

```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <time.h>
4
5  #define N 10
6
7  int buffer[N];
8  int counter = 0;
9  int in = 0;
10 int out = 0;
11 int total = 0;
12
13 pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
14 pthread_cond_t prod_cond = PTHREAD_COND_INITIALIZER;
15 pthread_cond_t cons_cond = PTHREAD_COND_INITIALIZER;
16
17 void *producer(void *junk) {
18     while(1) {
19         pthread_mutex_lock(&mutex);
20         if( counter == N )
21             pthread_cond_wait(&prod_cond, &mutex);
22
23         buffer[in] = total++;
24         printf("Produced: %d\n", buffer[in]);
25
26         in = (in + 1) % N;
27         counter++;
28
29         if( counter == 1 )
30             pthread_cond_signal(&cons_cond);
31
32         pthread_mutex_unlock(&mutex);
33     }
34 }
35
36 void *consumer(void *junk) {
37     while(1) {
38         pthread_mutex_lock(&mutex);
39
40         if( counter == 0 )
41             pthread_cond_wait(&cons_cond, &mutex);
42
43         printf("Consumed: %d\n", buffer[out]);
44         out = (out + 1) % N;
45         counter--;
46
47         if( counter == (N-1) )
48             pthread_cond_signal(&prod_cond);
49
50         pthread_mutex_unlock(&mutex);
51     }
52 }
53
54 int main() {
55     pthread_t thread;
56     pthread_create(&thread, NULL, producer, NULL);
57     consumer(NULL);
58 }
```

```
1  #include <stdio.h>
2  #include <pthread.h>
3  #include <semaphore.h>
4  #include <time.h>
5
6  #define N 10
7
8  int buffer[N];
9  int counter = 0;
10 int in = 0;
11 int out = 0;
12 int total = 0;
13
14 pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
15 sem_t semfull;
16 sem_t semempty;
17
18 void *producer(void *junk) {
19     while(1) {
20         sem_wait(&semempty);
21         pthread_mutex_lock(&mutex);
22
23         buffer[in] = total++;
24         printf("Produced: %d\n", buffer[in]);
25
26         in = (in + 1) % N;
27         counter++;
28
29         pthread_mutex_unlock(&mutex);
30         sem_post(&semfull);
31     }
32 }
33
34 void *consumer(void *junk) {
35     while(1) {
36         sem_wait(&semfull);
37         pthread_mutex_lock(&mutex);
38
39         printf("Consumed: %d\n", buffer[out]);
40         out = (out + 1) % N;
41         counter--;
42
43         pthread_mutex_unlock(&mutex);
44         sem_post(&semempty);
45     }
46 }
47
48 int main() {
49     pthread_t thread;
50
51     sem_init(&semfull, 0, 0);
52     sem_init(&semempty, 0, N);
53
54     pthread_create(&thread, NULL, producer, NULL);
55
56     consumer(NULL);
57 }
```