

# Operating System

## Report Assignment Simulation Based

### 06. Programming Problem 5.41 Page 251 Operating System By Galvin 9<sup>th</sup> Edition

**5.41** A barrier is a tool for synchronizing the activity of a number of threads. When a thread reaches a barrier point, it cannot proceed until all other threads have reached this point as well. When the last thread reaches the barrier point, all threads are released and can resume concurrent execution. Assume that the barrier is initialized to N —the number of threads that must wait at the barrier point:

```
init(N);
```

Each thread then performs some work until it reaches the barrier point:

```
/* do some work for awhile */  
barrier point();  
/* do some work for awhile */
```

Using synchronization tools described in this chapter, construct a barrier that implements the following API :

- `int init(int n)` —Initializes the barrier to the specified size.
- `int barrier point(void)` —Identifies the barrier point. All threads are released from the barrier when the last thread reaches this point.

The return value of each function is used to identify error conditions. Each function will return 0 under normal operation and will return -1 if an error occurs. A testing harness is provided in the source code download to test your implementation of the barrier.

**Student Name:** Nikhil Kumar

**Student ID:** 11606029

**Section No:** K1605

**Roll No.:** 41B

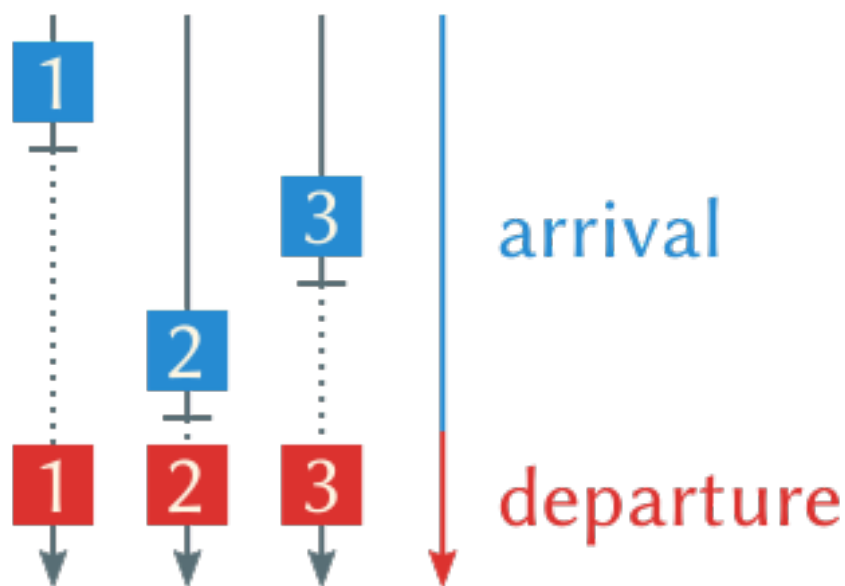
**Email Address:** [nikhil.dwivedi@outlook.com](mailto:nikhil.dwivedi@outlook.com)

**GitHub Link:** <https://github.com/NikhilKrDwivedi/OS-Assignment>

## Barriers

a barrier is a type of synchronization method. A barrier for a group of threads or processes in the source code means any thread/process must stop at this point and cannot proceed until all other threads/processes reach this barrier.

A barrier is a method to implement synchronization. Synchronization ensures that concurrently executing threads or processes do not execute specific portions of the program at the same time. When a barrier is inserted at a specific point in a program for a group of threads [processes], any thread [process] must stop at this point and cannot proceed until all other threads [processes] reach this barrier.



### Algorithm:

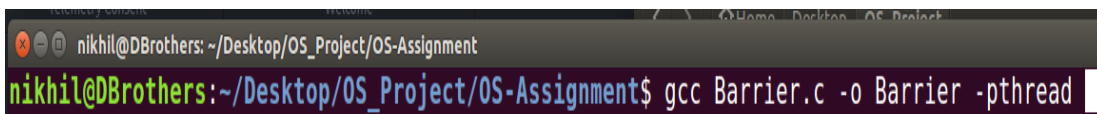
1. initialize barrier\_size and thread\_count;
2. create threads
3. threads doing some work
4. threads waiting at the barrier.
5. barrier is released when last thread comes at the thread.
6. all threads complete thier task and exit.
7. exit.

### Complexity:

$O(n)$  complexity. "n" is no of thread\_count.

