ELEC 377 Lab 1 Test Cases

Gia Lee, Valerie So, William Robson

Testing Completeness

This testing is thorough and complete. Testing consists of running internal and external commands and the terminal returns expected answers. Testing includes running commands coded within the lab such as “cd”, “ls”, “pwd” and “exit”. See Script file “test1.txt” in gitlab.

Blank:

Simply hitting enter or putting all spaces into the shell will just go to the next line, no arguments are seen so no error message is thrown, very clean.

SplitCommandLine:

It is critical to ensure that the function was able to parse out the arguments correctly. Commands with multiple arguments were tested and were all recognized by the debugging printout and were registered correctly. All edge cases were fleshed out, adding many spaces at the beginning and end of the argument as well as spaces between the arguments. Exceeding maximum arguments will also throw the correct error message. The program will always read groups of characters as intended and store them properly in the args[] array.

Internal commands:

“cd” was able to change directories when a valid directory was input after “cd” and an error message was outputted when an invalid directory was inputted. When “cd” was input without modifiers, it changed to the home directory. Too many arguments prints an error.

“ls” was able to output the existing files in the current directory. The ‘-a’ show all function works as intended. Given the wrong argument or too many arguments it prints an error.

“pwd” outputted the working directories, too many arguments print an error.

“exit” terminated the shell and exited, too many arguments print an error.

External commands:

If a command is not found in internal commands, it is tried against the execv function.

“No internal or external program found!” will be printed if a command is not recognized internally or externally. Most invalid answers wind up with this output. The commands we tested:

The script command worked as intended in the shell with different file names.

The custom “hello” c executable worked as intended.

The group made a custom “rot” c executable that returns text, that works as intended

The echo function will echo arguments back as intended.

Any *valid* command with *invalid* arguments will fail and print "Failed to execute command: %s\n" with command name.

Test cases are thorough and meet all the requirements discussed in the lab description. No matter how many keys you press the shell will not crash or have a major breakdown (unless commandbuffer size is exceeded). It is always there on the next line waiting for a valid command.