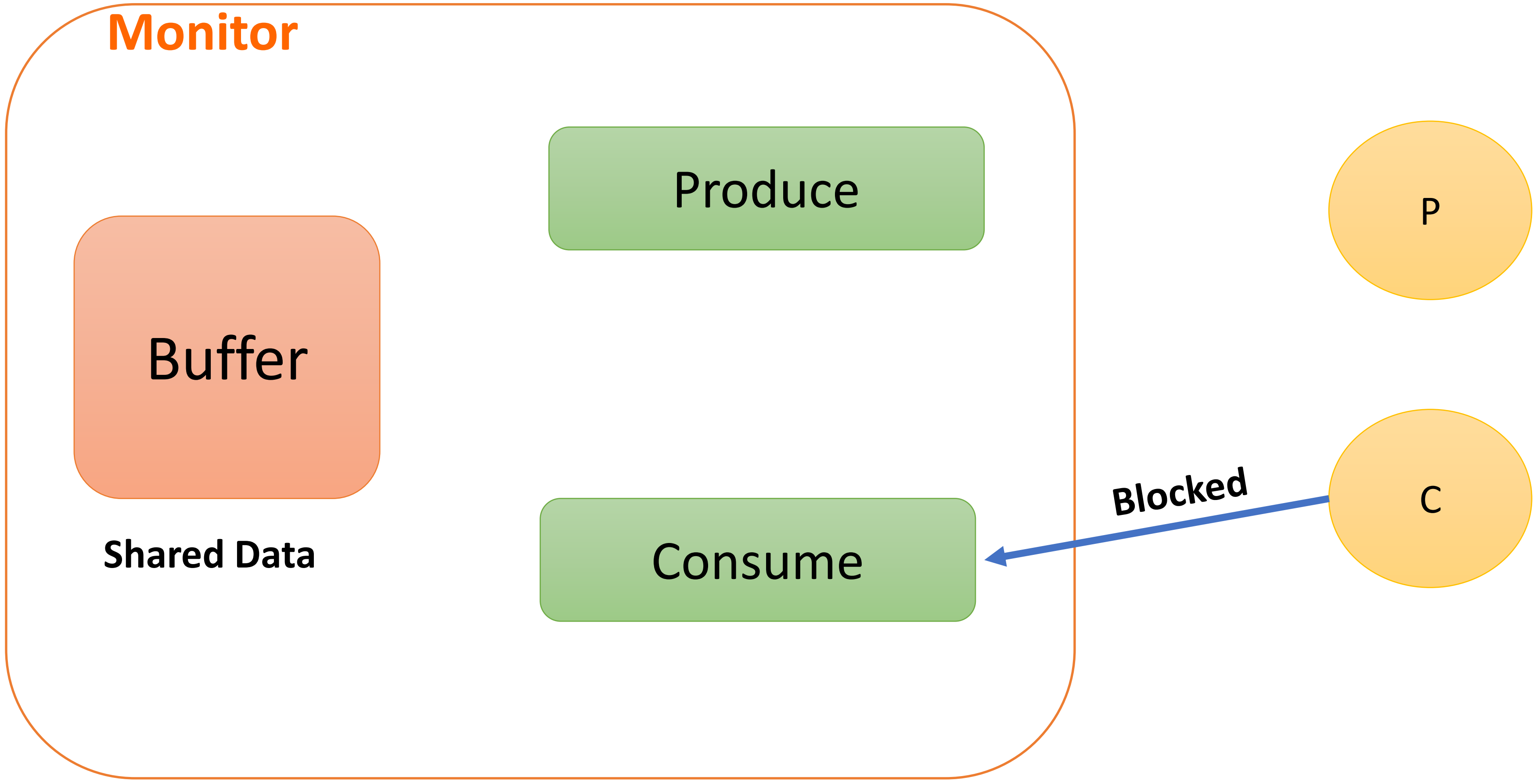

T1
T2
T3
withdraw(a1)
balance=balance-a1
Thread switch to T2
withdraw(a2)
Blocked, switch to T3
withdraw(a3)
Blocked, switch to T1
resume - signals
resume
resume

