

# CS454 Assignment 3 Manual

Shoubo Wang (sbwang, 20476417) and Xiaochen Liu (xcliu, 20469767)

## Data Marshalling/Unmarshalling

We use a character buffer as our data buffer for transmitting information through send and receive calls, so all data are converted to a character array using `memcpy()` prior to send or receive. The message starts of with a 4-byte integer indicating the length of the whole request, then another 4-byte integer representing the message type (e.g., `LOC_REQUEST`), following with the actual message content. The message content format varies with the type of the message.

## Binder Implementation

We use a map as the database for storing each server's registered functions. The map key is a `FuncSignature` object that stores the function name and an array of argument types. The value is a vector of `ServerLoc` objects containing the server identifier, port as well as the socket file descriptor.

This database design allows us to easily detect function overloading because overloaded function will result in a different `FuncSignature` object thus producing different map keys. Also with this design, array arguments of different lengths will produce the same `FuncSignature` object thus occupying the same key in the database map. We have also implemented the comparator methods to both `FuncSignature` and `ServerLoc` classes to facilitate searching functions and servers inside the database.

We use a vector of `ServerLoc` to store the registered servers as our round robin queue.

Upon receiving a register request from the servers, the server information is added to the database and the round robin queue if it is not yet on queue.

Upon receiving a location request from the client, the binder retrieves the list of available servers supporting the specified function by doing a map key lookup. Then it iterates through the list and the round robin queue to search for the next available server.

The binder first checks the server at the front of the queue to see if it supports the specified function; if so, the server information (id and port) is sent to the client and the server is pushed to the back of the queue using a rotate function. However, if the first server does not support the specified function, it is pushed to the back of the queue and move on to the next server. If there is no matching server, then the binder will send a location request failure message back to the client.

If the binder detects a disconnecting server, the binder will remove the server from the queue and the database map.

Upon receiving a terminate request, the binder sends terminate request to all registered servers and then checks the connection with all servers. Once verified that all servers are disconnected, it will then terminate itself.

## Error Codes

INIT\_BINDER\_ADDRESS\_NOT\_FOUND = -1,  
- binder address is not set in environment variables

INIT\_BINDER\_PORT\_NOT\_FOUND = 2,  
- binder port is not set in environment variables

INIT\_BINDER\_SOCKET\_OPEN\_FAILURE = -3,  
- cannot open a socket connection to binder

INIT\_BINDER\_SOCKET\_BIND\_FAILURE = -4,  
- cannot bind the socket connection to binder

INIT\_LOCAL\_SOCKET\_OPEN\_FAILURE = -5,  
- cannot open local socket connection

INIT\_LOCAL\_SOCKET\_BIND\_FAILURE = -6,  
- cannot bind local socket connection

REGISTER\_BINDER\_DID\_NOT\_INITIATE = -7,  
- binder is not yet initiated when attempting to register

REGISTER\_BINDER\_RET\_UNRECON\_TYPE = -8,  
- cannot recognize the returned message from binder when registering

EXECUTE\_ADDRINFO\_ERROR = -9,  
- address information is invalid when attempting to execute

EXECUTE\_SOCKET\_OPEN\_FAILURE = -10,  
- cannot open a socket connection when attempting to execute

EXECUTE\_SOCKET\_BIND\_FAILURE = -11,  
- cannot bind a socket connection when attempting to execute

EXECUTE\_SELECTION\_FAILURE = -12,  
- cannot select a socket when attempting to execute

CALL\_BINDER\_ADDRESS\_NOT\_FOUND = -31,  
- binder address is not set in environment variables

CALL\_BINDER\_PORT\_NOT\_FOUND = -32,  
- binder port is not set in environment variables

CALL\_BINDER\_FUNCTION\_NOT\_FOUND = -33,  
- the specified function is not found when performing a lookup

RECEIVE\_ERROR = -404  
- when select() fails

SOCKET\_OPEN\_FAILURE = -17,  
- cannot open a socket connection

SOCKET\_BIND\_FAILURE = -18,  
- cannot bind a socket connection

UNKNOWN\_MSG\_TYPE\_RESPONSE = -19  
- received message with invalid message type

TERMINATE\_BINDER\_DID\_NOT\_INITIATE = 1,  
- binder is not initiated when attempting to terminate

REGISTER\_BIDNER\_FUNCTION\_OVERRIDEN = 2,  
- registered function is overridden

## Unimplemented Features

All features for the assignment were implemented, except for the bonus functionalities.