Kendra Maggiore

Project 2

CS4310

Design Document

For this project, I used C because the agent file was written in C, I split the program into two different files, and two header files to keep everything abstracted and easier to find.

agentStruct.h is the header file for my AgentInfo struct

helper.h is the header file for all my helper functions

helper.c is the helper functions implemented

main.c is the server file that contains the main server configuration

agentStruct.h:

This is the header file that holds the struct AgentInfo. This struct contains a char array of 20 for the ipaddress’s of the agent. It also contains time\_t startTime to keep track of the time for the agent. I choose to give this struct its own header files so that the helper functions could access the same struct as the main.c file.

helper.h:

This file holds the functions prototypes for the functions that are implemented in helper.c. There are 6 function porotypes that are in this file. This file was a must have for the helper.c file.

helper.c:

This file holds all the implementations of the helper functions. There are six function implementations in this file. They are addAgent(), removeAgent(), listOfAgents(), checkAgent(), fTime(), and cAction(). I chose to separate the files because I wanted them to be abstracted, and if I ever wanted to add another server, I could still use these helper functions. addAgent() adds the ip addresses of the agent to the cAgents array, it also starts the time for the agent. removeAgent() removed the ip addresses of the agent from the cAgent array. listOfAgents() creates a agentList of current agents in the cAgent array, and it also adds a time stamp in seconds. checkAgent() checks if the agent is still in the cAgent array. fTime() uses the time.h, and sys/time.h to get the current time in hours ,minutes, seconds, and milliseconds to add to the log.txt file. cAction() takes the cBuffer and compares it to a array string and then returns a integer value for the action.

main.c:

This file is the server file that contains the main server configuration and setup. This file uses socket programing to create a socket, bind the socket to the port specified by the user, and then listen for a connection. Once, a connection is established the program accepts the connection, and then reads the socket, buffer, and ipAddress of the agent. Then the program gets the Action of the agent and then runs though a switch statement to see what action was sent by the server. When I originally created this switch statement. I had functions for each action such as joinAction that would then go and execute the join action, but this was only working correctly on my windows computer, and would not work on the linux server. So, I just put everything for the actions into the switch statement even though I would have liked to have abstracted all of them in separate functions.

Each switch case executes an action based on its number that was provided by the cAction() function. The #Join tells the agent that the server has received a #JOIN request and then checks if the agent is a member of the agent list or not if not the sever then adds the agent. if the agent is the member the server tells the agent that it is already a member.

The #LEAVE tells the agent that the server has received a #LEAVE request and then checks if the agent is a member of the agent list. If the agent is a member the sever removes the agent from the agent list. if the agent is not a member the server tells the agent that it is not a member.

The #LIST tells the agent that the server has received a #LIST request and then checks if the agent is a member of the agent list. If the agent is a member the sever sends the list to the agent. if the agent is not a member the server does not send the list and tells the agent that it is not a member.

The #LOG tells the agent that the server has received a #LOG request and then checks if the agent is a member of the agent list. If the agent is a member the sever send the log file to the agent. if the agent is not a member the server will not send the file and will tell the server that it is not a member.

All of these statements also writes to a log.txt file to keep track of everything going on with the agent connections.