Design Document: Asg1

Jinghao Shen
CruzID: jshen30

1 Goals

The goal for this assignment is to implement a simple single-threaded RPC server that will support the math functions and file functions. The server will respond to a standard RPC protocol, which is given, providing the results of several file system functions(read, write, list, unlink, status) that the server will implement. The program will run the server in a directory, and requests for files will be served from under that directory.

2 Design

This program first set up sockets for basic client and server communication.
Get host name and port from arguments.

```
uint16_t function
read(socket, &recvBuf, 0)
Store the first 2 byte in function (same way as get an uint16_t from buffer)
```

Depending on the function called retrieve the needed bytes.
Then, convert values to and from the given big-endian input, wire format, to a data that can be stored in five different types of buffer, uint8_t, uint16_t, uint32_t, uint64_t, and uint8_t *, using shift.
Do several unit tests on this part.

For uint:

```
uint8_t recvBuf[]
uint(8/16/32/64)_t variable
for (from startIndex to bytesNeed) {
    variable = variable << 8 | recvBuf[index]
}
```

For words:

```
Get the first 2 bytes (same way as get an uint16_t from buffer)
Count = length of the string
for(from startIndex to count){
    uint8_t *str = str << 8 | recvBuf[index]
```

Based on the function call, the program uses “switch” to call different functions.
To do the math functions:

```
function(buffer[] without function call and identifier, operator){
    a = first 8 bytes from buffer
    b = second half of the buffer
    return a+b / a-b / a*b
```

To do the file functions: