Name:Sujal Santosh Porte
PRN:122B1B227

Assignment 4

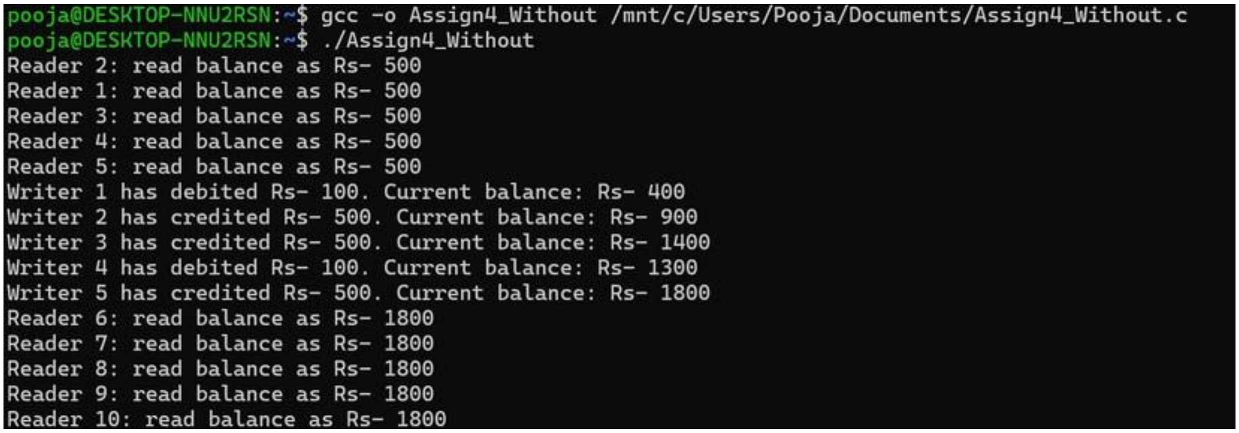
Write a program using pthreads to demonstrate the reader writer synchronization problem. Implement appropriate synchronization. Show the different results with and without synchronization

1) Without Using Synchronization

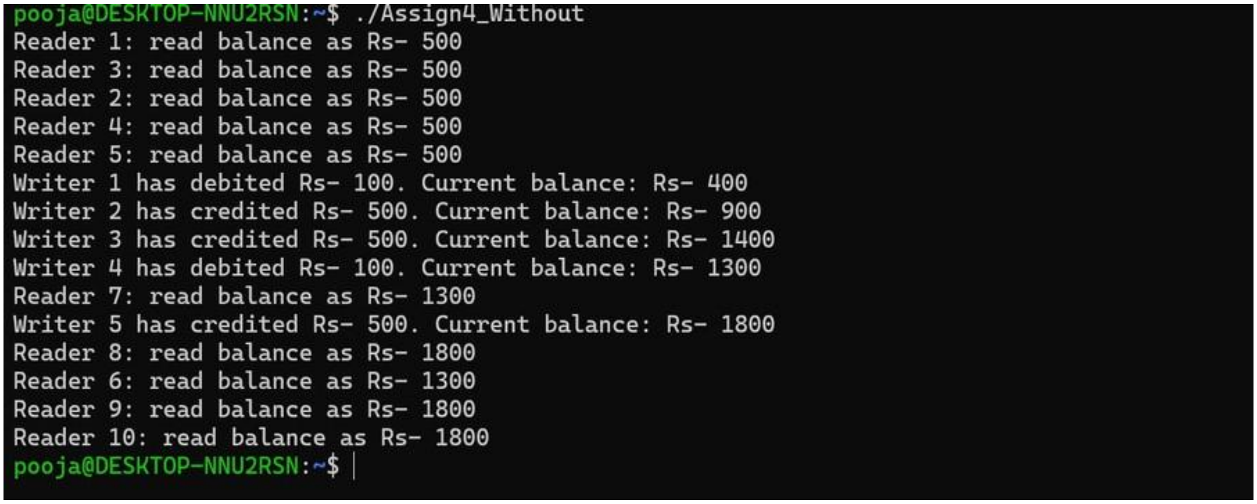
Code:

