GAME OF THE ROPE[*]

One of the most popular traditional games is the *Game of the Rope*. Two teams of contestants face each other on a playground trying to decide which one is the strongest by pulling at opposite ends of a rope. If both teams are equally strong, the rope does not move, a standstill occurs and there is a draw; if however one of the teams is stronger than the other, the rope moves in the direction of the stronger team so much faster as the strength difference is larger and this team wins.

A variation of this game will be assumed here. A match is composed of three games and each game may take up to six trials. A game win is declared by asserting the position of a mark placed at the middle of the rope after six trials. The game may end sooner if the produced shift is greater or equal to four length units. We say in this case that the victory was won by knock out, otherwise, it will be a victory by points.

A team has five elements, but only three compete at each trial. Member selection for the trial is carried out by the team's coach. He decides who will join for next trial according to some predefined strategy. Each contestant will loose one unit of strength when he is pulling the rope and will gain one unit when he is seating at the bench. Somehow the coach perceives the physical state of each team member and may use this information to substantiate his decision.

In order to ensure rules compliance, there is a referee. She has full control of the procedure and decides when to start a new game, or a trial within the game. She also decides when a game is over and declares who has won a game or the match.

Write a simulation of the life cycle of the referee, the coaches and the contestants using the client-server model with server replication, where the referee, the coaches and the contestants are the *clients* and the access to the information sharing regions are the services provided to them by the *servers*.

The operations that were previously assigned to activities carried out in the information sharing regions (for the already implemented concurrent version), must now be assigned to independent requests performed on the servers through message passing. In each case, a connection has to be established, a request has to be made, waiting for the reply will follow and the connection has to be closed.

One aims for a solution to be written in Java, to be run in Linux under TCP sockets, either in a concentrated manner (on a single platform), or in a distributed fashion (up to 7 different platforms), and to terminate (it must contemplate service shutdown). A *logging* file, that describes the evolution of the internal state of the problem in a clear and precise way, must be included.

[*] Concept by Pedro Mariano

Guidelines for solution implementation

1. Specify for each representative server of an information sharing region the structure of the messages to be exchanged.
2. Specify the general organization of the servers architecture.
3. Specify the general organization of the clients architecture.
4. Sketch the interaction diagram which describes in a compact, but precise, way the dynamics of your solution. Go back to steps 1, 2 and 3 until you are satisfied the description is correct.
5. Proceed to its coding in Java as specific reference data types.
6. Specify the mapping of the servers and the clients onto multiples nodes of the parallel machine and write the shell scripts which enable the deployment and the execution of the different modules the application is composed of.
7. Validate your solution by taking several runs and checking for each, through the detailed inspection of the logging file, that the output data is indeed correct.