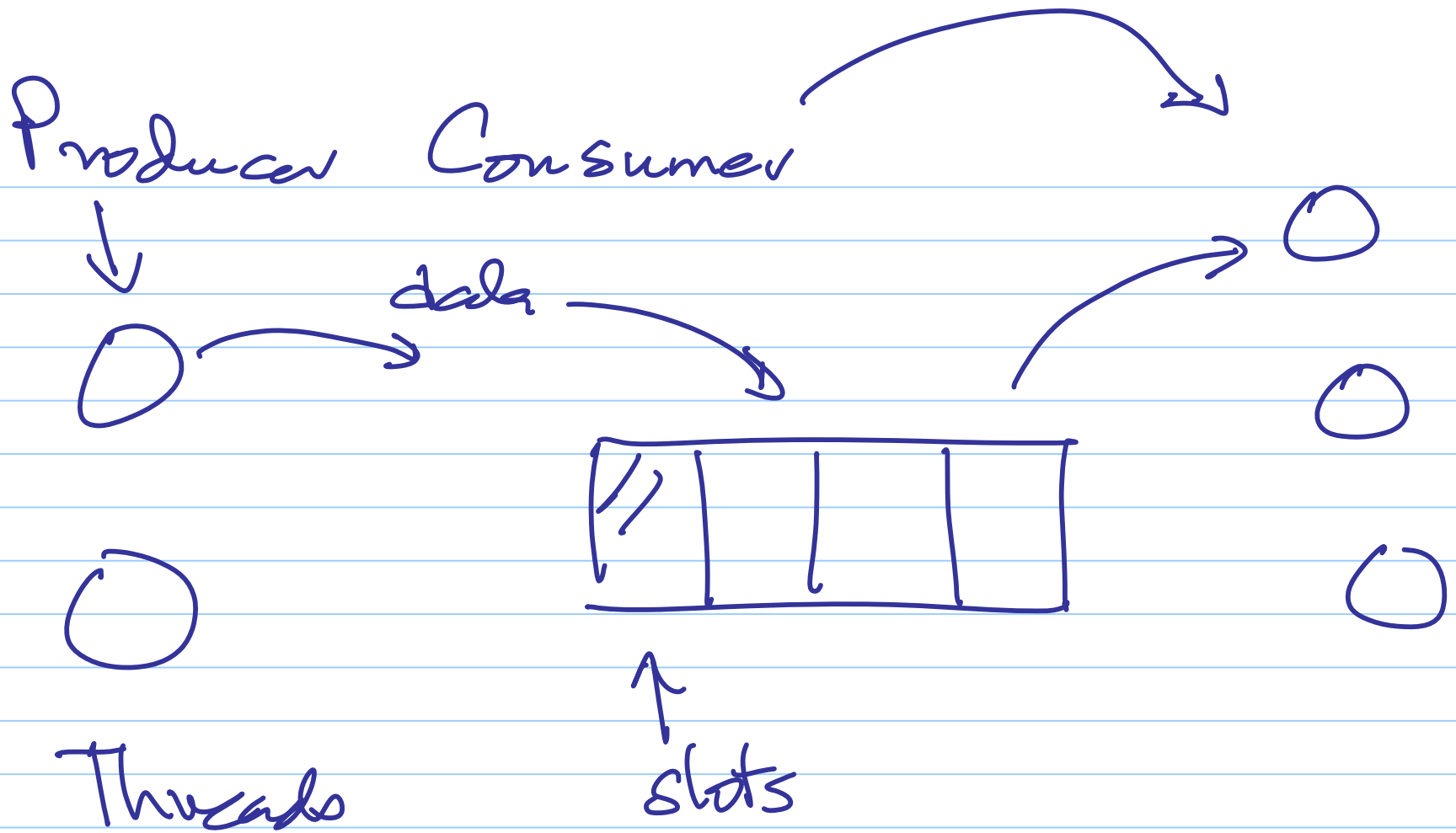


Classic Process Coordination Problems

- Producer Consumer
(bounded buffer)
- Readers & Writers
- Dining Philosophers



producer

∞ loop

{ obtain data

[copy data to buffer slot]

