SNOWCAST README

Brown University CS168 Network Project 1

Author: Yisheng Zhou
Login: yzhou53
Email: Yisheng_Zhou@brown.edu

snowcast_listener.c

In the listener file, there is only one main function. The program loads the UDP port from the command line and creates a socket. In the main while loop, the listener receives data from the server using **recvfrom**, saves it in the buffer, and directly uses write function to express the buffer. The wait time between two receive actions is 1/16 second as 1000000000/16 nanoseconds since the frequency is defined to be 16.

snowcast_control.c

Structures

There are three structures defined in the control file. One is *client_msg* recording the information the control will send to the server, which are one 8 bits unsigned integer and one 16 bits unsigned integer. *Welcome_msg* records the welcome message from the server, and *song_msg* records the songname message or the error message from the server. Since these two types of data are both two integers plus one strings, the program just has one structure for them.

Functions

The following parts include functions which can get integers and strings from buffers or save integers and strings into buffers. The next function is **open_client**, who tries to connect the server with the given server name and server port. **Send_hello** and **send_set_station** allow the control to send different kinds of message to the server. **Receive_welcome** and **receive_announce** can receive messages from the server and do actions based on the received messages. Since what the client received are buffers, **get_welcome** and **get_announce** make the bit be understandable for the receive functions.

Main

The last part is the main function. It has two while loops. The first one is used to receive a legal station number and the second one makes the client keep receiving information from the server. I failed to apply the selection function in this loop so the client cannot detect the input char to stop. It only stops when the server closes its connection of this control.

snowcast_server.c

Message Structures

There are four kinds of message structures in the server. Command records the messages from the clients. It includes type and number. If the message is *hello*, the number represents the UDP port. If the message is *set_station*, the number represents the station index. There are three kinds of messages that the server will send including *welcome_msg*, *song_msg* and *error_msg*. In *song_msg* or *error_msg*, there are an integer saving the length of the string and a pointer pointing the string it will send.

Thread Structures

There are two major structures saving and sending data. One is the client structure, which works as a node in a linked list saving in the station. It also has a pointer pointing to its station for easy access. Another one is the station structure. It records three pointers pointing to the song name string, the song file and the first client node. There is a global mutex array which can make sure one station is not modified in two different places.

Functions for Threads

The following parts include functions which can get integers and strings from buffers, or save integers and strings into buffers. The server program divides the send information parts into three individual functions including **send_welcome, send_songname** and **send_invalid**, but only has one function used to receive data from the clients, since the clients could only send messages with two integers.

Between clients and stations, there are two functions to build the connections between these two structures. **Station_add_client** works to add a client into a station by modifying the linked list in the station structure and the pointers in the client structure. **Station_del_client** does the reverse action. For client and station structures, they both have an initial function to start a new thread and an end function to kill a dead thread.

Functions for Server

There are several functions for the server part. **Accept_server** works to build a new connection between a control client and the main server program through TCP communication. **Quit_server** and **print_server** are used in the main loop interacting with the user of the server. **Open_receiver** is the function to initialize the TCP port of the server so it can listen the port.

Main

The main_loop has a while loop, which has a select function inside. It keeps listening the TCP port while detecting the input_char given by the user. The main function loads the port and song file names from the command line, creates station threads and the mutexes for those threads.