CS 536 lab 1 report

Shahab Rahimirad

Problem 1:

The reason that the code does not work is rooted in the use of execlp function. The function call execlp attempts to execute the command specified in the first variable with the arguments provided in the rest of them. When buf is set to "ls -l", the execlp function does not execute correctly because it expects the command and its arguments to be separate strings. To execute “ls -l”, the function must be called like: execlp("ls", "ls", "-l", NULL). But since buf=“ls -l”, it is being called like execlp("ls -l", "ls -l", NULL)

Problem 2:

Determining length limit:
To check to see if the command is less than 30 characters, we first need to extract commands from the message received by the server. First we split the message based on ‘\n’, then check each command from it to see if it contains more than 30 characters or not.

Bytes for atomic write:

The number of bytes up to which write() behaves atomically is defined by the PIPE_BUF constant in GNU based systems, which represents the maximum number of bytes that is guaranteed to be written atomically to a pipe. For FIFOs, this behavior is the same as for pipes. So the number of characters written must not be greater than PIPE_BUF.
The value of PIPE_BUF can be obtained using the pathconf() or fpathconf() functions with the _PC_PIPE_BUF argument. This value is usually at least 512. In linux systems, the value of PIPE_BUF is 4096 bytes.

Problem 3:

The main difference between execve and execvp lies in the way the file path is specified. The execvp function takes the file name and looks for it in ‘PATH’ environment variables to execute it. But for execve, it takes the file path instead of its name. So we need to specify the entire path to execute it. We need to add the PATH variable to the start of the first argument. In linux systems it is equal to ‘/bin/’ because it is where programs such as ‘ls’ are stored. We add this using snprintf function.