

# CS 511: Makeup Homework Assignment

## Due December 9, 11:55pm

### 1 Assignment Policies

**Collaboration Policy.** Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems or programming. Use of the Internet is allowed, but should not include searching for existing solutions.

**Under absolutely no circumstances code can be exchanged between students.** Excerpts of code presented in class can be used.

**Assignments from previous offerings of the course must not be re-used.** Violations will be penalized appropriately.

**Late Policy.** Late submissions are allowed with a penalty of 2 points per hour past the deadline.

### 2 Assignment Instructions

We wish to model a network of temperature sensors. Each sensor maintains a value representing its current temperature reading. This value may be modified in two ways: it can receive a direct reading through a `directReading` message (that updates the value of that sensor) or it can update its value based on the values of its neighboring sensors following the protocol which we describe next. **Each sensor has a list of its neighboring sensors (you may assume this list is fixed when the sensor is created).** All sensors are synchronized by means of a *Timer* process: the timer sends a `tick` message every certain number of milliseconds (given as a parameter), each sensor sends its value to all its neighbors each of which update their own value by taking the average of all received values.

Regarding the topology of the sensor network, it is read off from a file that consists of lines of comma separated strings indicating the name of a node and then the list of its neighbors. For example, here is the contents of a text file `star.txt` that encodes a “star” topology, that has `node1` at the center:

```
node1,node2,node3,node4
node2,node1
node3,node1
node4,node1
```

This should be read as an adjacency list in a graph. For example, the first line `node1,node2,node3,node4` means that `node1` has `node2,node3,node4` as neighbors. Here is the code that you can use to read and parse these text files:

```
1 readlines(FileName) ->
2   {ok, Device} = file:open(FileName, [read]),
3   try get_all_lines(Device)
4     after file:close(Device)
5   end.
6
7 get_all_lines(Device) ->
8   case io:get_line(Device, "") of
9     eof -> [];
10    Line -> Ss=string:tokens(Line, ",\n"),
11    [{hd(Ss),tl(Ss)}] ++ get_all_lines(Device)
12  end.
```

It produces the following output for the above mentioned text file:

```
1> makeup:readlines("star.txt").
[{"node1",["node2","node3","node4"]},
 {"node2",["node1"]},
 {"node3",["node1"]},
 {"node4",["node1"]}]
```

In your solution these strings will actually be understood as pids. In other words, each of `"node1"`, `"node2"`, `"node3"`, `"node4"` will have its own Pid and you will end up working with the Pids rather than the strings themselves.

### 3 Submission Instructions

Submit a `.erl` file named `makeup_<Surname>.erl` (where `<Surname>` should be replaced by your surname) through Canvas.