

```
1  #include <pthread.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <unistd.h>
5  #include <errno.h>
6
7  // Producer-Consumer (no semaphores)
8
9  pthread_mutex_t mutex;
10 pthread_cond_t condFull;
11 pthread_cond_t condEmpty;
12 const int NUM_CONSUMER=5, NUM_PRODUCER=1, DIM_BUFFER=10;
13 int buffer[DIM_BUFFER];
14 int head=0;
15
16 void* produce(void* arg) {
17     while(true){
18         sleep(1);
19         pthread_mutex_lock(&mutex);
20         while(head>=DIM_BUFFER){
21             pthread_cond_wait(&condFull,&mutex);
22         }
23
24         int num =rand()%100;
25         buffer[head]=num;
26         head++;
27         printf("produce: %d\n",num);
28         pthread_mutex_unlock(&mutex);
29         pthread_cond_broadcast(&condEmpty);
30     }
31     return NULL;
32 }
33
34 void* consume(void* arg) {
35     while(true){
36         sleep(1);
37         pthread_mutex_lock(&mutex);
38         while(head==0){
39             pthread_cond_wait(&condEmpty,&mutex);
40         }
41
42         int num = buffer[head-1];
43         head--;
44         printf("head: %d consume: %d\n",head,num);
45         pthread_mutex_unlock(&mutex);
46         pthread_cond_broadcast(&condFull);
47     }
48     return NULL;
49 }
50
51 int main(int argc, char* argv[]) {
52     pthread_t thC[NUM_CONSUMER];
53     pthread_t thP[NUM_PRODUCER];
54     pthread_mutex_init(&mutex, NULL);
55     pthread_cond_init(&condFull, NULL);
56     pthread_cond_init(&condEmpty, NULL);
57     for (int i = 0; i < NUM_CONSUMER; i++) {
58         if (pthread_create(&thC[i], NULL, &consume, NULL) != 0) {
59             perror("Failed to create thread");
60         }
61     }
62     for (int i = 0; i < NUM_PRODUCER; i++) {
63         if (pthread_create(&thP[i], NULL, &produce, NULL) != 0) {
64             perror("Failed to create thread");
65         }
66     }
67     for (int i = 0; i < NUM_CONSUMER; i++) {
68         if (pthread_join(thC[i], NULL) != 0) {
69             perror("Failed to join thread");
70         }
71     }
72     for (int i = 0; i < NUM_PRODUCER; i++) {
73         if (pthread_join(thP[i], NULL) != 0) {
74             perror("Failed to join thread");
75         }
76     }
```

```
77     pthread_mutex_destroy(&mutex);  
78     pthread_cond_destroy(&condFull);  
79     pthread_cond_destroy(&condEmpty);  
80     return 0;  
81 }  
82
```