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

int balance = 500;

void *credit_writer(void *wno) {
    int amount = 500;
    balance = balance + amount;
    printf("Writer %d has credited Rs- %d. Current balance: Rs- %d\n", *((int *)wno), amount,
balance);
}

void *debit_writer(void *wno) {
    int amount = 100;
    balance = balance - amount;
    printf("Writer %d has debited Rs- %d. Current balance: Rs- %d\n", *((int *)wno), amount,
balance);
}

void *reader(void *rno) {
    printf("Reader %d: read balance as Rs- %d\n", *((int *)rno), balance);
}

int main() {
    pthread_t reader_thread_id[10], writer_thread_id[5];
    char c[] = {'D', 'C', 'C', 'D', 'C'};
    int a[10] = {1,2,3,4,5,6,7,8,9,10};
    for(int i = 0; i < 5; i++) {
        pthread_create(&reader_thread_id[i], NULL, reader, &a[i]);
    }
    for(int i = 0; i < 5; i++) {
        if(c[i] == 'C')
            pthread_create(&writer_thread_id[i], NULL, credit_writer, &a[i]);
        else
            pthread_create(&writer_thread_id[i], NULL, debit_writer, &a[i]);
    }
    for(int i = 5; i < 10; i++) {
        pthread_create(&reader_thread_id[i], NULL, reader, &a[i]);
    }

    for(int i = 0; i < 10; i++) {
        pthread_join(reader_thread_id[i], NULL);
    }
    for(int i = 0; i < 5; i++) {
        pthread_join(writer_thread_id[i], NULL);
    }

    return 0;
}
```

OUTPUT:



