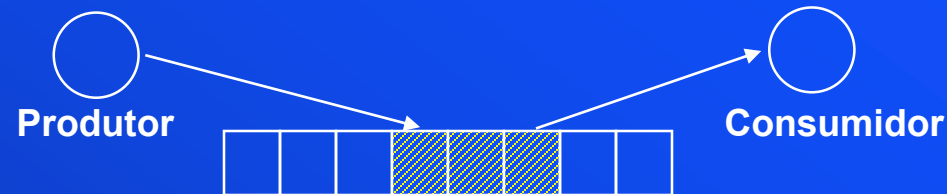


Problema do(s) Produtor(es) / Consumidor(es)

```
Var  
  ElemType Buffer[N] = ...  
  Semaphore full, empty, mutex;
```

Inicialização:

```
  full.value = 0;  
  empty.value = N;  
  mutex.value = 1;
```



Process Producer

```
...  
Repeat  
...  
Produce (Item) ;  
Wait (Empty) ;  
Wait (Mutex) ;  
Append (Item) ;  
Signal (Mutex) ;  
Signal (Full) ;  
...  
Until ...;
```

Process Consumer

```
...  
Repeat  
...  
Wait (Full) ;  
Wait (Mutex) ;  
Item=Take () ;  
Signal (Mutex) ;  
Signal (Empty) ;  
Consume (Item) ;  
...  
Until ...;
```

◆ full

- p/ sincronizar os 2 processos;
- não significa *buffer* cheio mas que tem pelo menos 1 item.

◆ empty

- p/ sincronizar os 2 processos;
- não significa *buffer* vazio mas que há espaço no *buffer*

◆ mutex

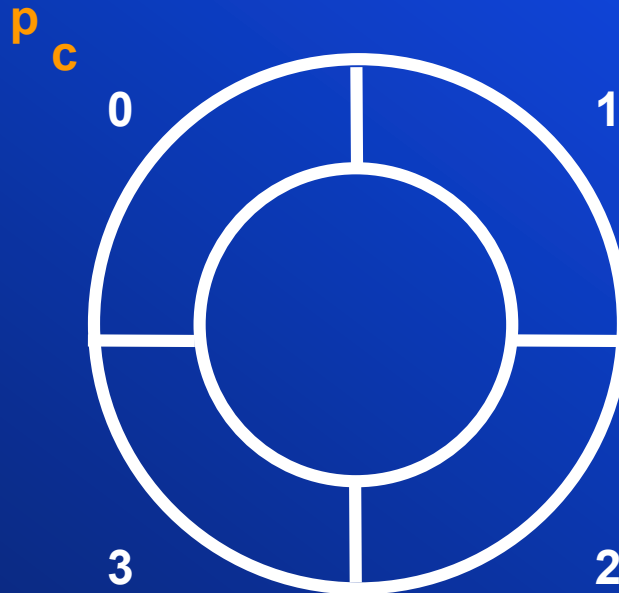
- p/implementar a exclusão mútua.

Problema do(s) Produtor(es) / Consumidor(es)

```
char buffer[4];  
Semaphore full, empty, mutex;  
  
semInit(full,0);  
semInit(empty,N);  
semInit(mutex,1);  
int p = c = 0;  
int count = 0;
```

Process Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```



Process Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```

Problema do(s) Produtor(es) / Consumidor(es)

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;

```

full

0

empty

4

mutex

1

p

0

c

0

p
c

item

Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;

```

Problema do(s) Produtor(es) / Consumidor(es)

item A

full 0
 empty 4
 mutex 1

p 0
 c 0

item

Process Producer

```
...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
```



p c



Process Consumer

```
...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

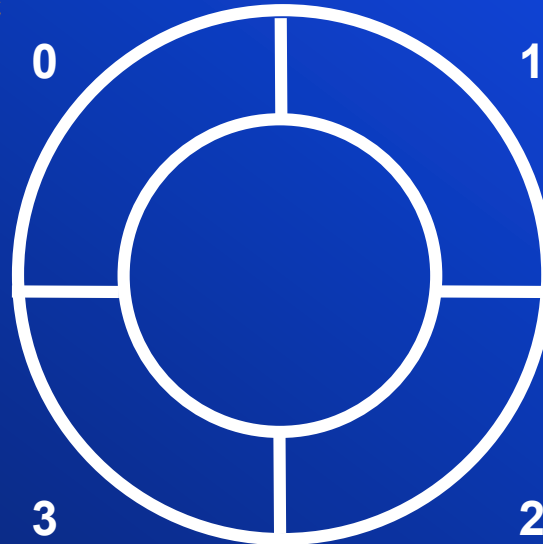
item A

full 0
empty 4
mutex 1

p 0
c 0

item Process Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```

p
cProcess Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

item A

full 0
 empty 3
 mutex 1

p 0
 c 0

item

Process Producer

