**Write a program illustrating deadlock.**

#include <stdio.h>

#include <pthread.h>

pthread\_mutex\_t mutex1 = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_mutex\_t mutex2 = PTHREAD\_MUTEX\_INITIALIZER;

void \*thread1(void \*arg) {

pthread\_mutex\_lock(&mutex1);

printf("Thread 1 locked mutex 1\n");

sleep(1);

printf("Thread 1 attempting to lock mutex 2...\n");

pthread\_mutex\_lock(&mutex2);

printf("Thread 1 locked mutex 2\n");

pthread\_mutex\_unlock(&mutex2);

printf("Thread 1 unlocked mutex 2\n");

pthread\_mutex\_unlock(&mutex1);

printf("Thread 1 unlocked mutex 1\n");

pthread\_exit(NULL);

}

void \*thread2(void \*arg) {

pthread\_mutex\_lock(&mutex2);

printf("Thread 2 locked mutex 2\n");

sleep(1);

printf("Thread 2 attempting to lock mutex 1...\n");

pthread\_mutex\_lock(&mutex1);

printf("Thread 2 locked mutex 1\n");

pthread\_mutex\_unlock(&mutex1);

printf("Thread 2 unlocked mutex 1\n");

pthread\_mutex\_unlock(&mutex2);

printf("Thread 2 unlocked mutex 2\n");

pthread\_exit(NULL);

}

int main() {

pthread\_t t1, t2;

pthread\_create(&t1, NULL, thread1, NULL);

pthread\_create(&t2, NULL, thread2, NULL);

pthread\_join(t1, NULL);

pthread\_join(t2, NULL);

return 0;

}

