Prashant Yadav

CS 4348.002 – Program 2 Report

Problem statement –

The assigned task was to implement the Reader/Writer model using pthreads and Linux for the given pseudocode modified to produce only one million output instead of an infinite amount. Include ten readers with one writer. The writer will produce one million of output, which should be shared among reader threads. We will have to synchronize reader and writer thread as they will try to access shared resource at a time.

Approach to solution –

I went through concept of semaphore and tried to understand its implementation in C. Pseudo code given along with question was helpful. I created a shared array of size 1000000 as a shared resource among reader and writer threads. I used pthread_create method to spawn 10 readers and a writer thread. Writer thread is writing to the array at index *end* and reader threads are reading from array at index *start*.

As we will have 10 reader and 1 writer thread trying to access shared array at a time. We need to implement a synchronization mechanism using semaphore such that, it meets below criterion –

- No reader thread should access shared array when writer is writing to it.
- Writer cannot write even if there is a single reader thread reading from shared array.
- Multiple reader threads can read from array at a time.

From the main routine I spawned 10 reader and 1 writer thread as shown below:-

```
pthread_create(&writer, NULL, write, NULL); //Initalize writer pthread.
for(i=0;i<10;i++){
pthread_create(&readers[i], NULL, read, (void *)i+1); //Initialize 10 reader pthreads.
}
```

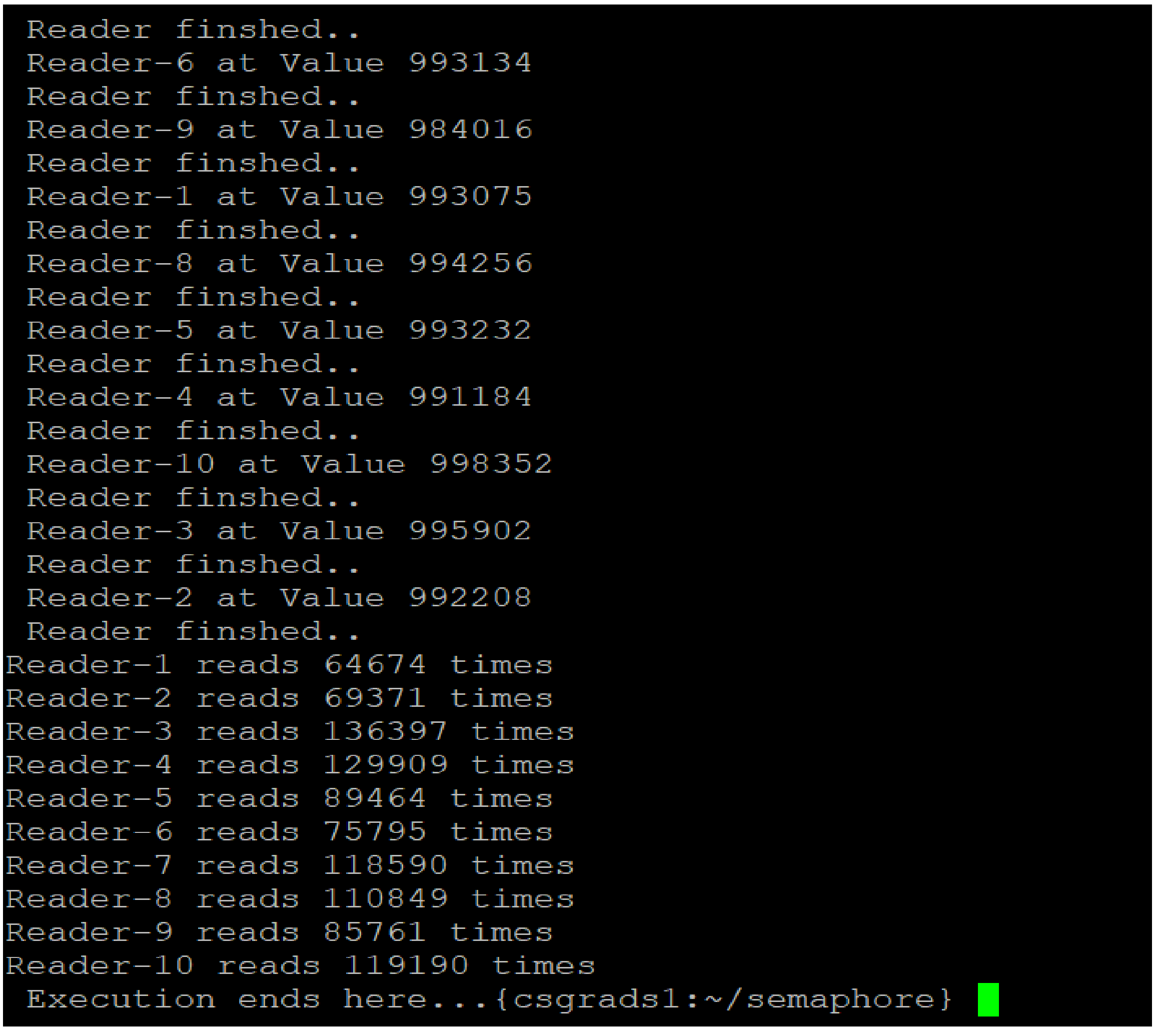
Reader thread start execution from read routine and writer thread starts execution from write routine. I used semaphore z, x, rsem in read routine to allow multiple readers to read from shared array while wsem and y in write routine to allow only one writer thread to write to shared array.

Solution description –

The assigned task was to achieve synchronization between 10 readers and 1 writer thread. I was able to achieve that using semaphores. I saved my code to file program2.c and ran it using command *gcc program2.c -pthread.* Below are the console logs after executing the code.

Screenshot below displays count of reads done by each thread i. Here is the table showing each thread and its read count –

Thread number	**Read count**
Reader-1	64674
Reader-2	69371
Reader-3	136397
Reader-4	129909
Reader-5	89464
Reader-6	75795
Reader-7	118590
Reader-8	110849
Reader-9	85761
Reader-10	119190
Total -	**1000000**



```
 Reader finshed..
 Reader-6 at Value 993134
 Reader finshed..
 Reader-9 at Value 984016
 Reader finshed..
 Reader-1 at Value 993075
 Reader finshed..
 Reader-8 at Value 994256
 Reader finshed..
 Reader-5 at Value 993232
 Reader finshed..
 Reader-4 at Value 991184
 Reader finshed..
 Reader-10 at Value 998352
 Reader finshed..
 Reader-3 at Value 995902
 Reader finshed..
 Reader-2 at Value 992208
 Reader finshed..
Reader-1 reads 64674 times
Reader-2 reads 69371 times
Reader-3 reads 136397 times
Reader-4 reads 129909 times
Reader-5 reads 89464 times
Reader-6 reads 75795 times
Reader-7 reads 118590 times
Reader-8 reads 110849 times
Reader-9 reads 85761 times
Reader-10 reads 119190 times
 Execution ends here...{csgrads1:~/semaphore}
```