```

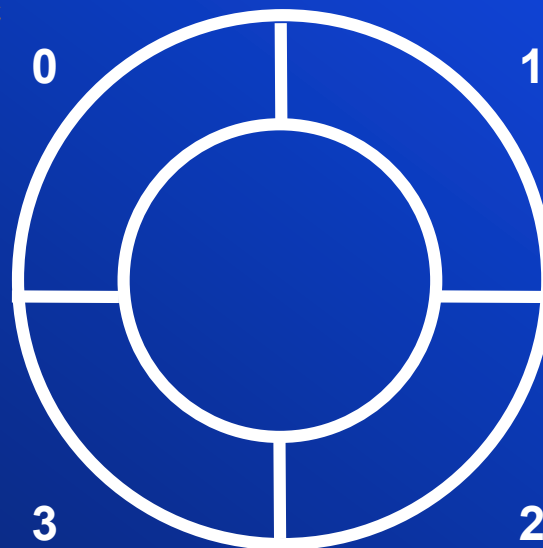
...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



p
c



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

item

A

full

0

empty

3

mutex

0

p

1

c

0

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;

```

c

0

A

p

1

3

2

Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;

```


Problema do(s) Produtor(es) / Consumidor(es)

item

A

full

0

empty

3

mutex

1

p

1

c

0

item

Process Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```



C

0

A

p

1

3

2

Process Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

item A

full 1
 empty 3
 mutex 1

p 1
 c 0

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

item

B

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;

```

full 1

empty 3

mutex 1

p

1

c

0

item

Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;

```

C

0

A

p

1

3

2

Problema do(s) Produtor(es) / Consumidor(es)

item B

full 1
 empty 3
 mutex 1

p 1
 c 0

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

item

B

full

1

empty

2

mutex

1

p

1

c

0

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;

```



c

0

A

p

1

3

2

Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;

```



Problema do(s) Produtor(es) / Consumidor(es)

item B

full 1
 empty 2
 mutex 0

p 1
 c 0

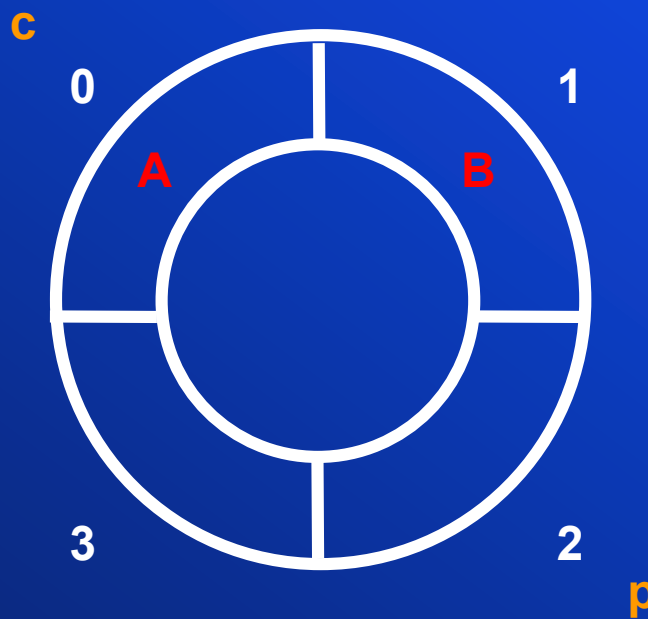
item

Process Producer

```
...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
```



