

用 semaphore.h 实现，4 个进程的生产者/消费者问题

1. 4 个进程，2 个生产者 2 个消费者

```
ocelot@ubuntu: ~/producer_consumer
ocelot@ubuntu:~/producer_consumer$ ./testk
3 Producing A ...
1 Producing B ...
0 ...Consuming A
2 ...Consuming B
3 Producing C ...
0 ...Consuming C
1 Producing D ...
2 ...Consuming D
3 Producing E ...
0 ...Consuming E
1 Producing F ...
2 ...Consuming F
^C
ocelot@ubuntu:~/producer_consumer$ g++ -Wall -g -o testk test.cpp -lpthread -fopenmp
ocelot@ubuntu:~/producer_consumer$ ./testk
1 Producing A ...
3 ...Consuming A
1 Producing B ...
0 ...Consuming B
1 Producing C ...
2 ...Consuming C
1 Producing D ...
```

2. 4 个进程，1 个生产者 3 个消费者

```
ocelot@ubuntu: ~/producer_consumer
ocelot@ubuntu:~/producer_consumer$ ./testk
1 Producing A ...
3 ...Consuming A
1 Producing B ...
0 ...Consuming B
1 Producing C ...
2 ...Consuming C
1 Producing D ...
2 ...Consuming D
1 Producing E ...
0 ...Consuming E
^C
ocelot@ubuntu:~/producer_consumer$ g++ -Wall -g -o testk test.cpp -lpthread -fopenmp
ocelot@ubuntu:~/producer_consumer$ ./testk
0 Producing A ...
1 ...Consuming A
3 Producing B ...
2 Producing C ...
0 Producing D ...
3 Producing E ...
1 ...Consuming B
2 Producing F ...
1 ...Consuming C
```

3. 4 个进程, 3 个生产者 1 个消费者

```
ocelot@ubuntu: ~/producer_consumer
ocelot@ubuntu:~/producer_consumer$ ./testk
0 Producing A ...
1 ...Consuming A
2 Producing B ...
3 Producing C ...
0 Producing D ...
1 ...Consuming B
2 Producing E ...
3 Producing F ...
0 Producing G ...
1 ...Consuming C
2 Producing H ...
1 ...Consuming D
2 Producing I ...
1 ...Consuming E
2 Producing J ...
1 ...Consuming F
3 Producing K ...
1 ...Consuming G
0 Producing L ...
^C
ocelot@ubuntu:~/producer_consumer$ g++ -Wall -g -o testk test.cpp -lpthread -fopenmp
ocelot@ubuntu:~/producer_consumer$ ./testk
```

4. 4 个进程, 4 个生产者

```
ocelot@ubuntu: ~/producer_consumer
2 Producing E ...
3 Producing F ...
0 Producing G ...
1 ...Consuming C
2 Producing H ...
1 ...Consuming D
2 Producing I ...
1 ...Consuming E
2 Producing J ...
1 ...Consuming F
3 Producing K ...
1 ...Consuming G
0 Producing L ...
^C
ocelot@ubuntu:~/producer_consumer$ g++ -Wall -g -o testk test.cpp -lpthread -fopenmp
ocelot@ubuntu:~/producer_consumer$ ./testk
1 Producing A ...
0 Producing B ...
2 Producing C ...
3 Producing D ...
1 Producing E ...
^C
ocelot@ubuntu:~/producer_consumer$
```