Step 2.10 ASSIGNMENT: Supervision in the frequency server

We've seen the frequency server, introduced models for its clients and seen how the system behaves in various modes using the observer tool. In this assignment we'll look at how to add a supervisor to a server/client system, and observe how it behaves.

Adding a supervisor

Program a supervisor process that can be used to start the server, and restart it whenever it has terminated unexpectedly.

In adding the supervisor you will need to think about the state of the server when it is restarted. The following discussion may help you to think about that.

In a typical client-server scenario, the clients are not under the control of the supervisor: any client can connect to the server as long as it knows the name (or the Pid) of the server. So, we can't expect the supervisor to restart the clients; on the other hand we can ensure that if the server terminates unexpectedly then the clients do too.

Using the observer

Using the observer tool as described in the previous exercise, observe how your system behaves when some of the constituent processes – including the supervisor itself – are killed.

Recall that to run the observer, type `observer:start()` in the Erlang shell, and that we use `exit(Pid,killed)` to kill the process with pid `Pid`.

Assignment Guidelines

The reviewers will be asked to give you feedback on the following aspects of your assignment, so you should consider these when writing:

- Ensure that the solution is correct; perhaps include some test code to show how the whole supervised system can be run.
- Ensure that the solution is readable; for example by including comments that explain how the functions work, and indicate any particular aspects that need explanation.
- Use comments to describe your experience of using the observer to examine how your system behaves when some of the processes are killed.