Process Consumer

```
...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

item

B

full 1
empty 2
mutex 1

p

1

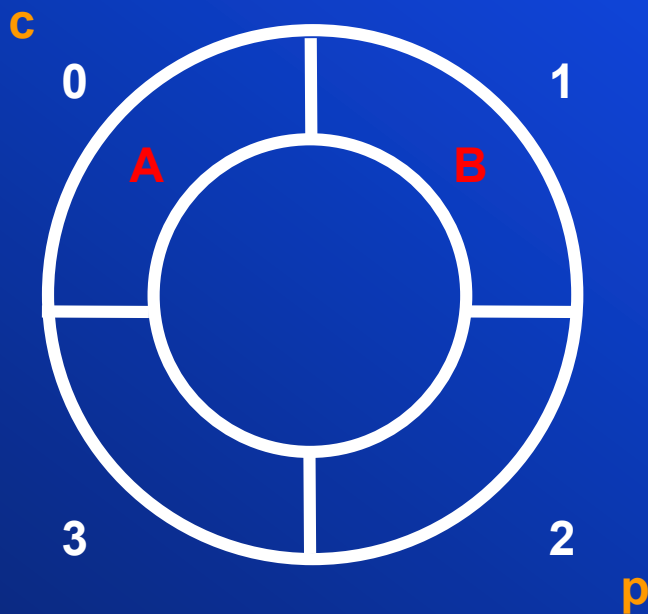
c

0

item

Process Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```

Process Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

item

B

full

2

empty

2

mutex

1

p

1

c

0

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;

```

c

0

A

1

B

3

2

p

Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;

```


Problema do(s) Produtor(es) / Consumidor(es)

item B

full 2
 empty 2
 mutex 1

p 1
 c 0

item

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

item B

full 1
 empty 2
 mutex 1

p 1
 c 0

item

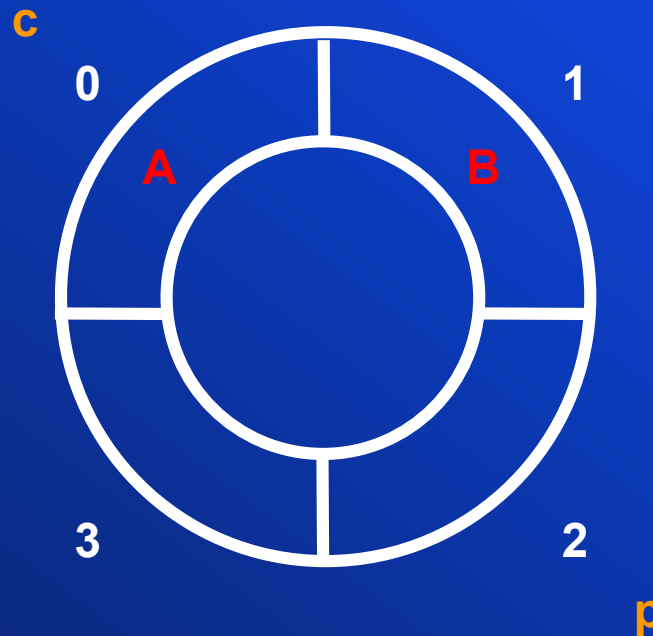
Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

item B

full 1
 empty 2
 mutex 0

p 1
 c 1

item B

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



p

Problema do(s) Produtor(es) / Consumidor(es)

item B

full 1
empty 2
mutex 1

p 1
c 1

item BProcess Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```

Process Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

item B

full 1
 empty 3
 mutex 1

p 1
 c 1

item B

Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)



Problema do(s) Produtor(es) / Consumidor(es)

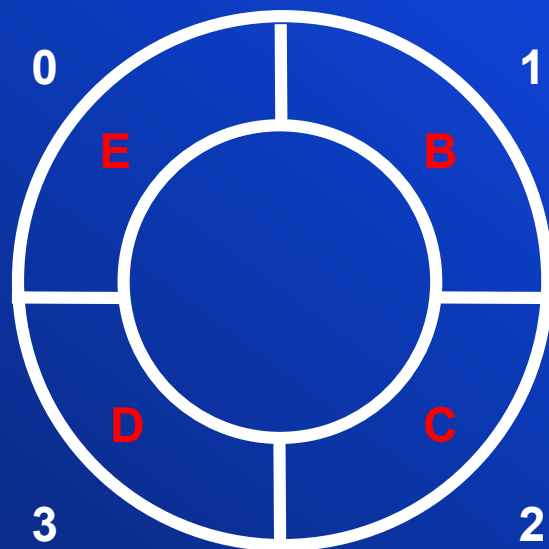
item F

full 3
empty 0
mutex 0

p 1
c 2

item BProcess Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```

Process Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

item F

full 3
 empty 1
 mutex 1

p 1
 c 2

item B

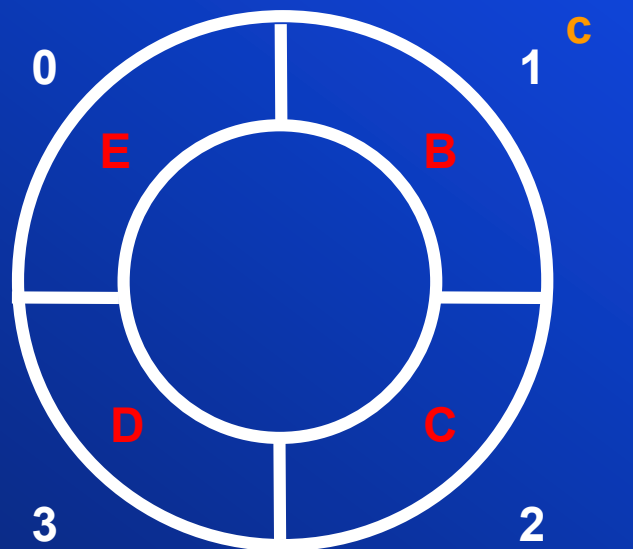
Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

item F

full 3
 empty 1
 mutex 1

p 1
 c 2

item B

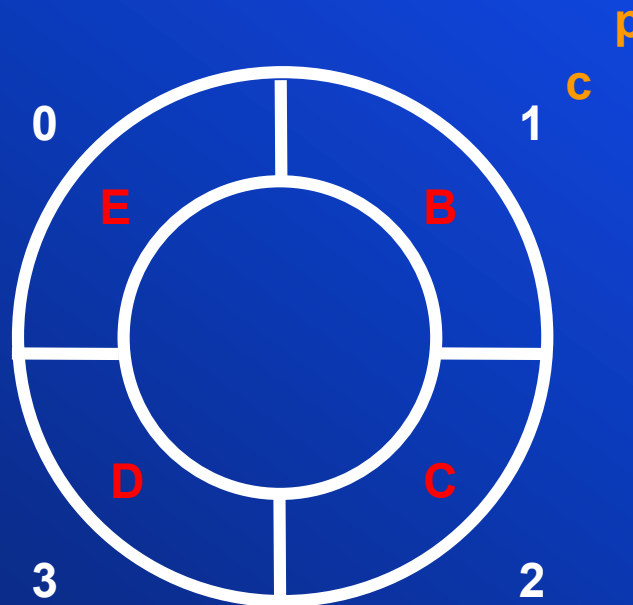
Process Producer

