Seguimiento productor consumidor con semáforos

Juan Felipe Muñoz Fernández

```
búfer
                                                   mutex
   #define N 3
1:
2: typedef int semaforo;
   semaforo mutex = 1;
3:
4: semaforo vacias = N;
                                                   vacías
                                                                     elemento
5: semaforo llenas = 0;
6: void productor(void)
7:
                                                   llenas
8: int elemento;
9: while (TRUE) {
     elemento = producir_elemento();
10:
11: wait(&vacias);
12: wait(&mutex);
13: insertar_elemento(elemento);
14: signal(&mutex);
15: signal(&llenas);
16: }
17: }
```

```
#define N 3
  1:
  2: typedef int semaforo;
      semaforo mutex = 1;
→ 3:
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
        elemento = producir_elemento();
  10:
  11: wait(&vacias);
  12: wait(&mutex);
  13: insertar_elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

vacías elemento

llenas

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
   3:
→ 4: semaforo vacias = N;
   5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
        elemento = producir_elemento();
  10:
  11: wait(&vacias);
  12: wait(&mutex);
  13: insertar_elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

1

vacías elemento

llenas

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
  4: semaforo vacias = N;
⇒ 5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
        elemento = producir_elemento();
  10:
  11: wait(&vacias);
  12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

1

vacías elemento
3

llenas
0

	1:	#define N 3
	2:	<pre>typedef int semaforo;</pre>
	3:	<pre>semaforo mutex = 1;</pre>
	4:	semaforo vacias = N;
	5:	<pre>semaforo llenas = 0;</pre>
	6:	<pre>void productor(void)</pre>
	7:	{
	8:	<pre>int elemento;</pre>
	9:	<pre>while(TRUE) {</pre>
>	10:	<pre>elemento = producir_elemento();</pre>
	11:	<pre>wait(&vacias);</pre>
	12:	<pre>wait(&mutex);</pre>
	13:	<pre>insertar_elemento(elemento);</pre>
	14:	<pre>signal(&mutex);</pre>
	15:	<pre>signal(&llenas);</pre>
	16:	}
	17:	}

búfer

elemento

100

mutex

1

vacías

3

llenas

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
     while(TRUE) {
         elemento = producir_elemento();
  10:
→ 11: wait(&vacias);
  12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

1

vacías elemento
2 100

llenas

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
  10: elemento = producir_elemento();
  11: wait(&vacias);
→ 12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

```
mutex búfer

0

vacías elemento
2 100

llenas
0
```

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17:	3

mutex búfer
0 100

vacías elemento
2 100

llenas

1: #define N 3
<pre>2: typedef int semaforo;</pre>
<pre>3: semaforo mutex = 1;</pre>
4: semaforo vacias = N;
<pre>5: semaforo llenas = 0;</pre>
6: <pre>void productor(void)</pre>
7: {
8: int elemento;
9: while (TRUE) {
<pre>10: elemento = producir_elemento();</pre>
<pre>11: wait(&vacias);</pre>
<pre>12: wait(&mutex);</pre>
<pre>13: insertar_elemento(elemento);</pre>
14: signal(&mutex);
<pre>15: signal(&llenas);</pre>
16:}
17· l

mutex búfer

1 100

vacías elemento
2 100

llenas
0

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = $0;$
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	1

mutex	búfer
1	100
vacías	elemento
2	100
llenas	
1	

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	semaforo mutex = 1;
4:	semaforo vacias = N;
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	wait(&vacias);
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	signal(&mutex);
15:	<pre>signal(&llenas);</pre>
16:	}
17:	}

búfer

100

elemento

101

mutex

1

vacías

llenas

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	semaforo vacias = N;
5:	semaforo llenas = $0;$
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10	<pre>elemento = producir_elemento();</pre>
11	<pre>wait(&vacias);</pre>
12	<pre>wait(&mutex);</pre>
13	: insertar_elemento(elemento);
14	signal(&mutex);
15	signal(&llenas);
16	: }
17	• 1

```
mutex búfer

1 100

vacías elemento

1 101
```

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = 0 ;
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while (TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17:	}

mutex búfer

0 100

vacías elemento

1 101

llenas

1

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = $0;$
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	1

mutex búfer
0 100,101

vacías elemento
1 101

llenas
1

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9: 1	while(TRUE) {
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	ì

