Phase 2- MIS threading

Andrew de Neve

23 November 2016

This report regards the implementation of Threading keywords into the MIS interpreter.

A new Threading keyword parent base class has been added to deal with passing new lists/data structures held in the MIS object (such as the vector of threads, thread id counter, locked variables map). All the new Threading keywords inherit from this class.

Process of creating a thread:

1. the parser hits the THREAD BEGIN keyword
2. Thread Begin object is created, making use of all resources in the MIS
3. using the parser, the Thread Begin object collects all instructions until hitting Thread End
4. instructions are placed in a list and a c++11 thread is created linking to a function called runThread.
5. in runThread the thread strips the first instruction and checks if its locked, if not it calls doInstruction method with the striped arguments
6. during doInstruction, a mutex is locked to prevent changes to the same variables

Under the hood, the computer running the MIS will jump between all the different threads the user has declared in the .mis program rather than executing sequentially. Thread jumping is to only happen after full execution of an instruction.

Additionally, the user must use BARRIER at the end of the .mis file to ensure that background threads are joined before main is terminated.