(a) The data involved in the race condition is `available_resources`.

(b) The race condition occurs in `decrease_count` function when two or more threads try to access the shared resource `available_resources` at the same.

(c) To address this issue, a mutex lock is used to ensure that only one thread can access the shared resource at a time. Modified code is as follows:

```
int decrease_count(int count) {
    static pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
    pthread_mutex_lock(&mutex);
    if (available_resources < count){
        pthread_mutex_unlock(&mutex);
        printf("Not enough resources\n");
        return -1;
    }
    else {
        available_resources -= count;
        pthread_mutex_unlock(&mutex);
        printf("Decrease count by %d, available resources: %d\n", count,
available_resources);
        return 0;
    }
}
```

additional print statements are added to show the decrease in the count of available resources.