```
mutex búfer

1 100,101

vacías elemento
1 101
```

1: #define N 3	
2: typedef int semafor	0;
<pre>3: semaforo mutex = 1;</pre>	
4: semaforo vacias = N	·;
5: semaforo llenas = 0	;
6: void productor (void)
7: {	
8: int elemento;	
9: while (TRUE) {	
10: elemento = produc	cir_elemento();
<pre>11: wait(&vacias);</pre>	
<pre>12: wait(&mutex);</pre>	
13: insertar_element	o(elemento);
<pre>14: signal(&mutex);</pre>	
15: signal(&llenas);	
16: }	
17· l	

mutex búfer

1 100,101

vacías elemento

1 101

llenas

1: #define N 3	IIIULEX
<pre>2: typedef int semaforo;</pre>	1
3: semaforo mutex = 1;	
4: semaforo vacias = N;	vacías
<pre>5: semaforo llenas = 0;</pre>	1
6: <pre>void productor(void)</pre>	_
7: {	7 7
8: int elemento;	llenas
9: while (TRUE) {	2
<pre>> 10: elemento = producir_elemento();</pre>	
<pre>11: wait(&vacias);</pre>	
<pre>12: wait(&mutex);</pre>	
<pre>13: insertar_elemento(elemento);</pre>	
<pre>14: signal(&mutex);</pre>	
<pre>15: signal(&llenas);</pre>	
16:}	
17: }	

mutex búfer

1 100,101

vacías elemento

1 102

	1:	#define N 3
	2:	<pre>typedef int semaforo;</pre>
	3:	<pre>semaforo mutex = 1;</pre>
	4:	<pre>semaforo vacias = N;</pre>
	5:	<pre>semaforo llenas = 0;</pre>
	6:	<pre>void productor(void)</pre>
	7:	{
	8:	<pre>int elemento;</pre>
	9:	<pre>while(TRUE) {</pre>
	10:	<pre>elemento = producir_elemento();</pre>
>	11:	<pre>wait(&vacias);</pre>
	12:	<pre>wait(&mutex);</pre>
	13:	<pre>insertar_elemento(elemento);</pre>
	14:	<pre>signal(&mutex);</pre>
	15:	<pre>signal(&llenas);</pre>
	16:	}
	17:	1

mutex búfer

1 100,101

vacías elemento
0 102

llenas

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = 0 ;
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while (TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	ı

mutex búfer
0 100,101

vacías elemento
0 102

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = 0 ;
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while (TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	1

mutex bufer
0 100,101,102

vacias elemento
0 102

llenas
2

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
  4:
      semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
        elemento = producir_elemento();
  10:
  11: wait(&vacias);
  12: wait(&mutex);
  13:
         insertar elemento(elemento);
→ 14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

```
mutex búfer

1 100,101,102

vacías elemento
0 102

llenas
```

```
mutex
      #define N 3
   1:
      typedef int semaforo;
      semaforo mutex = 1;
   3:
   4:
      semaforo vacias = N;
                                                     vacías
   5: semaforo llenas = 0;
                                                       0
   6: void productor(void)
  7:
                                                     llenas
  8: int elemento;
  9: while (TRUE) {
                                                       3
  10:
        elemento = producir_elemento();
  11: wait(&vacias);
  12: wait(&mutex);
  13:
         insertar elemento(elemento);
  14: signal(&mutex);
15:
        signal(&llenas);
  16: }
  17: }
```

Otro ciclo de producción...

búfer

100,101,102

elemento

	1:	#define N 3	
	2:	<pre>typedef int semaforo;</pre>	
	3:	<pre>semaforo mutex = 1;</pre>	
	4:	: semaforo vacias = N;	
	5:	: semaforo llenas = 0;	
	6:	<pre>void productor(void)</pre>	
	7:	{	
	8:	<pre>int elemento;</pre>	
	9:	<pre>while(TRUE) {</pre>	
>	10:	<pre>elemento = producir_elemento();</pre>	
	11:	<pre>wait(&vacias);</pre>	
	12:	<pre>wait(&mutex);</pre>	
	13:	<pre>insertar_elemento(elemento);</pre>	
	14:	<pre>signal(&mutex);</pre>	
	15:	<pre>signal(&llenas);</pre>	
	16:	}	
	17:	1	

mutex búfer

1 100,101,102

vacías elemento
0 104

llenas

```
#define N 3
  1:
      typedef int semaforo;
  3:
      semaforo mutex = 1;
  4:
      semaforo vacias = N;
  5: semaforo llenas = 0;
                                                       -1
  6: void productor(void)
  7:
  8: int elemento;
     while(TRUE) {
                                                        3
  10:
         elemento = producir elemento();
→ 11: wait(&vacias);
  12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

- Se bloquea productor en línea 11 en el semáforo vacías
- Siguiente línea a ejecutar del productor: línea 12
- Se ejecuta consumidor

búfer mutex 100,101,102 vacías elemento 104

llenas

```
void consumidor(void)
2:
3:
      int elemento;
   while (TRUE) {
5:
     wait(&llenas);
6:
   wait(&mutex);
     elemento = quitar_elemento();
7:
8:
   signal(&mutex);
9:
      signal(&vacias);
10:
      consumir elemento(elemento);
11: }
12: }
```

```
búfer
mutex
                    100,101,102
vacías
                      elemento
  -1
llenas
  3
```

```
void consumidor(void)
  1:
  2:
  3:
         int elemento;
      while (TRUE) {
5:
        wait(&llenas);
   6:
      wait(&mutex);
        elemento = quitar_elemento();
  7:
  8:
         signal(&mutex);
   9:
         signal(&vacias);
  10:
         consumir elemento(elemento);
  11: }
  12: }
```

