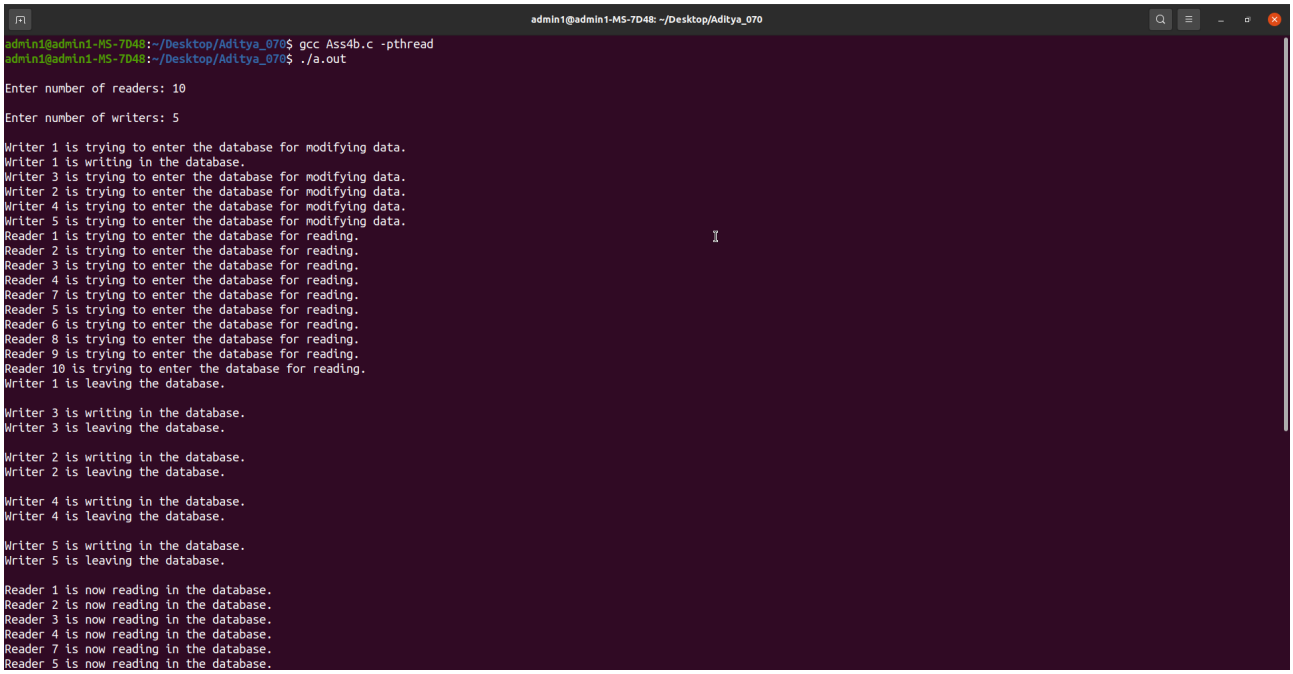
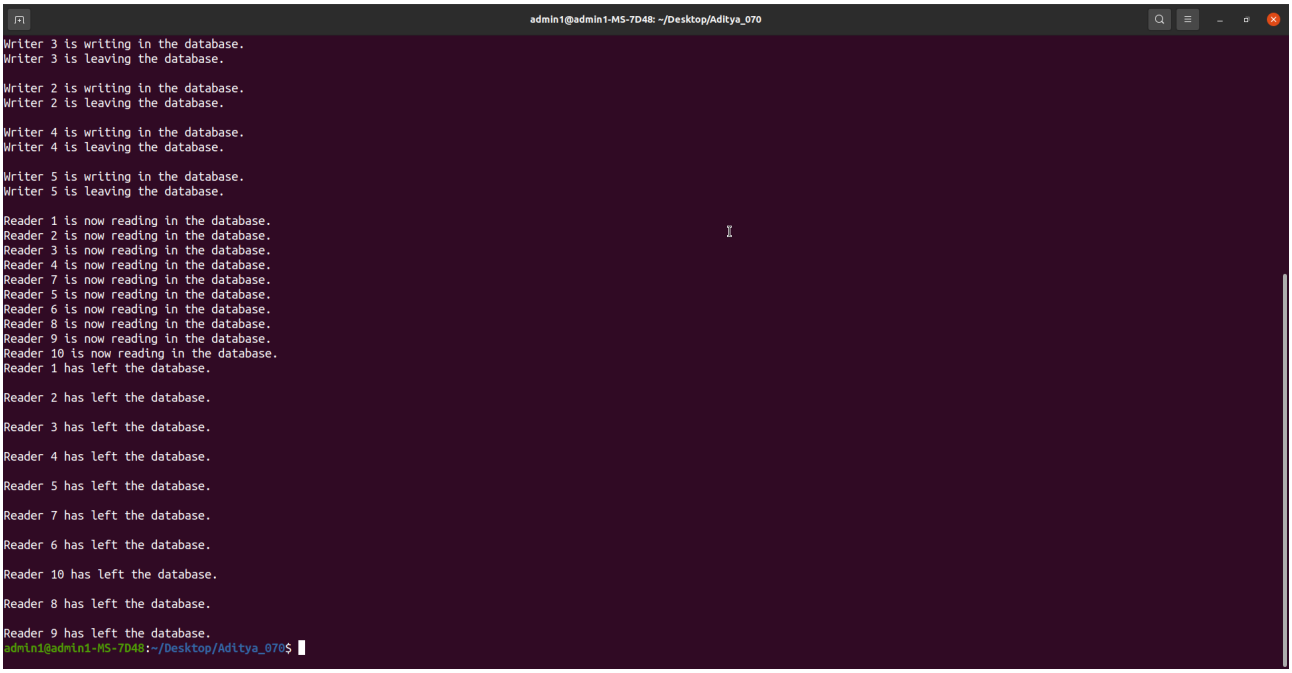
Reader- Writer Problem

