

운영체제 및 실습
-생산자_소비자 과제-

학번: 201404377

이름: 진승언

```

void *producer(void *args){
    int i;
    int producerID = ++producerCount;
    int item;

    for(i = 0; i < PRODUCE_TIME; i++){
        item = (random() % 100) + producerID;
        usleep(item);

        sem_wait(&produce_ok);//
        sem_wait(&buffer_lock);//
        /*생산 세마포어 변수 produce_ok를 함수를 사용해 1 감소
        /*생산 하는 도중에 소비자나 다른 생산자가 버퍼에 접근하지 못하도록
        버퍼 세마포어 변수 buffer_lock을 함수를 사용해 1 감소*/
        if(insertItem(item) == 0)
            printf("%s[PRODUCER_ERROR]insertItem Failed%s\n", T_RED, T_DEFAULT);
        else
            printf("%sproducer %d add CQ %d%s\n", T_YELLOW, producerID, item, T_D
EFAULT);

        /*생산이 끝나면 소비자나 다른 생산자가 버퍼에 접근할 수 있도록
        버퍼 세마포어 변수 buffer_lock을 함수를 사용해 1 증가*/
        /*소비자가 소비할 수 있도록
        소비 세마포어변수 consume_ok를 함수를 사용해 1 증가*/
        sem_post(&buffer_lock);
        sem_post(&consume_ok);//
    }
}

```

produce_ok는 생산자 세마포어 변수,
consumer_ok는 소비자 세마포어 변수,
buffer_lock은 버퍼 세마포어 변수이다.

<Producer 부분 수정>

sem_wait(&produce_ok)

=> sem_wait함수를 사용해서 생산 세마포어 변수를 1 감소시켜주었다.

sem_wait(&buffer_lock)

=> sem_wait함수를 사용해서 생산 하는 도중에 소비자나 다른 생산자가 버퍼에 접근하지 못하도록 버퍼 세마포어 변수를 1 감소해주었다.

sem_post(&buffer_lock)

=> sem_post함수를 사용해서 생산이 끝나면 소비자나 다른 생산자가 버퍼에 접근할 수 있도록 버퍼 세마포어 변수를 1 증가시켜주었다.

sem_post(&consume_ok)

=> sem_post함수를 사용해서 소비자가 소비를 할 수 있도록 소비 세마포어 변수를 1증가 시켜주었다.

```

void *consumer(void *args){
    int i;
    int consumerID = ++consumerCount;
    int time;

    for(i = 0; i < CONSUME_TIME; i++){
        int item;
        time = (random() % 100) + consumerID;
        usleep(time);

        sem_wait(&consume_ok);//
        sem_wait(&buffer_lock);

        /*소비 세마포어 변수 consume_ok를 함수를 사용해 1 감소
        /*생산자와 동일, 생산자나 다른 소비자가 버퍼에 접근하지 못하도록
        버퍼 세마포어 변수 buffer_lock을 함수를 사용해 1 감소*/
        if((item = takeItem()) == -1)
            printf("%s[CONSUMER_ERROR]takeItem Failed%s\n", T_RED, T_DEFAULT);

        else
            printf("%sconsumer %d take item from CQ%d%s\n", T_GREEN, consumerID,
            item, T_DEFAULT);
        /*생산자와 동일, 생산자나 다른 소비자가 버퍼에 접근할 수 있도록
        버퍼 세마포어 변수 buffer_lock을 함수를 사용해 1 증가*/
        /*소비자가 버퍼 안의 물건을 하나 꺼내 소비했으므로
        생산 세마포어변수 produce_ok를 함수를 사용해 1 증가*/

        sem_post(&buffer_lock);
        sem_post(&produce_ok);
    }
}

```

<Consumer 부분 수정>

sem_wait(&consume_ok)

=> sem_wait함수를 사용해서 소비자 세마포어 변수를 1 감소시켜주었다.

sem_wait(&buffer_ok)

=> sem_wait함수를 사용해서 소비 하는 도중에 소비자나 다른 생산자가 버퍼에 접근하지 못하도록 버퍼 세마포어 변수를 1 감소해주었다.

sem_post(&buffer_lock)

=> sem_post함수를 사용해서 생산이 끝나면 소비자나 다른 생산자가 버퍼에 접근할 수 있도록 버퍼 세마포어 변수를 1 증가시켜주었다.

sem_post(&produce_ok)

=> 소비자가 버퍼 안의 물건을 하나 꺼내 소비했으므로 생산 세마포어 변수를 sem_post함수를 사용해서 1 증가 시켜주었다.

<결과 화면>

```
u201404377@u201404377:~/sema/sem$ ./main
producer 1 add CQ 78
consumer 1 take item from CQ78
producer 2 add CQ 17
consumer 2 take item from CQ17
producer 3 add CQ 96
producer 4 add CQ 39
producer 5 add CQ 91
consumer 1 take item from CQ96
consumer 2 take item from CQ39
producer 1 add CQ 93
producer 2 add CQ 23
producer 3 add CQ 30
producer 5 add CQ 64
consumer 2 take item from CQ91
producer 1 add CQ 41
producer 2 add CQ 28
producer 4 add CQ 94
consumer 1 take item from CQ93
consumer 2 take item from CQ23
producer 3 add CQ 75
producer 5 add CQ 41
consumer 2 take item from CQ30
producer 1 add CQ 69
consumer 1 take item from CQ64
producer 1 add CQ 36
consumer 1 take item from CQ41
consumer 2 take item from CQ28
producer 1 add CQ 3
producer 2 add CQ 69
consumer 1 take item from CQ94
producer 3 add CQ 65
consumer 2 take item from CQ75
producer 4 add CQ 33
consumer 1 take item from CQ41
producer 5 add CQ 28
producer 5 add CQ 29
producer 3 add CQ 73
consumer 2 take item from CQ15
consumer 1 take item from CQ23
producer 4 add CQ 30
consumer 2 take item from CQ34
consumer 1 take item from CQ96
consumer 2 take item from CQ88
consumer 1 take item from CQ29
consumer 2 take item from CQ73
consumer 2 take item from CQ30
u201404377@u201404377:~/sema/sem$ vi semaphore.c
u201404377@u201404377:~/sema/sem$ vi semaphore.c
u201404377@u201404377:~/sema/sem$
```