

```

#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>
#include <semaphore.h>

void *reader(void *arg);
void *writer(void *arg);

sem_t *resource;
pthread_mutex_t rmutex;
int readcount;

int main() {

    int res,i,sizeR,sizeW,data;
    pthread_t *threadR,*threadW;

    data=rand()%100;
    //Accepting Number of Readers and Writers
    printf("\nNumber of Readers: ");
    scanf("%d",&sizeR);

    printf("Number of Writers: ");
    scanf("%d",&sizeW);
    printf("\n");

    threadR=(pthread_t*)malloc(sizeof(pthread_t)*sizeR);
    threadW=(pthread_t*)malloc(sizeof(pthread_t)*sizeW);

    //Creating semaphores
    resource=sem_open("/resource", O_CREAT, 0644, 1);
    pthread_mutex_init(&rmutex, NULL);
    readcount=0;

    //Function Calls
    for(i=0;i<sizeR;i++) {
        res=pthread_create(&threadR[i], NULL, reader, &data);
    }
    for(i=0;i<sizeW;i++) {
        res=pthread_create(&threadW[i], NULL, writer, &data);
    }
    for(i=0;i<sizeR;i++)
        res=pthread_join(threadR[i], NULL);
    for(i=0;i<sizeW;i++)
        res=pthread_join(threadW[i], NULL);

```

```

//destroying the semaphores
sem_close(resource);
pthread_mutex_destroy(&mutex);

exit(EXIT_SUCCESS);
}

void *reader(void *arg) {

while(1) {
    //<Enter Critical Section>
    pthread_mutex_lock(&mutex);
    readcount++;
    if(readcount==1) {
        sem_wait(resource);
    }
    //<Exit Critical Section>
    pthread_mutex_unlock(&mutex);

    //Reading
    printf("\n=====\\n");
    printf("\nThread ID      : %d\\nReading data as: %d\\n",
(int)pthread_self()%10,*(int*)arg);
    printf("\n=====\\n");

    //<Enter Critical Section>
    pthread_mutex_lock(&mutex);
    readcount--;
    if(readcount==0) {
        sem_post(resource);
    }
    //<Exit Critical Section>
    pthread_mutex_unlock(&mutex);
    sleep(5);
}
pthread_exit(NULL);
}

void *writer(void *arg) {

while(1) {
    //<Enter Critical Section>
    sem_wait(resource);

    //Writing
    printf("\n=====\\n");
    printf("\nThread ID      : %d\\nData is          : %d\\n",
(int)pthread_self()%10,*(int*)arg);

```

```

        *(int*)arg=rand()%100;
        printf("Writing data as: %d\n",*(int*)arg);
        printf("\n=====\\n");

        //<Exit Critical Section>
        sem_post(resource);
        sleep(5);
    }
    pthread_exit(NULL);
}

```

Output:

```

Someshwars-MacBook-Pro:Assignment someshwargaikwad$ atom .
Someshwars-MacBook-Pro:Assignment someshwargaikwad$ ./a.out

```

```

Number of Readers: 3
Number of Writers: 2

```

```

=====

```

```

Thread ID      : 4
Data is        : 7
Writing data as: 49

```

```

=====

```

```

=====

```

```

Thread ID      : 6
Reading data as: 49

```

```

=====

```

```

=====

```

```

Thread ID      : 8
Reading data as: 49

```

```

=====

```

```

=====

```

```

Thread ID      : 0
Data is        : 49
Writing data as: 73

```

=====

=====

Thread ID : 2
Reading data as: 73

=====

=====

Thread ID : 4
Data is : 73
Writing data as: 58

=====

=====

Thread ID : 0
Data is : 58
Writing data as: 30

=====

=====

Thread ID : 2
Reading data as: 30

=====

=====

Thread ID : 6
Reading data as: 30

=====

=====

Thread ID : 8
Reading data as: 30

=====

=====

Thread ID : 4
Data is : 30
Writing data as: 72

=====

=====

Thread ID : 0
Data is : 72
Writing data as: 44

=====

=====

Thread ID : 2
Reading data as: 44

=====

=====

Thread ID : 6
Reading data as: 44

=====

=====

Thread ID : 8
Reading data as: 44

=====

=====

Thread ID : 4
Data is : 44
Writing data as: 78

=====

=====

Thread ID : 6
Reading data as: 78

=====

=====

Thread ID : 2
Reading data as: 78

=====

=====

Thread ID : 8
Reading data as: 78

=====

=====

Thread ID : 0
Data is : 78
Writing data as: 23

=====

=====

Thread ID : 4
Data is : 23
Writing data as: 9

=====

=====

Thread ID : 8
Reading data as: 9

=====

=====

Thread ID : 0
Data is : 9
Writing data as: 40

=====

=====

Thread ID : 2
Reading data as: 40

=====

=====

Thread ID : 6
Reading data as: 40

=====

=====

Thread ID : 4
Data is : 40
Writing data as: 65

=====

=====

Thread ID : 8
Reading data as: 65

=====

Thread ID : 2
Reading data as: 65

=====

=====

=====

Thread ID : 6
Reading data as: 65

=====

=====

Thread ID : 0
Data is : 65
Writing data as: 92

=====

=====

Thread ID : 0
Data is : 92
Writing data as: 42

=====

=====

Thread ID : 8
Reading data as: 42

=====

=====

Thread ID : 2
Reading data as: 42

=====

=====

Thread ID : 6
Reading data as: 42

=====

=====

Thread ID : 4
Data is : 42
Writing data as: 87

=====

=====

Thread ID : 0
Data is : 87
Writing data as: 3

=====

=====

Thread ID : 8
Reading data as: 3

=====

=====

Thread ID : 4
Data is : 3
Writing data as: 27

=====

=====

Thread ID : 2
Reading data as: 27

=====

=====

Thread ID : 6
Reading data as: 27

=====