Assignment 4B

Roll no. 33331

Program :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <string.h>
#include <pthread.h>
#include <semaphore.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/shm.h>

sem_t mutex; sem_t writeSemaphore; int
readCount = 0, w, r, writer_count = 0;


void *writer_thr(void *temp)
{ int id = *((int *)temp); printf("\nWriter %d is waiting
      for writing...\n", id);

      int sem_val;
      sem_getvalue(&writeSemaphore, &sem_val);

      if(sem_val == 0)
      { printf("WriteSemaphore is locked by readers.\n");
      }
      else
      { printf("WriteSemaphore is unlocked.\n");
      }

      sem_wait(&writeSemaphore); //    locked / deceremented the
```

```
semaphore printf("Writer [ %d ] is writing...\n",
      id); sem_post(&writeSemaphore);
      printf("Writer [ %d ] is leaving the critical section...\n",
id); writer_count++;
      pthread_exit(NULL);
}
void *reader_thr(void *temp)
{
      int id = *((int *) temp); printf("\nReader %d wants to
      read the shared resource /
data...\n", id);
      sem_wait(&mutex);     //    locking the mutex
      readCount++; if(readCount == 1)
      { printf("This is the first reader; writeSemaphore is
locked.\n");
            sem_wait(&writeSemaphore);
      }
      sem_post(&mutex);     //    unlocking the mutex
      printf("Reader %d is reading...\n", id);
      printf("Reader %d is leaving...\n", id);
      sleep(3);
      sem_wait(&mutex);
      readCount--;          //    decrementing the reader
      if(readCount == 0)
      { printf("This is the last reader; writeSemaphore is
unlocked.\n");
            sem_post(&writeSemaphore);
      } sem_post(&mutex);
      pthread_exit(NULL);
}
int main()
{
```

```
      long int i; sem_init(&mutex, 0, 1);
      sem_init(&writeSemaphore, 0, 1);
      pthread_t reader[100], writer[100];

      printf("\nEnter number of readers : ");
      scanf("%d", &r);
      printf("\nEnter number of writers : ");
      scanf("%d", &w);

      int reader_ids[r];
      int writer_ids[w];


      for(i = 0; i < r; i++)
      { reader_ids[i] = i+1; sleep(1);
           pthread_create(&reader[i], NULL, reader_thr,
&reader_ids[i]);

      }

      for(i = 0; i < w; i++)
      { writer_ids[i] = i+1; pthread_create(&writer[i],
NULL, writer_thr, &writer_ids[i]);
      }


      {/**
      for (i = 1; i <= r; i++) {
         int *reader_id = malloc(sizeof(int));
         *reader_id = i; sleep(1); // Add a delay before starting
each reader thread pthread_create(&reader[i], NULL, reader_thr,
(void *)reader_id);
    }
    for (i = 1; i <= w; i++) {
         int *writer_id = malloc(sizeof(int));
         *writer_id = i; pthread_create(&writer[i], NULL,
writer_thr, (void *)writer_id);
    }

      */}
```

```
      for(int i = 0; i < r; i++)
            pthread_join(reader[i], NULL);

      for(int i = 0; i < w; i++)
            pthread_join(writer[i], NULL);

      sem_destroy(&writeSemaphore);

      sem_destroy(&mutex); printf("Total writers :

      %d\n", writer_count);

      pthread_exit(NULL);
      return 0;
}
```

Output

Test Case 1: Readers=3,Writers=3

```
Enter number of readers : 3

Enter number of writers : 3

Reader 1 wants to read the shared resource / data...
This is the first reader; writeSemaphore is locked.
Reader 1 is reading...
Reader 1 is leaving...

Reader 2 wants to read the shared resource / data...
Reader 2 is reading...
Reader 2 is leaving...

Writer 1 is waiting for writing...
WriteSemaphore is locked by readers.

Reader 3 wants to read the shared resource / data...
Reader 3 is reading...
Reader 3 is leaving...

Writer 3 is waiting for writing...
WriteSemaphore is locked by readers.
```

```
Writer 2 is waiting for writing...
WriteSemaphore is locked by readers.
This is the last reader; writeSemaphore is unlocked.
Writer [ 1 ] is writing...
Writer [ 1 ] is leaving the critical section... Writer
[ 3 ] is writing...
Writer [ 3 ] is leaving the critical section... Writer
[ 2 ] is writing...
Writer [ 2 ] is leaving the critical section...
Total writers : 3
```

Test Case 2: Readers=5,Writers=5

```
Enter number of readers : 5

Enter number of writers : 5
Reader 1 wants to read the shared resource / data...
This is the first reader; writeSemaphore is locked.
Reader 1 is reading...
Reader 1 is leaving...

Reader 2 wants to read the shared resource / data...
Reader 2 is reading...
Reader 2 is leaving...

Reader 3 wants to read the shared resource / data...
Reader 3 is reading...
Reader 3 is leaving...

Reader 4 wants to read the shared resource / data...
Reader 4 is reading...
Reader 4 is leaving...

Reader 5 wants to read the shared resource / data...
Reader 5 is reading...
Reader 5 is leaving...

Writer 1 is waiting for writing...
WriteSemaphore is locked by readers.
```

```
Writer 3 is waiting for writing...
WriteSemaphore is locked by readers.

Writer 4 is waiting for writing...

Writer 2 is waiting for writing...
WriteSemaphore is locked by readers.
WriteSemaphore is locked by readers.

Writer 5 is waiting for writing...
WriteSemaphore is locked by readers.
This is the last reader; writeSemaphore is unlocked.
Writer [ 1 ] is writing...
Writer [ 1 ] is leaving the critical section... Writer
[ 3 ] is writing...
Writer [ 3 ] is leaving the critical section... Writer
[ 2 ] is writing...
Writer [ 2 ] is leaving the critical section... Writer
[ 4 ] is writing...
Writer [ 4 ] is leaving the critical section... Writer
[ 5 ] is writing...
Writer [ 5 ] is leaving the critical section...
Total writers : 5
```