1: #include <pthread.h>

2: #include <semaphore.h>

3: #include <stdio.h>

4: #include <stdlib.h> // For rand() and srand() 5: #include <time.h> // For time():

7: sem\_t wrt; // Semaphore for writer access

8: pthread\_mutex\_t mutex; // Mutex for reader count 9: int cnt = 1; // Shared resource

10: int numreader = 0; // Number of active readers 12: void \*writer(void \*wno)

13: {

14: sem\_wait(&wrt);

15: cnt = cnt \* 2;

16: printf("Writer %d modified cnt to %d\n", \*((int \*)wno), cnt); 17: sem\_post(&wrt);

18: return NULL;

19: }

21: void \*reader(void \*rno) 22: {

23: // Reader acquires the lock before modifying numreader 24: pthread\_mutex\_lock(&mutex);

25: numreader++;

26: if (numreader == 1) {

27: sem\_wait(&wrt); // If this is the first reader, block the writer 28: }

29: pthread\_mutex\_unlock(&mutex); 31: // Reading Section

32: printf("Reader %d: read cnt as %d\n", \*((int \*)rno), cnt); 34: // Reader acquires the lock before modifying numreader 35: pthread\_mutex\_lock(&mutex);

36: numreader--;

37: if (numreader == 0) {

38: sem\_post(&wrt); // If this is the last reader, wake up the writer 39: }

40: pthread\_mutex\_unlock(&mutex);

42: return NULL;

43: }

45: int main()

46: {

47: pthread\_t read[10], write[5];

48: pthread\_mutex\_init(&mutex, NULL);

49: sem\_init(&wrt, 0, 1);

51: int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // Just used for numbering the readers and writers 53: srand((unsigned int)time(NULL)); // Seed for randomness

55: for (int i = 0; i < 10; i++) {

56: pthread\_create(&read[i], NULL, reader, (void \*)&a[i]); 57: }

59: for (int i = 0; i < 5; i++) {

60: pthread\_create(&write[i], NULL, writer, (void \*)&a[i]); 61: }

63: for (int i = 0; i < 10; i++) {

64: pthread\_join(read[i], NULL);

65: }

67: for (int i = 0; i < 5; i++) {

68: pthread\_join(write[i], NULL);

69: }

71: pthread\_mutex\_destroy(&mutex);

72: sem\_destroy(&wrt);

74: return 0;

75: }

