

159.355: Concurrent Systems

Assignment 2

Hans Guesgen

1 Introduction

This is the second assignment for this course. It is worth 25% of the marks for the course. You have two problems to solve, with equal weighting.

Once you have solutions, 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 solutions to the concurrency problems that we want to see, the rest of your code should be as simple as possible.

2 Your Tasks

2.1 The Library Problem

A small-town library has an inventory of 20 different books which the people of the town can borrow from. People make a request to the library for borrowing a book, and once the book becomes available, they will borrow it for a fixed amount of time. Only once they have returned the book to the library, can another person borrow the book.

Write a simulation of this problem, using the `synchronized` keyword as the synchronisation mechanism. Your solution should print out the details of what is happening, such as:

- Thread 5 wants book 3.
- Library gives book 4 to thread 2.

Your solution should generate 100 threads, each of which requests 3 books randomly from the inventory of the library. Once there are no requests left, your program should terminate.

2.2 The Budget Escalator

In order to save money, a department store decided to shut down all but one escalator in the store. The escalator connects the first with the second floor and changes the direction in which it travels every fixed time interval. Turnpikes at both ends of the escalator make sure that:

- There are at most 10 people on the escalator.
 - People can enter the escalator only if it operates in the direction they wish to travel.
 - Before changing direction, further access to the escalator is blocked, and only when all people have left the escalator, the direction is changed.
-

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

- Person 3, coming from the second floor, is leaving the escalator.
- Direction is about to change to allow travel from the first floor.
- Person 4, coming from the first floor, is entering the escalator.

Your solution should generate 100 threads, randomly distributed over the two floors. Once all people have travelled from one floor to the other, your program should terminate.