Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 41

OS Assignment-2

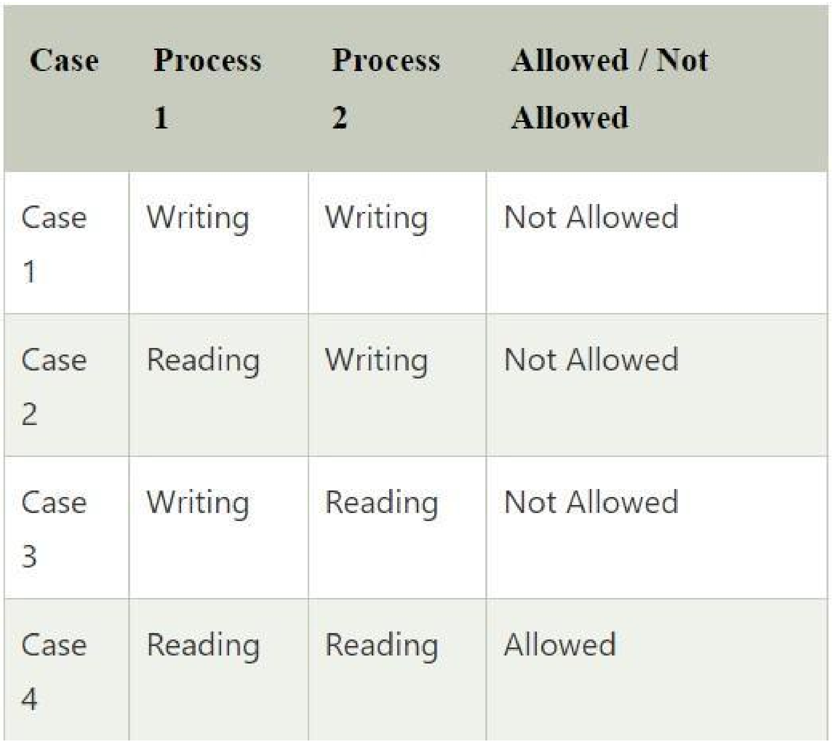
Explain Reader and writer problem. Write a C/C++ program to solve reader writer problem using semaphore.

The readers-writers problem is a classical problem of process synchronization, it relates to a data set such as a file that is shared between more than one process at a time. Among these various processes, some are Readers - which can only read the data set; they do not perform any updates, some are Writers - can both read and write in the data sets.

The readers-writers problem is used for managing synchronization among various reader and writer process so that there are no problems with the data sets, i.e. no inconsistency is generated.

If two or more than two readers want to access the file at the same point in time there will be no problem. However, in other situations like when two writers or one reader and one writer wants to access the file at the same point of time, there may occur some problems, hence the task is to design the code in such a manner that if one reader is reading then no writer is allowed to update at the same point of time, similarly, if one writer is writing no reader is allowed to read the file at that point of time and if one writer is updating a file other writers should not be allowed to update the file at the same point of time. However, multiple readers can access the object at the same time.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 41
OS Assignment-2



Case	Process 1	Process 2	Allowed / Not Allowed
Case 1	Writing	Writing	Not Allowed
Case 2	Reading	Writing	Not Allowed
Case 3	Writing	Reading	Not Allowed
Case 4	Reading	Reading	Allowed

The solution of readers and writers can be implemented using binary semaphores in C :

```c
#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#include <stdlib.h>

#define MAX_READERS 5

sem_t mutex, db; // mutex→RC ; db → database

int read_count = 0;
```

