#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#define NUM\_THREADS 2

#define NUM\_INCREMENTS 10000

int sharedCounter = 0;

pthread\_mutex\_t mutex;

void\* incrementCounter(void\* threadID) {

int id = \*((int\*)threadID);

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

pthread\_mutex\_lock(&mutex);

sharedCounter++;

pthread\_mutex\_unlock(&mutex);

}

printf("Thread %d finished\n", id);

pthread\_exit(NULL);

}

int main() {

pthread\_t threads[NUM\_THREADS];

int threadIDs[NUM\_THREADS];

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

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

threadIDs[i] = i;

int result = pthread\_create(&threads[i], NULL, incrementCounter, &threadIDs[i]);

if (result) {

fprintf(stderr, "Error creating thread %d: %d\n", i, result);

exit(-1);

}

}

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

pthread\_join(threads[i], NULL);}

pthread\_mutex\_destroy(&mutex);

printf("Shared Counter: %d\n", sharedCounter);

return 0;