mutex búfer

1 100,101,102

vacías elemento
-1

llenas

```
void consumidor(void)
  2:
  3:
         int elemento;
     while (TRUE) {
  5:
        wait(&llenas);
6:
      wait(&mutex);
      elemento = quitar_elemento();
  7:
  8:
      signal(&mutex);
  9:
         signal(&vacias);
  10:
         consumir elemento(elemento);
  11: }
  12: }
```

mutex búfer

0 100,101,102

vacías elemento
-1

llenas

```
void consumidor(void)
  2:
  3:
         int elemento;
     while (TRUE) {
  5:
        wait(&llenas);
  6:
      wait(&mutex);
      elemento = quitar_elemento();
7:
  8:
      signal(&mutex);
  9:
         signal(&vacias);
         consumir elemento(elemento);
  10:
  11: }
  12: }
```

mutex bufer
0 101,102

vacias elemento
-1 100

llenas

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
  5:
         wait(&llenas);
   6:
      wait(&mutex);
         elemento = quitar_elemento();
   7:
→ 8:
      signal(&mutex);
   9:
         signal(&vacias);
  10:
         consumir elemento(elemento);
  11: }
  12: }
```

mutex búfer

1 101,102

vacías elemento
-1 100

llenas
2

```
búfer
                                                         mutex
                                                                              101,102
                                                         vacías
       void consumidor(void)
                                                                             elemento
   2:
                                                                               100
   3:
          int elemento;
      while(TRUE) {
                                                         llenas
   5:
         wait(&llenas);
   6:
         wait(&mutex);
   7:
          elemento = quitar elemento();
   8:
          signal(&mutex);
→ 9:
          signal(&vacias);
   10:
         consumir elemento(elemento);
  11: }
  12: }
```

- Se desbloquea proceso productor porque se bloqueó en el semáforo vacías
- ¿Se ejecuta?

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
5:
        wait(&llenas);
   6:
      wait(&mutex);
        elemento = quitar_elemento();
  7:
  8:
        signal(&mutex);
  9:
         signal(&vacias);
         consumir_elemento(elemento);
  10:
  11: }
  12: }
```

mutex	búfer
1	101,102
_	
vacías	elemento
0	100
llenas	
1	

```
void consumidor(void)
  2:
  3:
         int elemento;
     while (TRUE) {
  5:
        wait(&llenas);
6:
      wait(&mutex);
      elemento = quitar_elemento();
  7:
  8:
      signal(&mutex);
  9:
         signal(&vacias);
         consumir elemento(elemento);
  10:
  11: }
  12: }
```

mutex búfer
0 101,102

vacías elemento
0 100

llenas
1

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
  5:
        wait(&llenas);
        wait(&mutex);
  6:
      elemento = quitar_elemento();
7:
  8:
      signal(&mutex);
  9:
         signal(&vacias);
         consumir elemento(elemento);
  10:
  11: }
  12: }
```

mutex búfer
0 102

vacías elemento
0 101

llenas

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
  5:
         wait(&llenas);
   6:
        wait(&mutex);
         elemento = quitar_elemento();
   7:
→ 8:
      signal(&mutex);
   9:
         signal(&vacias);
         consumir_elemento(elemento);
  10:
  11: }
  12: }
```

mutex	búfer
1	102
vacías	elemento
0	101
llenas	
1	

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
   5:
         wait(&llenas);
      wait(&mutex);
   6:
         elemento = quitar_elemento();
   7:
  8:
         signal(&mutex);
→ 9:
         signal(&vacias);
         consumir elemento(elemento);
   10:
  11: }
  12: }
```

mutex búfer
1 102

vacías elemento
1 101

llenas
1

```
void consumidor(void)
  2:
  3:
         int elemento;
     while (TRUE) {
5:
        wait(&llenas);
  6:
      wait(&mutex);
      elemento = quitar_elemento();
  7:
  8:
        signal(&mutex);
  9:
         signal(&vacias);
         consumir_elemento(elemento);
  10:
  11: }
  12: }
```

mutex	búfer
1	102
vacías	elemento
1	101
llenas	
0	

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
   5:
        wait(&llenas);
→ 6:
      wait(&mutex);
      elemento = quitar_elemento();
   7:
  8:
      signal(&mutex);
  9:
         signal(&vacias);
         consumir elemento(elemento);
  10:
  11: }
  12: }
```

mutex búfer

0 102

vacías elemento

1 101

llenas

0

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
  5:
        wait(&llenas);
        wait(&mutex);
   6:
→ 7:
      elemento = quitar_elemento();
  8:
      signal(&mutex);
  9:
         signal(&vacias);
  10:
         consumir elemento(elemento);
  11: }
  12: }
```

mutex bufer
0

vacias elemento
1 102

llenas