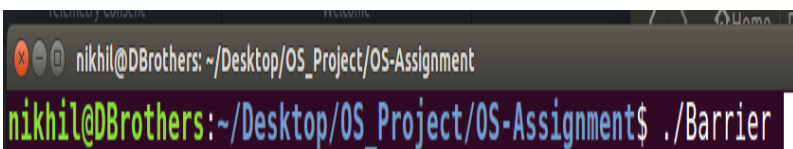
## Compile And Run:

Use following command to compile program -  
gcc Barrier.c -o barrier -pthread

A terminal window with a dark purple background. The prompt is 'nikhil@DBrothers: ~/Desktop/OS\_Project/OS-Assignment'. The command 'gcc Barrier.c -o Barrier -pthread' is entered and followed by a cursor.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
```

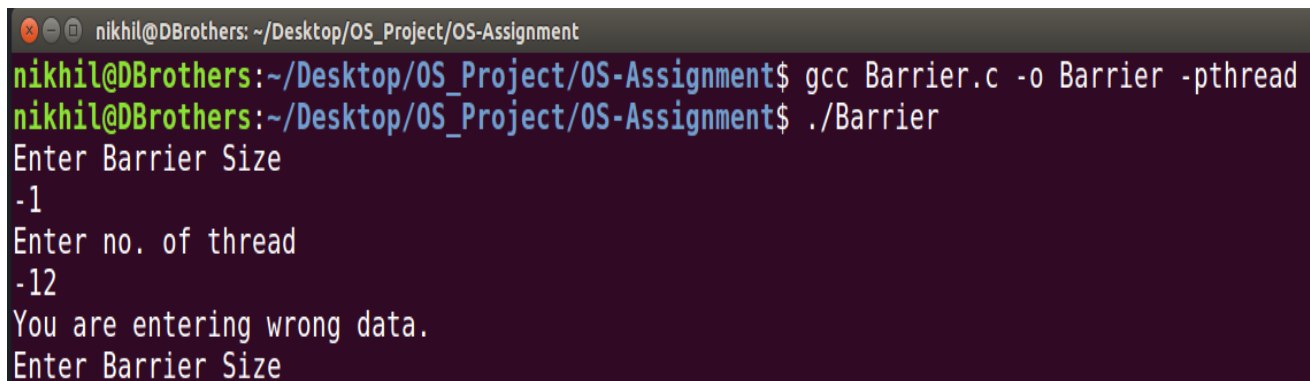
use following command to run program-  
./Barrier

A terminal window with a dark purple background. The prompt is 'nikhil@DBrothers: ~/Desktop/OS\_Project/OS-Assignment'. The command './Barrier' is entered and followed by a cursor.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment$ ./Barrier
```

## Test Cases :

**Case 1:** when user enter invalid input like – string, double, float, negative no. etc.

A terminal window with a dark purple background. The prompt is 'nikhil@DBrothers: ~/Desktop/OS\_Project/OS-Assignment'. The commands 'gcc Barrier.c -o Barrier -pthread' and './Barrier' are entered. The program prompts 'Enter Barrier Size' and '-1' is entered. It then prompts 'Enter no. of thread' and '-12' is entered. The program outputs 'You are entering wrong data.' and then 'Enter Barrier Size' followed by a cursor.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment$ ./Barrier
Enter Barrier Size
-1
Enter no. of thread
-12
You are entering wrong data.
Enter Barrier Size

```

**Case 2:** when no. of thread equal to size of barrier.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ ./Barrier
Enter Barrier Size
4
Enter no. of thread
4

Thread 1
Performing init task of length 3 sec

Thread 2
Performing init task of length 1 sec

Thread 3
Performing init task of length 2 sec

Thread 4
Performing init task of length 0 sec

Barrier is Released

I am task after barrier
I am task after barrier
I am task after barrier
I am task after barrier
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$
```

**Case 3:** when no. of thread is less than size of barrier .

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ ./Barrier
Enter Barrier Size
6
Enter no. of thread
3

Thread 1
Performing init task of length 3 sec

Thread 2
Performing init task of length 1 sec

Thread 3
Performing init task of length 2 sec

I am task after barrier
I am task after barrier
I am task after barrier
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$
```

**Case 4:** when no. of thread is greater than size of Barrier.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ ./Barrier
Enter Barrier Size
2
Enter no. of thread
4

Thread 1
Performing init task of length 3 sec

Thread 2
Performing init task of length 1 sec

Thread 4
Performing init task of length 0 sec

Thread 3
Performing init task of length 2 sec

Barrier is Released

I am task after barrier

I am task after barrier

Barrier is Released

I am task after barrier

I am task after barrier
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$
```

**Case 5:** when size of Barrier equal to '0'.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ ./Barrier
Enter Barrier Size
0
Enter no. of thread
2

Thread 1
Performing init task of length 3 sec

Thread 2
Performing init task of length 1 sec
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$
```

**Case 6:** when thread equal to '0'.

```
nikhil@DBrothers: ~/Desktop/OS_Project/OS-Assignment
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ gcc Barrier.c -o Barrier -pthread
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$ ./Barrier
Enter Barrier Size
4
Enter no. of thread
0
nikhil@DBrothers:~/Desktop/OS_Project/OS-Assignment$
```

**GitHub Link:** <https://github.com/NikhilKrDwivedi/OS-Assignment>