ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 312E

OPERATING SYSTEMS
ASSIGNMENT 3 REPORT

DATE : 09.06.2021

STUDENT:

NAME : ABDULKADİR

SURNAME : PAZAR

NUMBER : 150180028

SPRING 2021

General Information

There are m news sources (publishers).
There are n subscribers.
I used m binary semaphores for publishers. (pub_sem_arr in the code)
I used n counting semaphores for subscribers (sub_sem_arr in the code)
I used a mutex semaphore for managing access to the circular buffer.
I used a m sized circular buffer to store news.
"write_index" in the publish() pseudocode is a global variable.

Pseudocodes for Threads

read_news()

```
read_news(subscriber index)
    local index = 0
    total news = 0
    for publisher in publishers
        total news += publisher[number of news]
    for(i from 0 to total news)
        P(sub_sem[subscriber index], 1)
        P(mutex, 1)
        Read from buffer
        news[local index].read count++
        if(news[local index].read count = n)
            V(news[local index].pub_sem_id, 1)
        local index = (local index + 1) % m
        V(mutex, 1)
    exit()
```

publish()

```
publish(publisher index)
for(i from 0 to number of news for this publisher)
    P(pub_sem[publisher index], 1)
    P(mutex, 1)
    Generate news
    Write news to news[write index] position
```

```
        news[write_index].publisher_id = pub_sem[sem_index]
        news[write_index].read_count = 0
        write_index = (write_index + 1) % m
        for(all subscribers)
            V(sub_sem, 1)
        V(mutex, 1)
    exit()
```

Example Output

```
Publisher #1 published: News1
Publisher #2 published: News4
Publisher #3 published: News7
Subscriber #1 read: News1
Subscriber #2 read: News1
Subscriber #1 read: News4
Subscriber #2 read: News4
Publisher #1 published: News2
Subscriber #1 read: News7
Publisher #2 published: News5
Subscriber #2 read: News7
Subscriber #1 read: News2
Subscriber #2 read: News2
Publisher #3 published: News8
Subscriber #1 read: News5
Publisher #1 published: News3
Subscriber #2 read: News5
Subscriber #1 read: News8
Publisher #2 published: News6
Subscriber #2 read: News8
Subscriber #1 read: News3
Publisher #3 published: News9
Subscriber #2 read: News3
Subscriber #1 read: News6
Subscriber #2 read: News6
Subscriber #1 read: News9
Subscriber #2 read: News9
```

Example Output

In the example output there are 3 publishers, 2 subscribers, and each publisher has 3 news to publish. As we can see in the example output the code obeys to all given rules. For example publisher 1 waits until 2 readers read News1 to publish News2 and so on. Each subscriber read each news only once.