```
void consumidor(void)
  2:
  3:
         int elemento;
      while (TRUE) {
  5:
         wait(&llenas);
      wait(&mutex);
   6:
         elemento = quitar_elemento();
   7:
→ 8:
      signal(&mutex);
   9:
         signal(&vacias);
  10:
         consumir elemento(elemento);
  11: }
  12: }
```

mutex búfer

1

vacías elemento
 1 102

llenas
 0

```
void consumidor(void)
   2:
   3:
         int elemento;
      while (TRUE) {
   5:
         wait(&llenas);
   6:
      wait(&mutex);
         elemento = quitar_elemento();
   7:
   8:
         signal(&mutex);
→ 9:
         signal(&vacias);
   10: consumir elemento(elemento);
   11: }
   12: }
```

mutex búfer

1

vacías elemento

2 102

llenas

```
vacías
       void consumidor(void)
   2:
   3:
         int elemento;
      while(TRUE) {
                                                        llenas
5:
         wait(&llenas);
   6:
      wait(&mutex);
                                                          -1
   7:
         elemento = quitar_elemento();
  8:
         signal(&mutex);
   9:
         signal(&vacias);
  10:
         consumir elemento(elemento);
  11: }
```

- Se bloquea consumidor en línea 5, en el semáforo llenas
- Siguiente línea a ejecutarse en consumidor: línea 6

12: }

• Se ejecuta productor que había quedado bloqueado en línea 11

mutex búfer

1

vacías elemento

```
mutex
                                                                        búfer
   #define N 3
1:
   typedef int semaforo;
3:
   semaforo mutex = 1;
4: semaforo vacias = N;
                                                   vacías
                                                                       elemento
5: semaforo llenas = 0;
                                                                         104
6: void productor(void)
7:
                                                    llenas
8: int elemento;
9: while (TRUE) {
                                                     -1
10:
      elemento = producir elemento();
11: wait(&vacias);
12: wait(&mutex);
13: insertar elemento(elemento);
14: signal(&mutex);
15: signal(&llenas);
16: }
17: }
```

- Se ejecuta productor
- Siguiente línea a ejecutarse: 12
- Se restaura su estado: variable local elemento que al momento de bloquearse tenía el valor 104

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
  10: elemento = producir_elemento();
  11: wait(&vacias);
→ 12: wait(&mutex);
  13: insertar_elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

0

vacías elemento
2 104

llenas
-1

1: #define N 3
<pre>2: typedef int semaforo;</pre>
<pre>3: semaforo mutex = 1;</pre>
4: semaforo vacias = N;
<pre>5: semaforo llenas = 0;</pre>
6: <pre>void productor(void)</pre>
7: {
8: int elemento;
9: while (TRUE) {
<pre>10: elemento = producir_elemento();</pre>
<pre>11: wait(&vacias);</pre>
<pre>12: wait(&mutex);</pre>
13: insertar_elemento(elemento);
<pre>14: signal(&mutex);</pre>
<pre>15: signal(&llenas);</pre>
16:}
17: }

mutex búfer 0 104

vacías elemento 2 104

llenas

-1

1: #define N 3
<pre>2: typedef int semaforo;</pre>
<pre>3: semaforo mutex = 1;</pre>
4: semaforo vacias = N;
<pre>5: semaforo llenas = 0;</pre>
6: <pre>void productor(void)</pre>
7: {
8: int elemento;
9: while (TRUE) {
<pre>10: elemento = producir_elemento();</pre>
<pre>11: wait(&vacias);</pre>
<pre>12: wait(&mutex);</pre>
<pre>13: insertar_elemento(elemento);</pre>
<pre>14: signal(&mutex);</pre>
<pre>15: signal(&llenas);</pre>
16:}
17· l

```
mutex búfer

1 104

vacías elemento
2 104

llenas
-1
```

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
   3:
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
     while(TRUE) {
  10:
         elemento = producir elemento();
  11: wait(&vacias);
  12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
→ 15: signal(&llenas);
  16: }
  17: }
```

- Se desbloquea proceso consumidor en semáforo llenas
- ¿Se ejecuta?

mutex búfer

1 104

vacías elemento
2 104

llenas

1: #c	define N 3	IIIUCEX
•	<pre>/pedef int semaforo;</pre>	1
3: se	emaforo mutex = 1;	
4: s∈	emaforo vacias = N;	vacías
5: se	emaforo llenas = 0;	2
6: vc	oid productor(void)	2
7: {		
8: in	t elemento;	llenas
9: wh	ile(TRUE){	0
10:	<pre>elemento = producir_elemento();</pre>	
11:	<pre>wait(&vacias);</pre>	
12:	<pre>wait(&mutex);</pre>	
13:	<pre>insertar_elemento(elemento);</pre>	
14:	<pre>signal(&mutex);</pre>	
15:	<pre>signal(&llenas);</pre>	
16:}		
17:		

búfer mutex 1 104 vacías elemento 105

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while (TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	1

```
mutex búfer

1 104

vacías elemento

1 105
```

