

159.355: Concurrent Systems

Assignment 1: Semaphores

Hans Guesgen

1 Introduction

This is the first assignment for this course. It is worth 25% of the marks for the course. You have one problem to solve, for which you have to write code. You need to work on your own and you need to submit the assignment before the deadline stated on the Stream submission page.

Once you have a solution, submit a zip file containing your solution together with any comments about your code via Stream. Make sure you include your name and Massey ID number.

Remember that the course is about concurrency. It is your solution to the concurrency problem that I want to see, the rest of your code should be as simple as possible. Make sure that you have included sensible comments so that the marker can understand the code even if it is not running.

2 Your Task: The Cable Car to the Summit of Erebor

Long after the One Ring had been destroyed in the fires of Mount Doom, the dwarfs decided to return to their beloved mountain Erebor, also known as the Lonely Mountain. They decided to turn it into a tourist attraction and to build a cable car to the top of the mountain.

The cable car makes trips to the top of the mountain and back down again on a regular basis until it has transported 500 people up the mountain and back down again. Then it ceases operation for the day. On each trip it can take a maximum of 10 people to the summit, and on each return trip a maximum of 10 people back to the foot of the mountain. Since the dwarfs are very concerned about preserving Erebor, no more than 50 people are allowed on the summit at each point in time.

Tourists arrive randomly at the foot of the mountain during the day to use the cable car to get to the summit. They might have to wait for a space in the cable car because of its limited capacity. Also, the cable car only takes as many people to the summit as the limit of 50 people allows. Once a tourist has travelled to the top of the mountain, he or she stays there for a while and then waits for a cable car to return to the foot of Erebor.

Write a simulation of this problem using semaphores. Your solution should print out the details of what is happening, such as:

- A tourist arrives at the base station of the cable car.
- The cable car leaves with 7 passengers to the summit of the mountain.
- A tourist decides to leave the mountain and goes to the summit station.
- The cable car leaves with 4 passengers to the foot of the mountain.

Your solution should use a thread for tourists who want to get to the top of the mountain and another thread for tourists who want to get down again. Each of these threads terminates if they have dealt with 500 people.

Another thread should simulate the cable car. It waits for a certain amount of time to pick up tourists who want to get to the top of the mountain, travels to the top (simulated by sleeping for a while), releases the tourists that have been transported, waits again for a certain amount of time to pick up tourists for the return trip, travels back down, releases the returning tourists, and so on.

To simulate waiting for tourists to be transported, use the `tryAcquire` method:

```
public boolean tryAcquire(int permits,  
                           long timeout,  
                           TimeUnit unit)  
    throws InterruptedException
```

This method acquires the given number of permits from the semaphore, if all become available within the given waiting time and the current thread has not been interrupted. If insufficient permits are available then the current thread becomes dormant until one of three things happens:

- Some other thread invokes the release methods for this semaphore.
- Some other thread interrupts the current thread.
- The specified waiting time elapses.

Keep in mind that you have to count how many tourists are transported on each trip so that you can (1) release the appropriate number of tourists at the other end (simulated by another semaphore) and (2) make sure that the limit of 50 people at the summit is not exceeded. It is therefore a good idea to acquire the permits one at a time. If acquiring a permit was successful, `tryAcquire` will return `true`.