David Esperanza, Jonathan Ortiz, Pragun Sharma, Sean Woods

CMPS 109: Phase 4: SRI Engine Client/Server Virtual Machine

Implementation:

To implement our SRI Engine as Client and Server, abide to the following steps:

1. Implement a VituralBox network with preferably Ubuntu OS running on the machines.
    a. Implement as many sever and as many clients as needed to test the multithreaded component of the server.
2. With the folders provided in the submission, the folder named *server_files* shall be placed in the server and the folder named *client_files* shall be placed in the client.
3. To compile the sri engine & server component:
    a. g++ -std=c++11 -lpthread -pthread Connection.cpp GarbageCollector.cpp Inference_parse.cpp KB.cpp RB.cpp Node.cpp TCPServerSocket.cpp TCPSocket.cpp Thread.cpp Threads.cpp Transactions.cpp SocketMain -lboost_filesystem -lboost_system
4. To compile the command line Interface & client component:
    a. g++ -std=c++11 ClientMain.cpp
5. To run:
    a. On the Server: run ./a.out ;SRI server will start and wait for connections.
    b. On the Client: run ./aout x.x.x.x ;where(x.x.x.x is a IPv4 address)
6. Simulation:

    How to user our SRI Engine:

    **(See Assumptions)

List of Commands:	Example (Command Line)
LOAD	LOAD file_name.sri **
DUMP	DUMP dumpfile_name.sri **
FACT	FACT Father(Rick,Morty) **
RULE	RULE Parent($X, $Y) :- OR Father($X,$Y) Mother($X,$Y) **
INFERENCE	INFERENCE Parent($X, $Y) **
DROP	DROP Father; DROP Parent
CLEAR	clear **
QUIT	quit **

Assumptions:

1. User may load a file upon start :
    a. ./aout x.x.x.x file_name
    b. file must be placed on server in order to work
    c. same applies for while in command line session when using LOAD command

2. DUMP:
    a. When using DUMP command, file will be created on the server and reside on the server.
3. FACT:
    a. When entering the FACT command with a fact, user must be aware of the spaces in the command to ensure correct functionality.
        i. FACT_Father(Donald,Jeb) (where '_' represents a space)
4. RULE
    a. When entering the RULE command with a rule, user must be aware of the spaces in the command to ensure correct functionality.
        i. RULE_Parent($X,$Y)_:-_OR_Father($X,$Y)_Mother($X,$Y) (where '_' represents a space)
5. INFERENCE
    a. When entering the INFERENCE command with an inference to be performed, user must be aware of the spaces in the command to ensure correct functionality.
        i. INFERENCE_Parent($X,$Y) (where '_' represents a space)
6. DROP
    a. Drop can work on either a type of Fact like "Father" or can work on a type of rule like "GrandMother"
7. CLEAR
    a. Acts like "clear" on the UNIX Terminal
8. QUIT
    a. When user types quit, Server will receive quit command and closes connection with the client.

Files Included:

- Server:
    - Headers:
        - common.h
        - Connection.h
        - GarbageCollector.h
        - Includes.h
        - Inference_parse.h
        - KB.h
        - Node.h
        - RB.h
        - TCPServerSocket.h
        - TCPSocket.h
        - Thread.h
        - Threads.h
        - Transactions.h
    - Sources:
        - Connection.cpp

        - GarbageCollector.cpp
        - Inference_parse.cpp
        - KB.cpp
        - Node.cpp
        - RB.cpp
        - TCPServerSocket.cpp
        - TCPScoekt.cpp
        - Thread.cpp //Karim's
        - Threads.cpp//Ours
        - Transactions.cpp
        - SocketMain.cpp
- Client:
    - Headers:
        - Common.h
        - Connection.h
        - GarabageCollector.h
        - Includes.h
        - TCPServerSocket.h
        - TCPSocket.h
        - Thread.h
    - Sources:
        - ClientMain.cpp
        - GarabageCollector.cpp
        - TCPServerSocket.cpp
        - TCPSocket.cpp
        - Thread.cpp