

DS Assignment

Release date: 10 October 2015

Assignment deadline: November 13 2015

Feedback return: December 4 2015

This assignment is worth **25%** of the final mark. The assignment is a programming exercise that will be marked out of 25.

Read the complete description below before starting work.

Assignment Description

In this assignment, you have to create a simulation of a network and implement the ring based Chang and Roberts algorithm for leader election among the nodes of the network. You are also given an input file which contains the network; i.e., the nodes with their neighbours.

System design:

Your implementation should simulate a distributed system using multiple threads. Each node must run on its own thread. Additionally, a different thread will simulate the actions of the network -- delivering messages from one node to another. You can only use strings as messages.

Communication is assumed to be synchronous. Each synchronous round lasts for 20ms. Your program should simulate this behavior of the network receiving and delivering messages exactly once every round.

A node is allowed to send messages only to its neighbors as specified in the input file. It can send at most one message to each neighbor in one round. The network should enforce these constraints. You are also allowed to use a BROADCAST message that goes to all neighbors of a node.

Input file specification

The beginning of the input file contains the network graph. Each line describes a node; the first item is the id of the node, and the following ones are its neighbors.

The ordering of the rows gives an ordering of the nodes on the ring (the first follows the last).

The next part contains a list of leader elections initiated by different sets of nodes. These lines start with ELECT followed by the round number, followed by nodes that start election at that round. For example:

```
ELECT 5 18 7
```

Means that in round 5, nodes 18 and 7 start leader election.

Next there are lines describing nodes failing. For example, FAIL 20 means that node 20 fails.

A sample input file is available on the assignment page.

Assignments

The assignment is divided in two parts. Your tasks for each part are as follows:

Part A (marks 15/25)

1. Read the network specification from the input file.
2. Construct the ring topology. Use the order in which the nodes appear in the input file. (you can assume that this will be a valid ring on any input.)
3. Implement the Chang and Roberts algorithm for leader election among the nodes and initiate and execute elections according to the ELECT lines. (details of the algorithm can be found in the course slides as well as in the suggested references).

Observe that elections can start while one election is in progress, and your design must handle that.

Part B (marks 10/25)

This part handles node failures. When node X fails, we assume that the network knows this, and informs all neighbors of X.

On each failure, the nodes should take suitable actions, if any. For example, if the leader fails, a new leader must be elected.

You can assume that all distributed computations you need to complete on a failure ends before the next failure happens. (you need a way to check this.) The first failure happens after all elections from part A are completed.

Outputs

Log file: Your simulator should produce a log file “log.txt” that contains the main outputs. This will be the leader elected each time a leader election completes, in the following format:

```
LEADER 20  
LEADER 20  
LEADER 18
```

etc..

Print to screen: In addition, you should print to screen all important events as created and processed by your program. For example, initiation of elections, node failures, message delivery etc. Each must be on a separate line, and contain the round number, entities involved and contents.

Submission Instructions

You are expected to use **Java**. If you have a strong reason to use a different language, discuss with us first. However, we strongly suggest that you use Java.

The code will be tested on DICE. **If it does not run directly on DICE, we cannot evaluate it.**

The submission should contain the following in a single folder:

1. Your source code java files. **NOT just the executable**. If the source code is not included in the submission folder, your submission cannot be evaluated. The source code must be well commented and easy to understand. Note: all files should be in this one folder, and not in a folder substructure. Do not submit it as a GUI-specific project. Your code must compile and run as a flat set of java files.
2. A shell script called run.sh. This must compile and run your program when executed, and should take the input file as a parameter e.g. ./run.sh input.txt
3. A readme file (max 1 page in 12 pt font, 1 inch margin) that contains a summary of any specific implementation decisions you have made and a short summary description of the logic you followed for Part B. This is necessary in order to receive full marks.

To submit this, put everything in a single folder with name “ds_assignment_<your uun>” and use from DICE:

submit ds 1 ds_assignment_<your uun>

****IMPORTANT**** Outputs and submissions must conform to specifications given above. For example, **log.txt** must be generated and in the format above. Code must be in the top level folder. While you are free to use any development environment of your choice, do not submit an environment “project”, and make sure that the files work in flat source code format. If your submission does not conform to specifications, you cannot be given full marks.

Always use the most recent description on the web page. That is, use the up-to-date one on the web page and not a previously downloaded version.

University regulations:

On good Scholarly Practice. Please remember the University requirement as regards all assessed work. Details about this can be found at:

<http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct>

and at:

<http://www.inf.ed.ac.uk/admin/ITO/DivisionalGuidelinesPlagiarism.html>.

Remember, if you use ideas from elsewhere (including other students), cite them. And try not use too much of these. The regulation says you can pick up “general ideas” but not “pivotal ideas”. But “general” and “pivotal” are very subjective and depends very much on the person making the judgement. Play safe and avoid getting into trouble.