#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#define NUM\_READERS 3

#define NUM\_WRITERS 2

sem\_t mutex, wrt;

int data = 0;

int readers\_count = 0;

void \*reader(void \*arg) {

while (1) {

sem\_wait(&mutex);

readers\_count++;

if (readers\_count == 1) {

sem\_wait(&wrt);

}

sem\_post(&mutex);

printf("Reader %ld reads data: %d\n", (long)arg, data);

sem\_wait(&mutex);

readers\_count--;

if (readers\_count == 0) {

sem\_post(&wrt);

}

sem\_post(&mutex);

}

pthread\_exit(NULL);

}

void \*writer(void \*arg) {

while (1) {

sem\_wait(&wrt);

data++;

printf("Writer %ld writes data: %d\n", (long)arg, data);

sem\_post(&wrt);

}

pthread\_exit(NULL);

}

int main() {

pthread\_t readers[NUM\_READERS], writers[NUM\_WRITERS];

sem\_init(&mutex, 0, 1);

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

for (long i = 0; i < NUM\_READERS; i++) {

pthread\_create(&readers[i], NULL, reader, (void \*)i);

}

for (long i = 0; i < NUM\_WRITERS; i++) {

pthread\_create(&writers[i], NULL, writer, (void \*)i);

}

for (int i = 0; i < NUM\_READERS; i++) {

pthread\_join(readers[i], NULL);

}

for (int i = 0; i < NUM\_WRITERS; i++) {

pthread\_join(writers[i], NULL);

}

sem\_destroy(&mutex);

sem\_destroy(&wrt);

return 0;

}