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = 0 ;
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	ı

mutex búfer

0 104

vacías elemento
1 105

llenas
0

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	1

mutex búfer
0 104,105

vacías elemento
1 105

llenas

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = $0;$
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	ı

```
mutex búfer

1 104,105

vacías elemento
1 105
```

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	while(TRUE) {
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17:	1

mutex	búfer
1	104,105
vacías	elemento
1	105
llenas	

1: #d	lefine N 3	IIIUCEX
•	<pre>pedef int semaforo;</pre>	1
3: se	emaforo mutex = 1;	
4: se	emaforo vacias = N;	vacías
5: se	emaforo llenas = $0;$	1
6: vo	oid productor(void)	*
7: {		
8: int	t elemento;	llenas
9: whi	ile(TRUE){	1
10:	<pre>elemento = producir_elemento();</pre>	
11:	<pre>wait(&vacias);</pre>	
12:	<pre>wait(&mutex);</pre>	
13:	<pre>insertar_elemento(elemento);</pre>	
14:	<pre>signal(&mutex);</pre>	
15:	<pre>signal(&llenas);</pre>	
16:}		
17: }		

búfer mutex 1 104,105 vacías elemento 1 106

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	<pre>semaforo mutex = 1;</pre>
4:	<pre>semaforo vacias = N;</pre>
5:	semaforo llenas = 0 ;
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17.	ì

```
mutex búfer
1 104,105

vacías elemento
0 106
```

llenas

```
#define N 3
   1:
      typedef int semaforo;
      semaforo mutex = 1;
   3:
   4: semaforo vacias = N;
   5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
  10: elemento = producir_elemento();
  11: wait(&vacias);
→ 12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

0 104,105

vacías elemento
0 106

llenas

```
búfer
                                                       mutex
      #define N 3
   1:
                                                         0
                                                                         104,105,106
      typedef int semaforo;
   3:
      semaforo mutex = 1;
      semaforo vacias = N;
                                                       vacías
                                                                          elemento
      semaforo llenas = 0;
                                                         0
                                                                             106
   6: void productor(void)
  7:
                                                       llenas
  8: int elemento;
      while(TRUE) {
                                                         1
         elemento = producir_elemento();
  10:
  11: wait(&vacias);
  12: wait(&mutex);
→ 13:
         insertar elemento(elemento);
  14: signal(&mutex);
  15: signal(&llenas);
  16: }
  17: }
```

- Supongamos que este proceso/hilo sale del procesador en este punto.
- ¿Se ejecuta consumidor?
- Podría pero su siguiente instrucción es wait(&mutex) lo que hace que se bloquee en ese semáforo
- Continuemos con la ejecución del productor

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
  4:
      semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
        elemento = producir_elemento();
  10:
  11: wait(&vacias);
  12: wait(&mutex);
  13:
         insertar_elemento(elemento);
→ 14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

mutex búfer

1 104,105,106

vacías elemento
0 106

llenas
1

```
#define N 3
  1:
      typedef int semaforo;
      semaforo mutex = 1;
  3:
      semaforo vacias = N;
  5: semaforo llenas = 0;
  6: void productor(void)
  7:
  8: int elemento;
  9: while (TRUE) {
  10:
         elemento = producir_elemento();
  11: wait(&vacias);
  12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
15:
        signal(&llenas);
  16:}
  17: }
```

• Otro ciclo de producción...

mutex búfer

1 104,105,106

vacías elemento
0 106

llenas

1:	#define N 3
2:	<pre>typedef int semaforo;</pre>
3:	semaforo mutex = 1;
4:	semaforo vacias = N;
5:	<pre>semaforo llenas = 0;</pre>
6:	<pre>void productor(void)</pre>
7:	{
8:	<pre>int elemento;</pre>
9:	<pre>while(TRUE) {</pre>
10:	<pre>elemento = producir_elemento();</pre>
11:	<pre>wait(&vacias);</pre>
12:	<pre>wait(&mutex);</pre>
13:	<pre>insertar_elemento(elemento);</pre>
14:	<pre>signal(&mutex);</pre>
15:	<pre>signal(&llenas);</pre>
16:	}
17:	}

búfer

104,105,106

elemento

107

mutex

1

vacías

0

llenas

```
#define N 3
  1:
      typedef int semaforo;
  3:
      semaforo mutex = 1;
  4:
      semaforo vacias = N;
      semaforo llenas = 0;
                                                        -1
      void productor(void)
  7:
  8: int elemento;
     while(TRUE) {
  10:
         elemento = producir elemento();
→ 11: wait(&vacias);
  12: wait(&mutex);
  13: insertar elemento(elemento);
  14: signal(&mutex);
  15:
        signal(&llenas);
  16: }
  17: }
```

- Se bloquea productor en línea 11 en el semáforo vacías.
- Siguiente línea a ejecutar del productor: línea 12

búfer mutex 104,105,106 vacías elemento 107

llenas

Seguimiento paso a paso

En ejecución paralela: atentos(a)s