Consumer

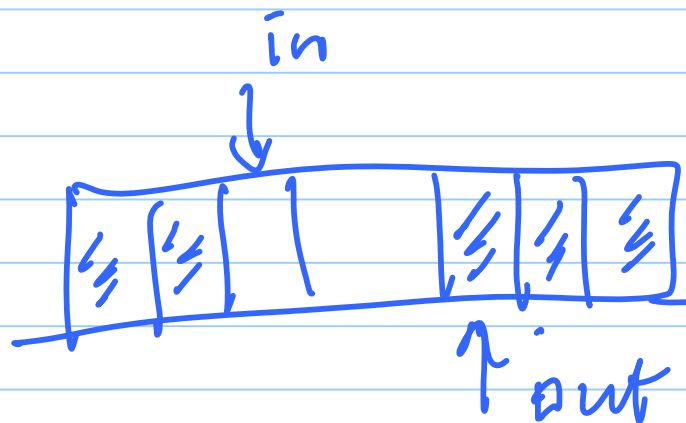
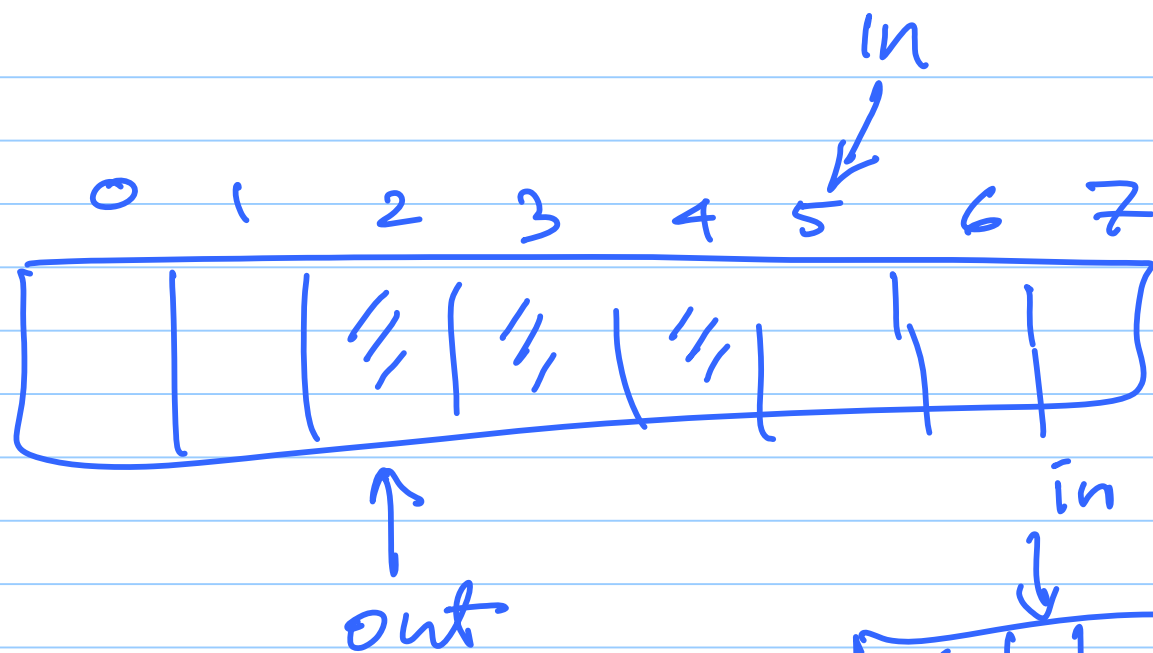
∞ loop

{ [copy data from buffer slot;
use the data]

→ buffer_t buffer[N]
type int in, out = 0;

put(item)
[
 buffer[in] = item
 in = (in + 1) % N
]

item = get
{
 item =
 buffer[out]
 out =
 (out + 1) % N
 return item
}



put(item)

full = N
empty = 0

P(full) •
buff[in] = item;
in = (in++) % N
V(empty) •

↑
1 prod, 1 consumer

get

P(empty) •
item = buff[out]
out = (out++) % N
V(full) •

put

mutex = 1

P(full) P(mutex)

buff(
in + 1

— V(mutex)

V(empty)

P(empty)

P(mutex)

buffer, out

V(mutex)

V(full)

P(full)

P(mutex P)

arr^{add}
V(mutex P)

V(empty)

P(empty)

P(mutex C)

sub^{sub}
 arr^{sub}

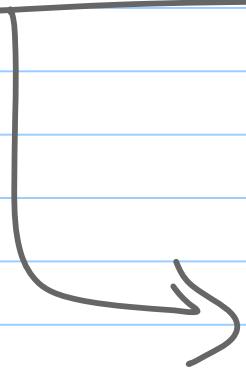
V(mutex C)

V(full)

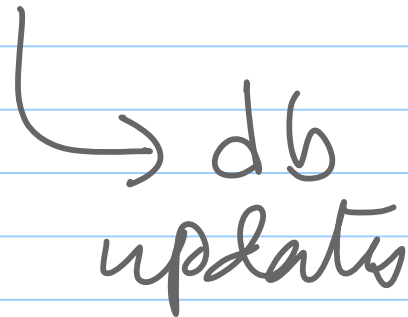
$P(mutex)$
 $P(full)$
 $cs[$
 $V(empty)$
 $V(mutex)$

$P(mutex)$
 $cs[$
 $P(empty)$
 $V(full)$
 $V(mutex)$

Readers & Writers

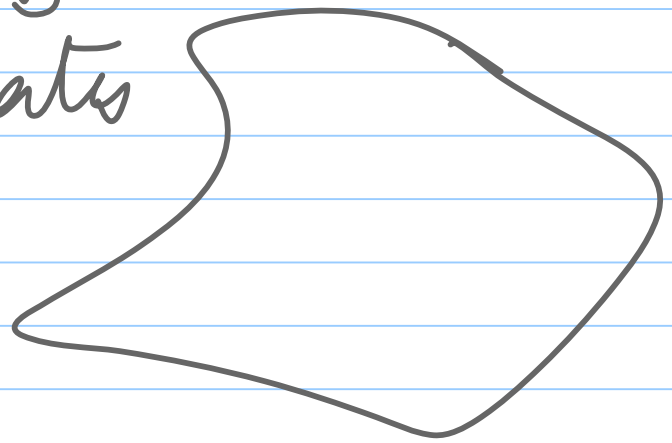


query
operations



db
updates

database



R & W problem

↳ single writer OK

exclusive
OR

multiple readers may
access the db at the
same time

→ none is OK

$$\underline{S=1}$$

$P(S)$

READ

$V(S)$

$P(S)$

write

$V(S)$

Reader Entry

```
P(mutex);  
rc++;  
if rc==1 then P(wsem);  
V(mutex);
```

READ

Reader Exit

```
P(mutex);  
rc--;  
if rc==0 then V(wsem);
```

Writer Entry

```
P(wsem)
```

WRITE

Writer Exit

```
V(wsem)
```

Wsem
= 1

rc
[reader
count]
= 0

mutex
= 1