answers/Code_Q2_6/2c/simple.c

```c
#include <pthread.h>
#include <stdio.h>
#include <time.h>
#include <aio.h>
#include <math.h>
#include <unistd.h>

pthread_mutex_t mutex;

typedef struct {
    double latitude;
    double longitude;
    double altitude;
    double roll;
    double pitch;
    double yaw;
    struct timespec sample_time;
} NavigationState;

NavigationState nav_state, nav_state_shouldbe;

pthread_mutex_t state_mutex;

void* update_thread(void * arg){
     for (int i = 0; i < 180; ++i) {
        pthread_mutex_lock(&mutex);
        nav_state.latitude = i;
        nav_state.longitude = 0.5 * i;
        nav_state.altitude = 0.25 * i;
        nav_state.roll = sin(i);
        nav_state.pitch = cos(i * i);
        nav_state.yaw = cos(i);
        clock_gettime(CLOCK_REALTIME, &nav_state.sample_time);
        pthread_mutex_unlock(&mutex);
        printf("updated reading\n");
        printf("Yaw: %f, Roll: %f, Pitch: %f, Latitude %f, Longitude %f, Altitude %f
\n",nav_state.yaw, nav_state.roll, nav_state.pitch, nav_state.latitude,
nav_state.longitude, nav_state.altitude);
        sleep(1);
    }
    return NULL;
}


void* read_thread(void* arg) {
    for (int i = 0; i < 18; ++i) {
        pthread_mutex_lock(&mutex);
        NavigationState temp_state = nav_state;
        pthread_mutex_unlock(&mutex);
        printf("Reading number %d\n", i);
        printf("Yaw: %f, Roll: %f, Pitch: %f, Latitude %f, Longitude %f, Altitude %f
\n",temp_state.yaw, temp_state.roll, temp_state.pitch, temp_state.latitude,
temp_state.longitude, temp_state.altitude);
```

```c
        printf("Time : tv_sec: %ld, tv_ns: %ld \n",temp_state.sample_time.tv_sec, temp_state.sample_time.tv_nsec);

        nav_state_shouldbe.latitude = i*10;
        nav_state_shouldbe.longitude = 0.5 * i * 10;
        nav_state_shouldbe.altitude = 0.25 * i * 10;
        nav_state_shouldbe.roll = sin(i*10);
        nav_state_shouldbe.pitch = cos(i * i * 100);
        nav_state_shouldbe.yaw = cos(i*10);

        printf("should be: Yaw: %f, Roll: %f, Pitch: %f, Latitude %f, Longitude %f, Altitude %f \n\n", nav_state_shouldbe.yaw, nav_state_shouldbe.roll, nav_state_shouldbe.pitch, nav_state_shouldbe.latitude, nav_state_shouldbe.longitude, nav_state_shouldbe.altitude);

        sleep(10);
    }
    return NULL;
}

int main(){
    pthread_t t1, t2;
    pthread_mutex_init(&mutex, NULL);
    pthread_create(&t1, NULL, update_thread, NULL);
    pthread_create(&t2, NULL, read_thread, NULL);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    return 0;
}

```