```
void consumidor(void)
1:
    #define N 3
                                 elemento
                                                                                       elemento
    typedef int semaforo;
2:
                                                     2:
3:
    semaforo mutex = 1;
                                                     3:
                                                             int elemento;
4: semaforo vacias = N;
                                                         while(TRUE) {
5: semaforo llenas = 0;
                                                     5:
                                                            wait(&llenas);
6:
    void productor(void)
                                                     6:
                                                            wait(&mutex);
7:
                                                     7:
                                                             elemento = quitar elemento();
8:
   int elemento;
                                                     8:
                                                             signal(&mutex);
9:
   while(TRUE) {
                                                     9:
                                                             signal(&vacias);
10:
       elemento = producir elemento();
                                                     10: }
11:
       wait(&vacias);
                                                     11: }
12:
       wait(&mutex);
13:
       insertar elemento(elemento);
14:
       signal(&mutex);
                                                                                 búfer
                                                         mutex
15:
       signal(&llenas);
16: }
17: }
                                                        vacías
```

- Mutex, vacías y llenas con datos ya que hacen parte del programa principal y son globales a los dos procesos/hilos
- Cada proceso/hilo tiene su variable local elemento.

```
llenas
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      101
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
10:
          elemento = producir elemento();
  11:
         wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

- Se bloquea consumidor en línea 5 en el semáforo llenas.
- Siguiente línea a ejecutar en consumodor: línea 6

```
void consumidor(void)
                                      elemento
   2:
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
      vacías
      llenas
                                Consumidor
         -1
                               Productor
```

```
elemento
      #define N 3
      typedef int semaforo;
  2:
                                     101
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
                                                   5:
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
11:
        wait(&vacias);
  12:
         wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

```
vacías
                                                                   2
Consumidor está bloqueado: no avanza
                                                                llenas
                                                                                            Consumidor
                                                                   -1
                                                                                            Productor
```

void consumidor(void)

int elemento;

wait(&llenas);

signal(&mutex);

signal(&vacias);

elemento = quitar elemento();

búfer

wait(&mutex);

while(TRUE) {

2:

3:

6:

7:

8:

9:

10: }

11: }

mutex

elemento

```
elemento
      #define N 3
      typedef int semaforo;
  2:
                                      101
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Consumidor está bloqueado: no avanza

```
void consumidor(void)
                                      elemento
   2:
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
      vacías
         2
       llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
      #define N 3
      typedef int semaforo;
  2:
                                      101
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Consumidor está bloqueado: no avanza

```
void consumidor(void)
                                      elemento
   2:
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
                                101
         0
      vacías
         2
       llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
      #define N 3
      typedef int semaforo;
  2:
                                      101
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Consumidor está bloqueado: no avanza

```
void consumidor(void)
                                      elemento
   2:
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
                                101
      vacías
         2
       llenas
                                Consumidor
         -1
                                Productor
```

```
#define N 3
                                    elemento
       typedef int semaforo;
  2:
                                       101
  3:
       semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
<del>1</del>5:
          signal(&llenas);
  16: }
  17: }
```

- Se desbloquea consumidor y sigue su ejecución inmediatamente.
- Consumidor estaba bloqueado en el semáforo llenas.
- Siguiente línea a ejecutar en consumidor: línea 6

```
void consumidor(void)
                                    elemento
2:
3:
        int elemento;
    while(TRUE) {
5:
        wait(&llenas);
6:
        wait(&mutex);
7:
        elemento = quitar elemento();
8:
        signal(&mutex);
9:
        signal(&vacias);
10: }
11: }
                            búfer
    mutex
      0
                              101
   vacías
    llenas
                             Consumidor
      0
                             Productor
```

```
#define N 3
                                   elemento
  1:
      typedef int semaforo;
  2:
                                      102
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

```
elemento
    void consumidor(void)
2:
                                      101
3:
        int elemento;
    while(TRUE) {
4:
5:
        wait(&llenas);
6:
        wait(&mutex);
7:
        elemento = quitar elemento();
8:
        signal(&mutex);
9:
        signal(&vacias);
10: }
11: }
                            búfer
    mutex
      0
                             101
   vacías
      2
    llenas
                             Consumidor
      0
                             Productor
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      102
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

```
void consumidor(void)
                                      elemento
   2:
                                         101
   3:
           int elemento;
       while(TRUE) {
   4:
   5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
      vacías
         0
       llenas
                                Consumidor
         0
                                Productor
```

```
elemento
  1:
      #define N 3
                                                           void consumidor(void)
      typedef int semaforo;
                                                       2:
  2:
                                      102
  3:
      semaforo mutex = 1;
                                                       3:
                                                               int elemento;
  4: semaforo vacias = N;
                                                           while(TRUE) {
                                                       4:
  5: semaforo llenas = 0;
                                                       5:
                                                               wait(&llenas);
  6:
      void productor(void)
                                                       6:
                                                               wait(&mutex);
  7:
                                                       7:
                                                               elemento = quitar elemento();
  8:
      int elemento;
                                                       8:
                                                               signal(&mutex);
  9:
      while(TRUE) {
                                                    9:
                                                               signal(&vacias);
  10:
          elemento = producir elemento();
                                                       10: }
  11:
          wait(&vacias);
                                                       11: }
12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
                                                                                   búfer
                                                           mutex
  15:
          signal(&llenas);
  16: }
                                                             0
  17: }
                                                           vacías
                                                             2
```

