#include <pthread.h>

#include <semaphore.h>

#include <stdio.h>

sem\_t wrt;

pthread\_mutex\_t mutex;

int cnt=1;

int numreader=0;

void \*writer(void \*wno)

{

sem\_wait(&wrt);

cnt=cnt\*2;

printf("writer %d modified count to %d\n",(\*((int\*)wno)),cnt);

sem\_post(&wrt);

}

void \*reader(void \*rno)

{

//reader acquires lock before modifying numreader

pthread\_mutex\_lock(&mutex);

numreader++;

if(numreader==1)

{

sem\_wait(&wrt); //reader is present block the writer

}

pthread\_mutex\_unlock(&mutex);

printf("reader %d modified count to %d\n",(\*((int\*)rno)),cnt);//read section

pthread\_mutex\_lock(&mutex);

numreader--;

if(numreader==0)

{

sem\_post(&wrt); //reader is present block the writer

}

pthread\_mutex\_unlock(&mutex);

}

int main()

{

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

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 < 10; i++)

{

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

}

for(int i = 0; i < 5; i++)

{

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

}

for(int i = 0; i < 10; i++)

{

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

}

for(int i = 0; i < 5; i++)

{

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

}

pthread\_mutex\_destroy(&mutex);

sem\_destroy(&wrt);

return 0;

}

/\*output

comp84@comp84:~/Desktop/ti57$ gcc 4b.c

comp84@comp84:~/Desktop/ti57$ ./a.out

reader 1 modified count to 1

reader 2 modified count to 1

reader 3 modified count to 1

reader 8 modified count to 1

reader 4 modified count to 1

reader 6 modified count to 1

reader 9 modified count to 1

reader 7 modified count to 1

reader 5 modified count to 1

reader 10 modified count to 1

writer 1 modified count to 2

writer 3 modified count to 4

writer 2 modified count to 8

writer 5 modified count to 16

writer 4 modified count to 32

\*/