Conditional variable

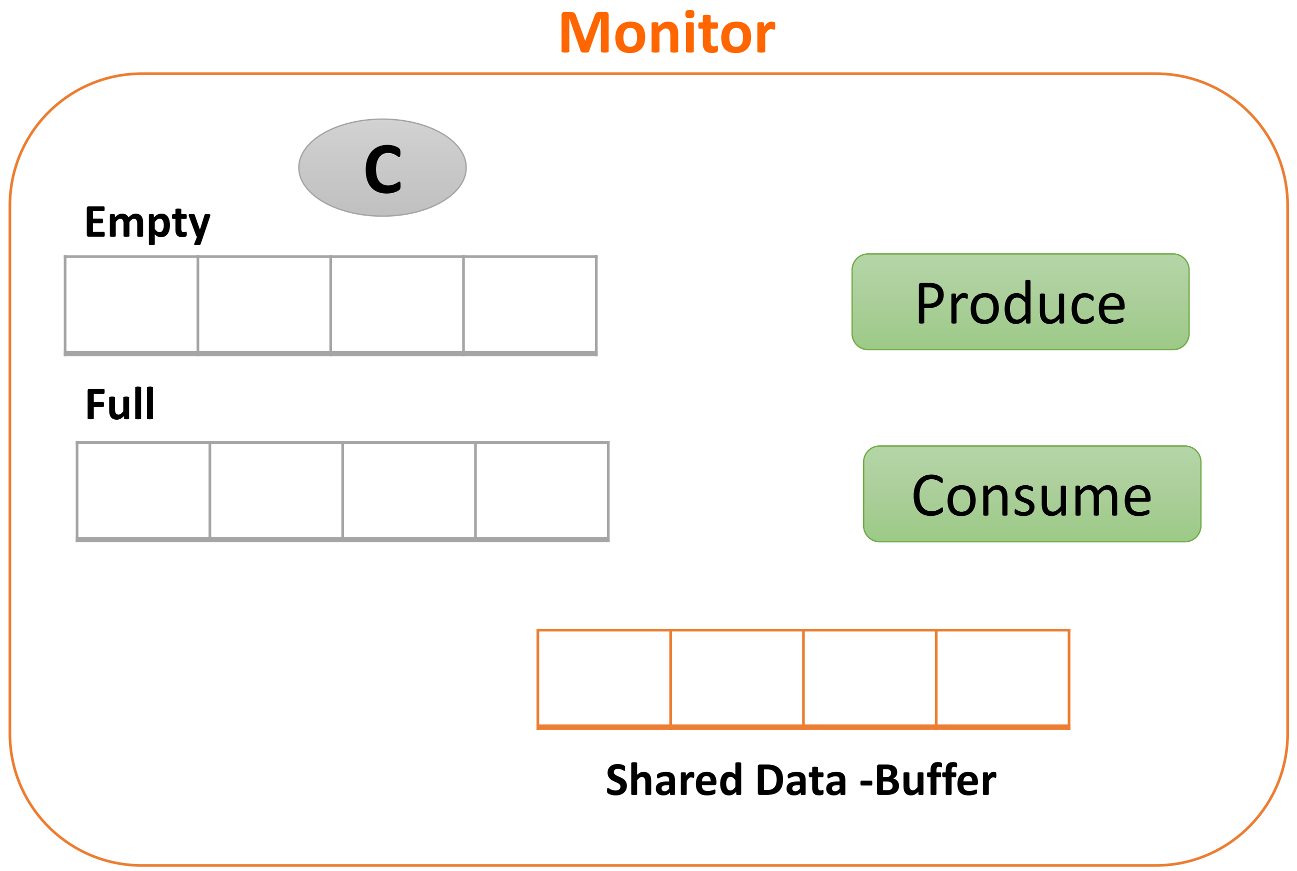
- Wait operation
  - If resource is currently not available, the current process is put into sleep in a waiting queue and releases the lock of the monitor
- Signal operation
  - Signal operation send a signal to one process in sleep state to resume its operation. Lock is transferred to the process that resumes.
- Broadcast
  - Signals to all waiting processes and resumes as prioritized in the queue or longest waiting time.

Bounded buffer problem

Conditional Variables are used



Bounded buffer problem



```
monitor BoundBuff
{
 int data[4];
 int  count;
 condition full;
 condition empty;
 produce()
{
  if count== 4 then full.wait
  add item to buffer
  count++;
  empty.signal
}
consume()
{
 if count==0 then empty.wait
 read item from buffer
 count--
 full.signal
}}
```

Worksheet 4

- Complete Q1 and Q2

- Q3 is HW4 (10 points)
  - Read Dining Philosophers Problem (4.5.7 Activity) from Zybook.
  - And trace the given program
  - Write the states of all philosophers whenever a state changes for any philosopher.
  - Read the problem and ask question if any
  - Submit the completed table as a pdf file.

Announcements as on 03/1/2022

- Complete Worksheet 04– Q1 and Q2, submit Q3 for HW4.
- In ZyBook
  - 4.5.4: Understand Readers-writers problem using a monitors.
  - 4.5.12: Behavior of the dining-philosophers monitor.

- Next class 03/03 Q&A only, those who have questions on topics until now come to class , class will not be streamed in Teams.

- Mid term take home programming will be available from 03/03 Noon to 03/05 Once opened will get 24 hours to complete and submit.

- Mid Term exam on 03/08 – topics until Monitors