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

void *writer_thr(void *arg);
void *reader_thr(void *arg);
sem_t mutex;
sem_t wrt;
int readcount = 0, nwt, nrd;

int main() {
    long i;
    sem_init(&mutex, 0, 1);
    sem_init(&wrt, 0, 1);
    pthread_t reader[100], writer[100];

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

    // Create writer threads
    for (i = 0; i < nwt; i++) {
        long *arg = malloc(sizeof(*arg)); // Allocate memory for argument
        *arg = i + 1; // Pass the writer number
        pthread_create(&writer[i], NULL, writer_thr, arg);
    }

    // Create reader threads
    for (i = 0; i < nrd; i++) {
        long *arg = malloc(sizeof(*arg)); // Allocate memory for argument
        *arg = i + 1; // Pass the reader number
        pthread_create(&reader[i], NULL, reader_thr, arg);
    }

    // Join writer threads
    for (i = 0; i < nwt; i++) {
        pthread_join(writer[i], NULL);
    }

    // Join reader threads
    for (i = 0; i < nrd; i++) {
        pthread_join(reader[i], NULL);
    }

    sem_destroy(&wrt);
    sem_destroy(&mutex);
    return 0;
}

void *reader_thr(void *arg) {
    long temp = *(long *)arg; // Get the reader number
    free(arg); // Free allocated memory
    printf("\nReader %ld is trying to enter the database for reading.", temp);

    sem_wait(&mutex);
    readcount++;
    if (readcount == 1) {
```

```
        sem_wait(&wrt);
    }
    sem_post(&mutex);

    printf("\nReader %ld is now reading in the database.", temp);
    sleep(3); // Simulate reading

    sem_wait(&mutex);
    readcount--;
    if (readcount == 0) {
        sem_post(&wrt);
    }
    sem_post(&mutex);

    printf("\nReader %ld has left the database.\n", temp);
}

void *writer_thr(void *arg) {
    long temp = *(long *)arg; // Get the writer number
    free(arg); // Free allocated memory
    printf("\nWriter %ld is trying to enter the database for modifying data.", temp);

    sem_wait(&wrt);
    printf("\nWriter %ld is writing in the database.", temp);
    sleep(3); // Simulate writing
    printf("\nWriter %ld is leaving the database.\n", temp);

    sem_post(&wrt);
}
```

Output:




admin1@admin1-MS-7D48: ~/Desktop/Aditya_070
Writer 3 is writing in the database.
Writer 3 is leaving the database.
Writer 2 is writing in the database.
Writer 2 is leaving the database.
Writer 4 is writing in the database.
Writer 4 is leaving the database.
Writer 5 is writing in the database.
Writer 5 is leaving the database.
Reader 1 is now reading in the database.
Reader 2 is now reading in the database.
Reader 3 is now reading in the database.
Reader 4 is now reading in the database.
Reader 7 is now reading in the database.
Reader 5 is now reading in the database.
Reader 6 is now reading in the database.
Reader 8 is now reading in the database.
Reader 9 is now reading in the database.
Reader 10 is now reading in the database.
Reader 1 has left the database.
Reader 2 has left the database.
Reader 3 has left the database.
Reader 4 has left the database.
Reader 5 has left the database.
Reader 7 has left the database.
Reader 6 has left the database.
Reader 10 has left the database.
Reader 8 has left the database.
Reader 9 has left the database.
admin1@admin1-MS-7D48:~/Desktop/Aditya_070$