```
pooja@DESKTOP-NNU2RSN:~$ gcc -o Assign4_Without /mnt/c/Users/Pooja/Documents/Assign4_Without.c
pooja@DESKTOP-NNU2RSN:~$ ./Assign4_Without
Reader 2: read balance as Rs- 500
Reader 1: read balance as Rs- 500
Reader 3: read balance as Rs- 500
Reader 4: read balance as Rs- 500
Reader 5: read balance as Rs- 500
Writer 1 has debited Rs- 100. Current balance: Rs- 400
Writer 2 has credited Rs- 500. Current balance: Rs- 900
Writer 3 has credited Rs- 500. Current balance: Rs- 1400
Writer 4 has debited Rs- 100. Current balance: Rs- 1300
Writer 5 has credited Rs- 500. Current balance: Rs- 1800
Reader 6: read balance as Rs- 1800
Reader 7: read balance as Rs- 1800
Reader 8: read balance as Rs- 1800
Reader 9: read balance as Rs- 1800
Reader 10: read balance as Rs- 1800
```



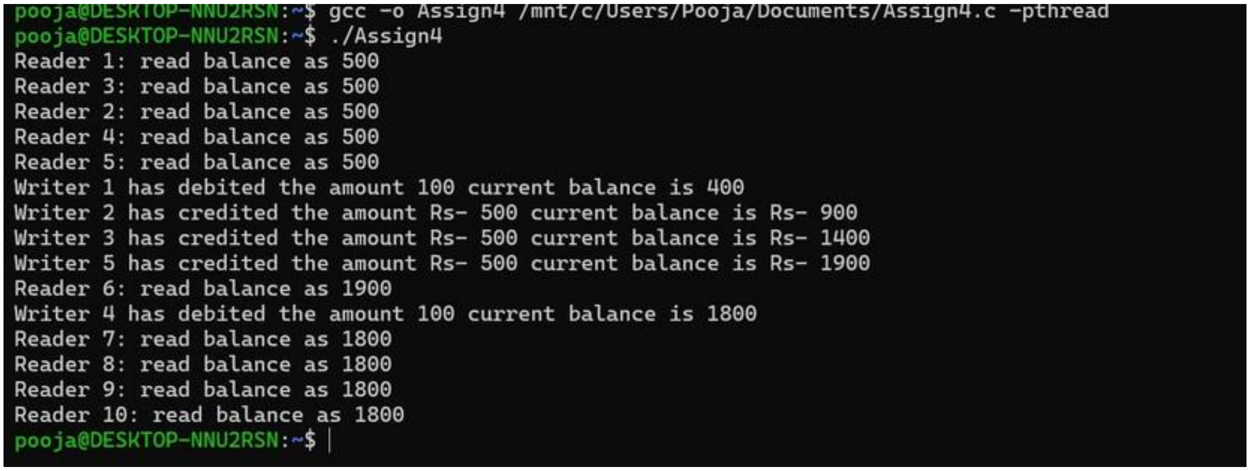
```
pooja@DESKTOP-NNU2RSN:~$ ./Assign4_Without
Reader 1: read balance as Rs- 500
Reader 3: read balance as Rs- 500
Reader 2: read balance as Rs- 500
Reader 4: read balance as Rs- 500
Reader 5: read balance as Rs- 500
Writer 1 has debited Rs- 100. Current balance: Rs- 400
Writer 2 has credited Rs- 500. Current balance: Rs- 900
Writer 3 has credited Rs- 500. Current balance: Rs- 1400
Writer 4 has debited Rs- 100. Current balance: Rs- 1300
Reader 7: read balance as Rs- 1300
Writer 5 has credited Rs- 500. Current balance: Rs- 1800
Reader 8: read balance as Rs- 1800
Reader 6: read balance as Rs- 1300
Reader 9: read balance as Rs- 1800
Reader 10: read balance as Rs- 1800
pooja@DESKTOP-NNU2RSN:~$ |
```

2) With Using Synchronization

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



sem_t wrt;
pthread_mutex_t mutex;
int balance = 500;
int numreader = 0;


void *credit_writer(void *wno)
{ int amount=500;

   sem_wait(&wrt);
   balance = balance+ amount;
   printf("Writer %d has credited the amount Rs- %d current balance is Rs- %d\n",(*((int
*)wno)),amount,balance);
   sem_post(&wrt);

}
void *debit_writer(void *wno)
{ int amount=100;

   sem_wait(&wrt);

   balance = balance-amount;
```

```
 printf("Writer %d has debited the amount %d current balance is %d\n",(*((int
*)wno)),amount,balance);
    sem_post(&wrt);

}
void *reader(void *rno)
{

    pthread_mutex_lock(&mutex);
    numreader++;
    if(numreader == 1) {
        sem_wait(&wrt);
    }
    pthread_mutex_unlock(&mutex);

    printf("Reader %d: read balance as %d\n",*((int *)rno),balance);


    pthread_mutex_lock(&mutex);
    numreader--;
    if(numreader == 0) {
        sem_post(&wrt);
    }
    pthread_mutex_unlock(&mutex);
}

int main()
{

    pthread_t reader_thread_id[10],writer_thread_id[5];
    char c[]= {'D','C','C','D', 'C'};
    pthread_mutex_init(&mutex, NULL);
    sem_init(&wrt,0,1);

    int a[10] = {1,2,3,4,5,6,7,8,9,10};

    for(int i = 0; i < 5; i++) {
        pthread_create(&reader_thread_id[i], NULL, (void *)reader, (void *)&a[i]);
    }
    for(int i = 0; i < 5; i++) {
            if(c[i]=='C')
             pthread_create(&writer_thread_id[i], NULL, (void *)credit_writer, (void *)&a[i]);
        else
             pthread_create(&writer_thread_id[i], NULL, (void *)debit_writer, (void *)&a[i]);
    }
            for(int i = 5; i < 10; i++) {
        pthread_create(&reader_thread_id[i], NULL, (void *)reader, (void *)&a[i]);
    }

    for(int i = 0; i < 10; i++) {
        pthread_join(reader_thread_id[i], NULL);
    }
    for(int i = 0; i < 5; i++) {
        pthread_join(writer_thread_id[i], NULL);
    }

    pthread_mutex_destroy(&mutex);
    sem_destroy(&wrt);

    return 0;

}
```

OUTPUT:



```
pooja@DESKTOP-NNU2RSN:~$ gcc -o Assign4 /mnt/c/Users/Pooja/Documents/Assign4.c -pthread
pooja@DESKTOP-NNU2RSN:~$ ./Assign4
Reader 1: read balance as 500
Reader 3: read balance as 500
Reader 2: read balance as 500
Reader 4: read balance as 500
Reader 5: read balance as 500
Writer 1 has debited the amount 100 current balance is 400
Writer 2 has credited the amount Rs- 500 current balance is Rs- 900
Writer 3 has credited the amount Rs- 500 current balance is Rs- 1400
Writer 5 has credited the amount Rs- 500 current balance is Rs- 1900
Reader 6: read balance as 1900
Writer 4 has debited the amount 100 current balance is 1800
Reader 7: read balance as 1800
Reader 8: read balance as 1800
Reader 9: read balance as 1800
Reader 10: read balance as 1800
pooja@DESKTOP-NNU2RSN:~$ |
```