Practical-10

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;
}
```

```
Thread 2 locked mutex 2
Thread 1 locked mutex 1
Thread 2 attempting to lock mutex 1...
Thread 1 attempting to lock mutex 2...
```