llenas

0

elemento

101

Consumidor

Productor

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      102
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Se bloquea consumidor en semáforo llenas

```
void consumidor(void)
                                      elemento
   2:
                                         101
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
                                102
      vacías
         2
      llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      102
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Consumidor sigue bloqueado

```
void consumidor(void)
                                      elemento
   2:
                                         101
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
                                102
      vacías
         2
       llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
  1:
       #define N 3
       typedef int semaforo;
  2:
                                       102
  3:
       semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
<del>1</del>5:
          signal(&llenas);
  16: }
  17: }
```

• Se desbloquea consumidor y reanuda ejecución

```
void consumidor(void)
                                   elemento
2:
                                      101
3:
        int elemento;
    while(TRUE) {
5:
        wait(&llenas);
6:
        wait(&mutex);
7:
        elemento = quitar elemento();
8:
        signal(&mutex);
9:
        signal(&vacias);
10: }
11: }
                            búfer
    mutex
      0
                              102
   vacías
      2
    llenas
                             Consumidor
      0
                             Productor
```

```
#define N 3
                                   elemento
  1:
      typedef int semaforo;
  2:
                                      103
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

```
elemento
       void consumidor(void)
   2:
                                         102
   3:
           int elemento;
       while(TRUE) {
   4:
   5:
           wait(&llenas);
   6:
           wait(&mutex);
7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
      vacías
         2
       llenas
                                Consumidor
         0
                                Productor
```

```
elemento
  1:
      #define N 3
                                                           void consumidor(void)
      typedef int semaforo;
                                                       2:
  2:
                                      103
  3:
      semaforo mutex = 1;
                                                       3:
                                                               int elemento;
  4: semaforo vacias = N;
                                                           while(TRUE) {
                                                       4:
  5: semaforo llenas = 0;
                                                       5:
                                                               wait(&llenas);
  6:
      void productor(void)
                                                       6:
                                                               wait(&mutex);
  7:
                                                       7:
                                                               elemento = quitar elemento();
  8:
      int elemento;
                                                    → 8:
                                                               signal(&mutex);
  9:
      while(TRUE) {
                                                       9:
                                                               signal(&vacias);
  10:
          elemento = producir elemento();
                                                       10: }
11:
          wait(&vacias);
                                                       11: }
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
                                                                                   búfer
                                                           mutex
  15:
          signal(&llenas);
  16: }
  17: }
                                                           vacías
                                                           llenas
```

0

elemento

102

Consumidor

Productor

```
elemento
  1:
      #define N 3
                                                           void consumidor(void)
      typedef int semaforo;
                                                       2:
  2:
                                      103
  3:
      semaforo mutex = 1;
                                                       3:
                                                               int elemento;
  4: semaforo vacias = N;
                                                           while(TRUE) {
                                                       4:
  5: semaforo llenas = 0;
                                                       5:
                                                               wait(&llenas);
  6:
      void productor(void)
                                                       6:
                                                               wait(&mutex);
  7:
                                                       7:
                                                               elemento = quitar elemento();
  8:
      int elemento;
                                                       8:
                                                               signal(&mutex);
  9:
      while(TRUE) {
                                                    9:
                                                               signal(&vacias);
  10:
          elemento = producir elemento();
                                                       10: }
  11:
          wait(&vacias);
                                                       11: }
12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
                                                                                   búfer
                                                           mutex
  15:
          signal(&llenas);
  16: }
                                                             0
  17: }
                                                           vacías
                                                             2
                                                           llenas
```

0

elemento

102

Consumidor

Productor

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      103
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
13:
          insertar_elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Se bloquea consumidor

```
void consumidor(void)
                                      elemento
   2:
                                         102
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
                                103
         0
      vacías
         2
      llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      103
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Consumidor sigue bloqueado

```
void consumidor(void)
                                      elemento
   2:
                                         102
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
                                103
      vacías
         2
       llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
  1:
       #define N 3
       typedef int semaforo;
  2:
                                       103
  3:
       semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
<del>1</del>5:
          signal(&llenas);
  16: }
  17: }
```

• Se desbloquea consumidor

