

```
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; // more than one!
13 const int NUM_PRODUCER=2; // more than one!
14 const int DIM_BUFFER=10;
15 int buffer[DIM_BUFFER];
16 int head=0;
17
18 void* produce(void* arg) {
19     while(true){
20         sleep(1);
21         pthread_mutex_lock(&mutex);
22         while(head>=DIM_BUFFER){
23             pthread_cond_wait(&condFull,&mutex);
24         }
25
26         int num =rand()%100;
27         buffer[head]=num;
28         head++;
29         printf("produce: %d\n",num);
30         pthread_mutex_unlock(&mutex);
31         pthread_cond_broadcast(&condEmpty);
32     }
33     return NULL;
34 }
35
36 void* consume(void* arg) {
37     while(true){
38         sleep(1);
39         pthread_mutex_lock(&mutex);
40         while(head==0){
41             pthread_cond_wait(&condEmpty,&mutex);
42         }
43
44         int num = buffer[head-1];
45         head--;
46         printf("head: %d consume: %d\n",head,num);
47         pthread_mutex_unlock(&mutex);
48         pthread_cond_broadcast(&condFull);
49     }
50     return NULL;
51 }
52
53 int main(int argc, char* argv[]) {
54     pthread_t thC[NUM_CONSUMER];
55     pthread_t thP[NUM_PRODUCER];
56     pthread_mutex_init(&mutex, NULL);
57     pthread_cond_init(&condFull, NULL);
58     pthread_cond_init(&condEmpty, NULL);
59     for (int i = 0; i < NUM_CONSUMER; i++) {
60         if (pthread_create(&thC[i], NULL, &consume, NULL) != 0) {
61             perror("Failed to create thread");
62         }
63     }
64     for (int i = 0; i < NUM_PRODUCER; i++) {
65         if (pthread_create(&thP[i], NULL, &produce, NULL) != 0) {
66             perror("Failed to create thread");
67         }
68     }
69     for (int i = 0; i < NUM_CONSUMER; i++) {
70         if (pthread_join(thC[i], NULL) != 0) {
71             perror("Failed to join thread");
72         }
73     }
74     for (int i = 0; i < NUM_PRODUCER; i++) {
75         if (pthread_join(thP[i], NULL) != 0) {
76             perror("Failed to join thread");
77         }
78     }
79 }
```

```
77         }
78     }
79     pthread_mutex_destroy(&mutex);
80     pthread_cond_destroy(&condFull);
81     pthread_cond_destroy(&condEmpty);
82     return 0;
83 }
84
```