//reader writer problem using threads

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#define NUM\_READERS 5

#define NUM\_WRITERS 5

pthread\_t readerThreads[NUM\_READERS], writerThreads[NUM\_WRITERS];

sem\_t mutex, resourceAccess, readCountAccess;

int readCount = 0;

void \*reader(void \*arg)

{

int readerId = \*((int \*)arg);

while (1)

{

sem\_wait(&readCountAccess);

readCount++;

if (readCount == 1)

{

sem\_wait(&resourceAccess);

}

sem\_post(&readCountAccess);

printf("Reader %d is reading.\n", readerId);

sem\_wait(&readCountAccess);

readCount--;

if (readCount == 0)

{

sem\_post(&resourceAccess);

}

sem\_post(&readCountAccess);

sleep(1);

}

return NULL;

}

void \*writer(void \*arg)

{

int writerId = \*((int \*)arg);

while (1)

{

sem\_wait(&resourceAccess);

printf("Writer %d is writing.\n", writerId);

sem\_post(&resourceAccess);

sleep(1);

}

return NULL;

}

int main() {

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

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

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

int readerIds[NUM\_READERS], writerIds[NUM\_WRITERS];

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

{

readerIds[i] = i + 1;

pthread\_create(&readerThreads[i], NULL, reader, &readerIds[i]);

}

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

{

writerIds[i] = i + 1;

pthread\_create(&writerThreads[i], NULL, writer, &writerIds[i]);

}

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

{

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

}

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

{

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

}

sem\_destroy(&mutex);

sem\_destroy(&resourceAccess);

sem\_destroy(&readCountAccess);

return 0;

}

Output:

