Relay Design Plan

Jeremiah Katen, Jack Sanchez, Michael Miano

22 April 2021

1 Introduction

This is the design plan for the relay project. The purpose of this project is to create two programs, a dispatcher, and a listener. The dispatcher broadcasts messages to all current listeners. If a listener is opened and there is no dispatcher, the program will exit. If the dispatcher's execution is stopped then all listeners executions will also stop.

2 Implementation

In order for the program to run correctly the RELAY environment variable must be set by using this code: "export RELAY=23669". The dispatcher will then set up sockets and handle new listener connections for the relay program so that the listeners can then connect to the dispatcher and receive the messages that are being broadcast. Once the program dispatcher is running, the user can type into their terminal and when the return key is pressed their message will be broadcast to all available listeners.

2.1 Program Flags

The dispatcher can be run with two different flags:

- -b: removes the buffer from the program so that as the user types into the dispatcher the listeners will be receiving what is typed.
- -l <num>: allows the user to specify the max number of listeners that can connect. Must be a number between 1-100.

Once the number of listeners has reached the max number of connections, further attempted connections will be refused. While the dispatcher is running, all connected listener's will continue to receive messages. Once the dispatcher is exited using either ctrl+c or ctrl+d, all connected listeners will also stop execution.

2.2 Sockets

The dispatcher will handle the connection of new listeners using a thread to continue checking for new connections while the program is running. New listeners can connect at any time while the dispatcher is running. If the dispatcher is at the maximum number of connections allowed then any new connections will be refused.

2.3 -b Flag

When the -b flag is given the program will no longer wait for the return key to be pressed in order for the listeners to receive a message. The format for this flag is "./dispatcher -b". As the message is being typed the listeners will be receiving the messages using termios. With the use of termios the dispatcher WILL NOT be able to use the backspace key to make corrections. The return key may still be used to get to a new line.

2.4 -l Flag

The -l flag allows the dispatcher to determine how many listeners can connect to the program at once. The format for this flag is "./dispatcher -l <num>". The number given must be in the range of 1-100. Once the maximum number of connections has been reached then the dispatcher will no longer accept new connections. If a listener disconnects then another listener will be able to connect to the dispatcher.

3 Dispatcher Important Functions

The following functions are going to contain the major functionality of the dispatcher program.

3.1 get_clients

This function will be running on a thread and will be used to check for new listener connections and set them up appropriately. If the dispatcher is at the max number of connections then this function will block new listeners from connecting.

3.2 buffer_input

This function will be used to get input and then send the data to the listeners once the dispatchers has pressed enter. If the dispatcher is closed using ctrl+c or ctrl+d then the program will exit and all connected listeners will be disconnected and their execution will also terminate.

3.3 no_buffer_input

This function will be used to get input as it is being typed and continually send the data to the listeners. This function allows the -b flag to work and will not wait for the dispatcher to have the return key pressed to send data. Data will constantly be send to the listeners. In this mode the dispatcher will be unable to use the backspace key to erase what is typed. Once data is typed it will be sent to the listeners.

4 Listener Important Functions

The following functions are going to contain the major functionality of the listener program.

4.1 setup_socket

This function will be used to setup the socket for the listener and then attempt to connect to the dispatcher if it is available and it is not at the max number of connections allowed.

4.2 get_input

This function will be used to continue to read data that is being sent by the dispatcher. Whenever new data is sent from the dispatcher the listener will read the data and then print it to the terminal window so that the user can read the data. If the -b flag was used with the dispatcher then this input will be received a character at a time instead of a line at a time.

5 Summary

This project consists of two programs, the listener, and the dispatcher. Once the dispatcher is running then listeners are able to connect to the dispatcher and wait for data to be sent out from the dispatcher. If the max number of connections have been reached then no further connections will be allowed until a listener disconnects from the dispatcher. If the dispatcher's execution is terminated through the use of ctrl+c or ctrl+d then all connected listener's execution will also be terminated.

6 Contact Information

For more information on this program you can contact Jeremiah Katen(jkaten@gmail.com), Jack Sanchez(jsanchez@gmail.com), or Michael Miano(mmiano@gmail.com).