운영체제 및 실습
-생산자_소비자 과제-

학번: 201404377
이름: 진승언

```
for(i = 0; i<P_COUNT; i++){
        //버퍼에 넣을 임의의 값 생성
        input = random()%100;
        usleep(input);
        printf("producer %d add Q %d\n",id, input);

        /**********************  코드 작성 부분  ************************/
        /*  소비자가 버퍼에 접근하지 못하도록 임계영역으로 설정          */
        pthread_mutex_lock(&buffer_lock);

        /*  버퍼가 가득차면 대기(반드시 while문 사용, if문 사용하지 말 것)*/
        while(CQ_count ==10){
                pthread_cond_wait(&buffers , &buffer_lock);              }

        /* input을 버퍼에 넣는다 */
        addQ(input);

        /*              버퍼 안에 item이 있음을 소비자에게 알림          */
        pthread_cond_signal(&items);

        /*  삽입이 끝나면 임계영역을 빠져 나가므로 mutex 락을 해제       */
        pthread_mutex_unlock(&buffer_lock );
        /*****************************************************************/


}
```

<생성자>

1. 지정된 mutext를 lock하는 함수를 이용하여 소비자가 버퍼에 접근하지 못하도록 임계영역으로 설정한다.

2. 버퍼가 최대크기가 10이므로 버퍼크기가 10일 경우를 while문으로 돌려서 wait시킨다. 첫 번째 매개변수는 생산 버퍼 조건변수를 이용하고 두 번째 매개변수는 mutex변수를 이용하면 된다. 해당 쓰레드가 pthread_cond_wait함수를 마친 후에도 대기 조건이 참임을 확인해야 하므로 if문이 아닌 while문 같은 반복문을 조건문으로 사용해야 한다.

3. 큐 버퍼가 꽉 안찬 경우에는 input을 버퍼에 넣어준다.

4. 소비자에게 버퍼안에 item이 있다고 시그널 함수로 알려준다.

5. 삽입이 끝나면 임계영역을 빠져나가므로 unlock을 해주면 된다.

```c
for(i = 0; i<C_COUNT; i++){
        usleep(random()%100);

        /*********************  코드 작성 부분  ************************/

        /*  생산자가 버퍼에 접근하지 못하도록 임계영역으로 설정          */
        pthread_mutex_lock(&buffer_lock);
        /*버퍼가 비어있으면 대기(반드시 while문 사용, if문 사용하지 말 것)*/
        while(CQ_count ==0){
                pthread_cond_wait(&items , &buffer_lock);
        }
        /* item을 버퍼로부터 가져온다            */
        output=getQ();
        /*       버퍼 안에 item을 소비했다고 생산자에게 알림          */
        pthread_cond_signal(&buffers);

        /*     소비가 끝나면 임계영역을 빠져 나가므로 mutex 락을 해제     */

        pthread_mutex_unlock(&buffer_lock);
        /*****************************************************************/

        if(output != -1)
                printf("consumer %d get Q %d\n",id, output);
}
```

<소비자>

=> 생성자와 동일한(비슷한) 방법으로 해주었다. 수정할 점은 소비자는 버퍼가 비어있으면 대기 시켜야하므로 while문을 큐 버퍼 크기 0일 경우로 조건을 바꿔주면 된다. 그리고 조건 변수들을 items로 바꿔주고 signal을 생성자에게 보내주는 것으로 바꿔주면 된다.

<결과화면>

```
u201404377@u201404377:~
u201404377@u201404377:~$ vi producer-consumer.c
u201404377@u201404377:~$ make
gcc -c producer-consumer.c
gcc -o pc producer-consumer.o -lpthread
u201404377@u201404377:~$ ./pc
producer 1 add Q 77
consumer 1 get Q 77
producer 2 add Q 15
consumer 2 get Q 15
producer 3 add Q 93
producer 4 add Q 35
producer 5 add Q 86
consumer 2 get Q 93
consumer 1 get Q 35
producer 1 add Q 92
producer 2 add Q 21
producer 3 add Q 27
producer 4 add Q 90
consumer 1 get Q 86
consumer 2 get Q 92
producer 1 add Q 40
producer 2 add Q 26
producer 5 add Q 59
consumer 1 get Q 21
consumer 2 get Q 27
producer 1 add Q 67
consumer 1 get Q 90
```

```
u201404377@u201404377:~
producer 2 add Q 26
producer 5 add Q 59
consumer 1 get Q 21
consumer 2 get Q 27
producer 1 add Q 67
consumer 1 get Q 90
producer 2 add Q 29
producer 3 add Q 72
producer 4 add Q 36
producer 5 add Q 82
consumer 2 get Q 40
producer 2 add Q 35
producer 1 add Q 23
producer 3 add Q 29
producer 4 add Q 2
producer 5 add Q 22
consumer 1 get Q 26
consumer 2 get Q 59
producer 2 add Q 69
producer 1 add Q 67
producer 4 add Q 11
consumer 2 get Q 67
consumer 1 get Q 29
producer 3 add Q 93
consumer 2 get Q 72
producer 3 add Q 21
consumer 1 get Q 36
```