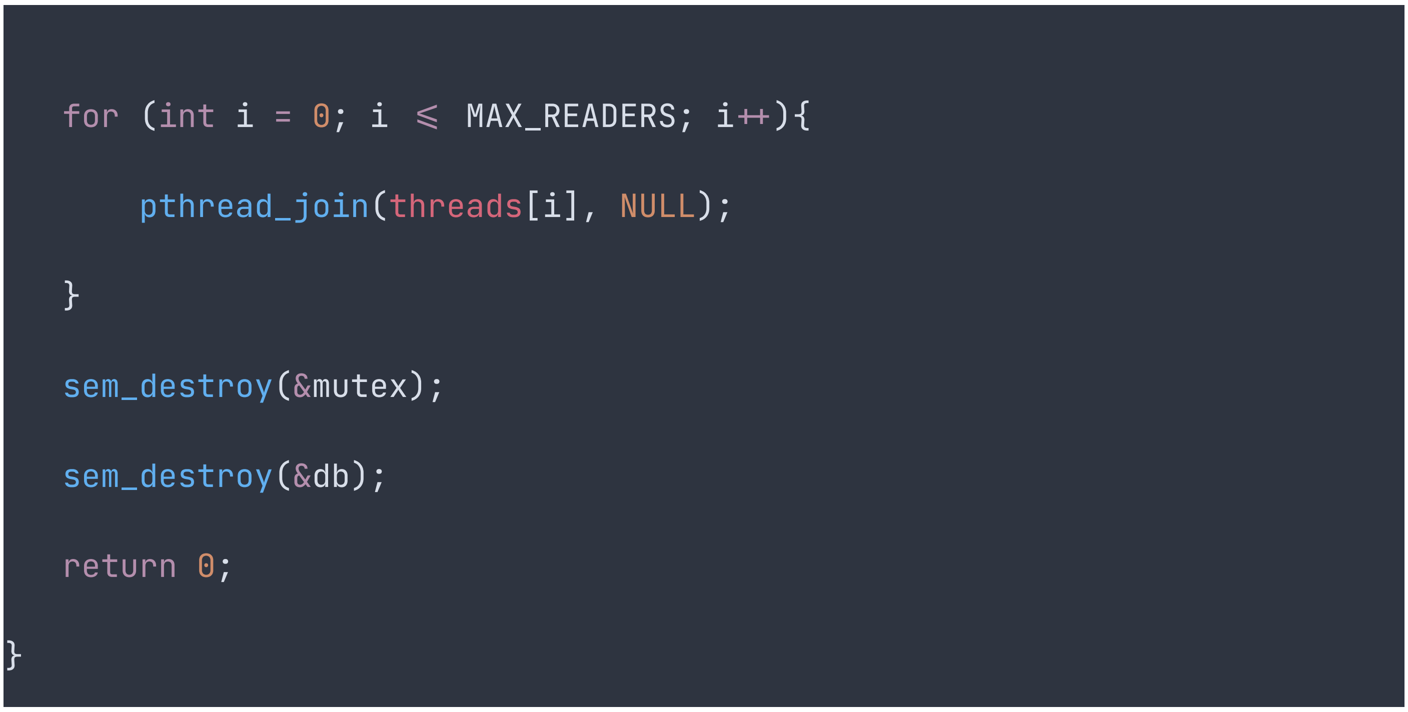
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 41
OS Assignment-2

```c
void* writer(void* arg) {

   int id = ((int) arg);

   sem_wait(&db);

   printf("Writer %d is writing.\n", id);

   sem_post(&db);

   pthread_exit(NULL);

}

void* reader(void* arg) {

   int id = ((int) arg);

   sem_wait(&mutex);

   read_count++;

   printf("Reader %d has entered db. [RC= %d]\n", id, read_count);

   // First Reader takes db access

   if (read_count == 1) {

       sem_wait(&db);

   }

   sem_post(&mutex);

   printf("Reader %d is reading.\n", id);

   sleep(1);

   sem_wait(&mutex);
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 41
OS Assignment-2

```c
    read_count--;

    printf("Reader %d has exited db. [RC= %d]\n", id, read_count);

    // Last Reader releases db

    if (read_count == 0) {

        sem_post(&db);

    }

    sem_post(&mutex);

    pthread_exit(NULL);

}

int main() {

    pthread_t threads[MAX_READERS + 1];

    int ids[MAX_READERS + 1];

    sem_init(&mutex, 0, 1);

    sem_init(&db, 0, 1);

    ids[0] = 0;

    pthread_create(&threads[0], NULL, writer, (void*) &ids[0]);

    for (int i = 1; i <= MAX_READERS; i++){

        ids[i] = i;

        pthread_create(&threads[i], NULL, reader, (void*) &ids[i]);

    }
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA   Batch - 41
OS Assignment-2



```c
    for (int i = 0; i <= MAX_READERS; i++){
        pthread_join(threads[i], NULL);
    }
    sem_destroy(&mutex);
    sem_destroy(&db);
    return 0;
}
```