#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#define SLEEPTIME 5

volatile unsigned long counter = 0;

unsigned long numloops;

pthread\_mutex\_t mutexA = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_mutex\_t mutexB = PTHREAD\_MUTEX\_INITIALIZER;

void \*dowork1(void \*param)

{

for (int i = 0; i < numloops; i++) {

pthread\_mutex\_lock(&mutexA);

pthread\_mutex\_lock(&mutexB);

counter++;

if ((counter % 1000000) == 0) {

printf("WORKER1: %ld\n", counter);

}

pthread\_mutex\_unlock(&mutexB);

pthread\_mutex\_unlock(&mutexA);

}

return NULL;

}

void \*dowork2(void \*param)

{

for (int i = 0; i < numloops; i++) {

pthread\_mutex\_lock(&mutexA);

pthread\_mutex\_lock(&mutexB);

counter++;

if ((counter % 1000000) == 0) {

printf("WORKER2: %ld\n", counter);

}

pthread\_mutex\_unlock(&mutexB);

pthread\_mutex\_unlock(&mutexA);

}

return NULL;

}

int main(int argc, char \*\*argv) {

if (argc != 2) {

fprintf(stderr, "USAGE: %s <size>\n", argv[0]);

exit(-1);

}

numloops = atol( argv[1] );

pthread\_t w1, w2;

pthread\_create(&w1, NULL, dowork1, NULL);

pthread\_create(&w2, NULL, dowork2, NULL);

pthread\_join(w1, NULL);

pthread\_join(w2, NULL);

printf("FINISHED: counter == %ld\n", counter);

return 0;

}