

159.355: Concurrent Systems

Assignment 1

Hans Guesgen

1 Introduction

This is the first 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 Sausage Sizzle

To raise funds for the new school hall, students are selling sausages in front of a heavily frequented warehouse. They have set up two barbecues, which produce sausages on a regular basis. These are then sold to the customers as they come in.

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:

- Barbecue 2 has another sausage ready.
- Customer 42 buys 2 sausages.

Your solution should generate 100 threads, each of which requests up to 3 sausages. Once there are no requests left, your program should terminate (even if there are sausages left over).

2.2 The Harbour Shuttle

To help easing traffic problems in Auckland, a company has introduced a shuttle service across the harbour. The shuttle seats 10 people and operates on a demand basis: once all 10 seats are taken, the shuttle crosses the harbour and waits there until 10 people on the other side want to cross the harbour.

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

- Person 3 is boarding the shuttle on the North Shore.
- Shuttle is going from the North Shore to Auckland.
- Person 4 is leaving the shuttle in Auckland.

Your solution should generate 100 threads, simulating people evenly distributed at both sides of the harbour (Auckland and the North Shore). Once all people have travelled from one side of the harbour to the other, your program should terminate.