```
void consumidor(void)
                                   elemento
2:
                                      102
3:
        int elemento;
    while(TRUE) {
5:
        wait(&llenas);
6:
        wait(&mutex);
7:
        elemento = quitar elemento();
8:
        signal(&mutex);
9:
        signal(&vacias);
10: }
11: }
                            búfer
    mutex
      0
                              103
   vacías
      2
    llenas
                             Consumidor
      0
                             Productor
```

```
#define N 3
                                   elemento
  1:
      typedef int semaforo;
  2:
                                      104
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

```
elemento
       void consumidor(void)
   2:
                                         103
   3:
           int elemento;
       while(TRUE) {
   4:
   5:
           wait(&llenas);
   6:
           wait(&mutex);
7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
      vacías
         2
       llenas
                                Consumidor
         0
                                Productor
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
                                                       2:
  2:
                                      104
  3:
      semaforo mutex = 1;
                                                       3:
                                                               int elemento;
  4: semaforo vacias = N;
                                                           while(TRUE) {
                                                       4:
  5: semaforo llenas = 0;
                                                        5:
  6:
      void productor(void)
                                                        6:
                                                               wait(&mutex);
  7:
                                                        7:
  8:
      int elemento;
                                                    ■ 8:
  9:
      while(TRUE) {
                                                       9:
  10:
          elemento = producir elemento();
                                                       10: }
11:
          wait(&vacias);
                                                       11: }
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
                                                           mutex
  15:
          signal(&llenas);
  16: }
  17: }
                                                           vacías
```

```
void consumidor(void)
                               elemento
                                  103
    wait(&llenas);
    elemento = quitar elemento();
    signal(&mutex);
    signal(&vacias);
                        búfer
llenas
                        Consumidor
  0
                        Productor
```

```
elemento
  1:
      #define N 3
                                                           void consumidor(void)
      typedef int semaforo;
                                                       2:
  2:
                                      104
  3:
      semaforo mutex = 1;
                                                       3:
                                                               int elemento;
  4: semaforo vacias = N;
                                                           while(TRUE) {
                                                       4:
  5: semaforo llenas = 0;
                                                       5:
                                                               wait(&llenas);
  6:
      void productor(void)
                                                       6:
                                                               wait(&mutex);
  7:
                                                       7:
                                                               elemento = quitar elemento();
  8:
      int elemento;
                                                       8:
                                                               signal(&mutex);
  9:
      while(TRUE) {
                                                    9:
                                                               signal(&vacias);
  10:
          elemento = producir elemento();
                                                       10: }
  11:
          wait(&vacias);
                                                       11: }
12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
                                                                                   búfer
                                                           mutex
  15:
          signal(&llenas);
  16: }
                                                             0
  17: }
                                                           vacías
                                                             2
                                                           llenas
```

0

elemento

103

Consumidor

Productor

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      104
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Se bloquea consumidor

```
void consumidor(void)
                                      elemento
   2:
                                         103
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
                                104
      vacías
         2
      llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                      104
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Consumidor sigue bloqueado

```
void consumidor(void)
                                      elemento
   2:
                                         103
   3:
           int elemento;
       while(TRUE) {
5:
           wait(&llenas);
   6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
                                104
      vacías
         2
       llenas
                                Consumidor
         -1
                                Productor
```

```
elemento
  1:
       #define N 3
      typedef int semaforo;
  2:
                                       104
  3:
       semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
  10:
          elemento = producir elemento();
  11:
          wait(&vacias);
  12:
          wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
<del>1</del>5:
          signal(&llenas);
  16: }
  17: }
```

• Se desbloquea consumidor

```
void consumidor(void)
                                      elemento
   2:
                                         103
   3:
           int elemento;
       while(TRUE) {
   5:
           wait(&llenas);
6:
           wait(&mutex);
   7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
                                104
      vacías
         2
       llenas
                                Consumidor
         0
                                Productor
```

```
elemento
  1:
      #define N 3
      typedef int semaforo;
  2:
                                     105
  3:
      semaforo mutex = 1;
  4: semaforo vacias = N;
  5: semaforo llenas = 0;
  6:
      void productor(void)
  7:
  8:
      int elemento;
  9:
      while(TRUE) {
10:
          elemento = producir elemento();
  11:
       wait(&vacias);
  12:
       wait(&mutex);
  13:
          insertar elemento(elemento);
  14:
          signal(&mutex);
  15:
          signal(&llenas);
  16: }
  17: }
```

• Y así sucesivamente por los siglos de los siglos...

```
1: void consumidor (void)
                                      elemento
   2:
                                         104
   3:
           int elemento;
       while(TRUE) {
   4:
   5:
           wait(&llenas);
   6:
           wait(&mutex);
7:
           elemento = quitar elemento();
   8:
           signal(&mutex);
   9:
           signal(&vacias);
   10: }
   11: }
                               búfer
       mutex
         0
      vacías
         2
       llenas
                                Consumidor
         0
                                Productor
```