```

...
Repeat
...
Produce(item);
Wait(Empty);

Wait(Mutex);
buffer[p] = item;
p = (p+1) % 4;
Signal(Mutex);

Signal(Full);
...
Until ...;
  
```



Process Consumer

```

...
Repeat
...
Wait(Full);

Wait(Mutex);
item=buffer[c];
c = (c+1) % 4;
Signal(Mutex);

Signal(Empty);
Consume(item);
...
Until ...;
  
```



Problema do(s) Produtor(es) / Consumidor(es)

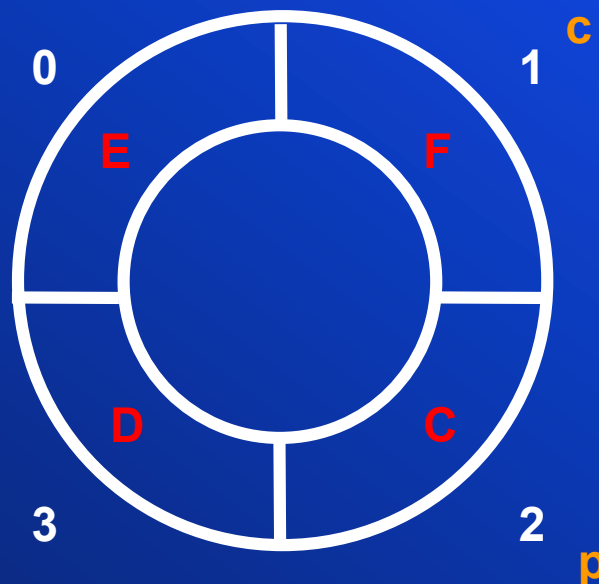
item F

full 3
empty 0
mutex 0

p 1
c 2

item BProcess Producer

```
...  
Repeat  
...  
Produce(item);  
Wait(Empty);  
  
Wait(Mutex);  
buffer[p] = item;  
p = (p+1) % 4;  
Signal(Mutex);  
  
Signal(Full);  
...  
Until ...;
```

Process Consumer

```
...  
Repeat  
...  
Wait(Full);  
  
Wait(Mutex);  
item=buffer[c];  
c = (c+1) % 4;  
Signal(Mutex);  
  
Signal(Empty);  
Consume(item);  
...  
Until ...;
```



Problema do(s) Produtor(es) / Consumidor(es)

