Lab 1 Testing Documentation

3.2.2

After functions *skipChar* and *splitCommandLine* have been coded, they were tested using multiple phrases and varying cases. The special case of multiple spaces was considered and tested. During testing, one logic error was discovered and corrected. This error was caused by a missing increment of the pointer to the back of the word and was noticed through testing because the output was incorrect after the first word. After this logic error was corrected, the functions produced the expected output given the various inputs entered. An example of such inputs and outputs is an input of “one two three” and a corresponding output of “0: one\n1:two\n2:three\n”. This testing indicates the proper functionality of the *skipChar* and *splitCommandLine* functions because, together and independently, they produce their desired and expected outputs of an array of words from a user entered string.

3.2.3-3.2.4

After creating the typedef of a function pointer and creating a structure of this typedef, the code was checked for errors. The code compiled and was able to run successfully without any compiler errors. After writing the dispatch table code, the code was checked for visible errors and run. The code successfully compiled without any errors; there was no output related to the recent changes to check at this step.

3.2.5

After the *doCommand* function was completed, the function was compiled with no errors. After the implementation of the dispatch table, the function was tested using valid commands and invalid commands with additional debugging print statements, not included in the submission file. The function again ran without error and produced the correct output when an invalid entry was requested.

3.2.6

The additional commands were completed individually before being tested all together. *pwd* was implemented first. The code would not run at first until it was realized that the pwd library is in linux, not C. When tested on the VM, the pwd compiled and was run without errors.

cd was implemented next. With *cd*, we had an error in our pointer logic for finding the directory. As a result, we could not find the home directory and change correctly. To find the error, we reviewed the password structure and how the directory was obtained, which is where our error was located. The value from pw_dir→ pw was being saved to *directory* rather than saving the value from pw→pw_dir. After swapping the two, the directory changed correctly. After being fixed, an error message was attached in the case where there is an error getting the home directory or if no such directory exists. Tests were conducted to validate the three different cases, and each were successful.

exit was implemented third and had no problems compiling and producing the correct response.

ls was implemented last as it was the hardest to comprehend. The *ls* implements the filter, so both were tested together. The first test was conducted to test the prefix --a and then with no letter. The string was compared and successfully disabled the filter if the letter was present and printed out the error if the option was invalid. The next test was conducted to see if the entries would print correctly. This was conducted using the *scandir* function and print statements. The original test failed as it was printing directories out of order but was promptly fixed after realizing the for loop was incrementing incorrectly. After fixing the second test, the entire function compiled successfully and produced the correct output.

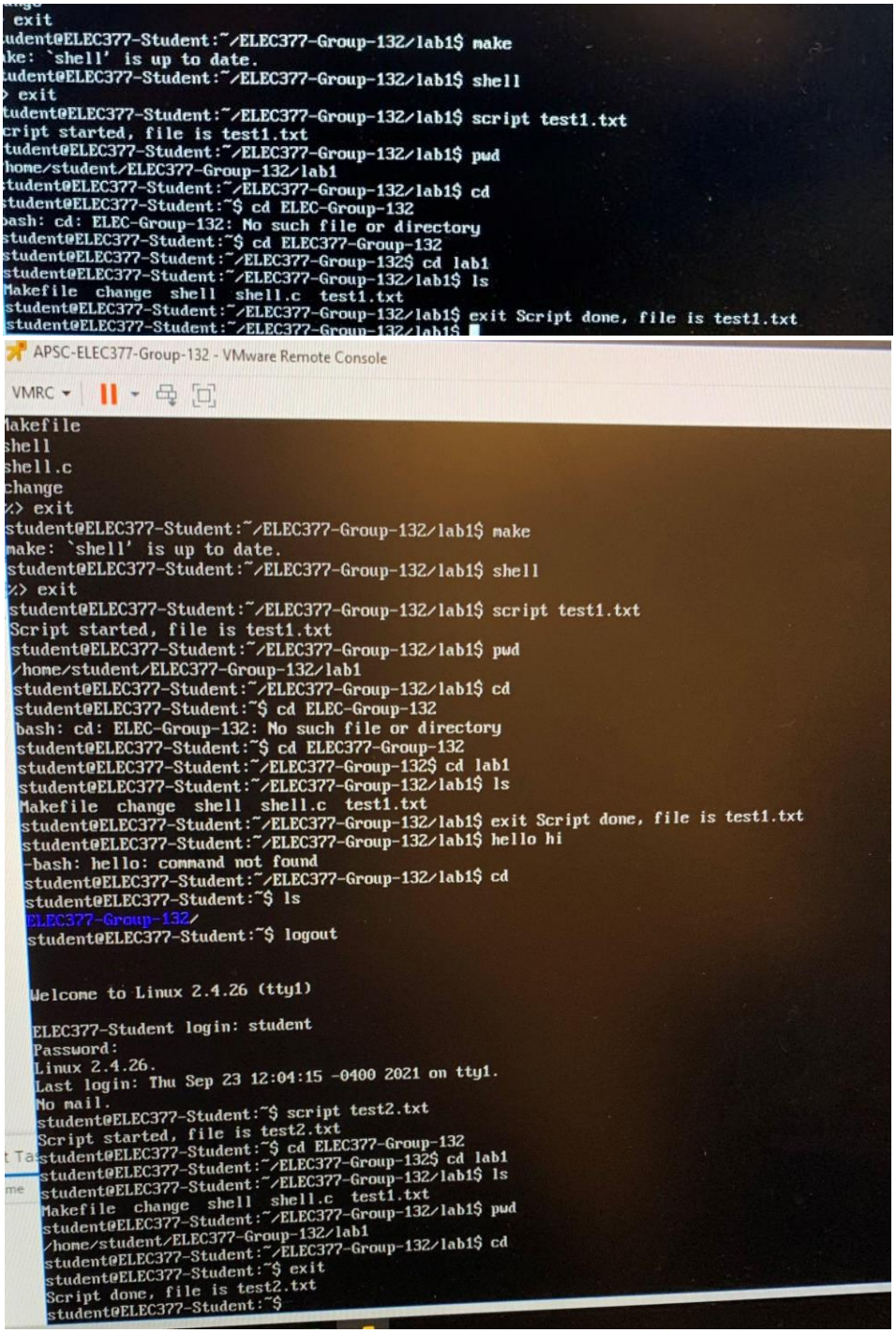
4. Testing

The first test we performed was the *ls* command. *ls* is deemed functional if the terminal lists the contents of the current directory, one per line. When the command was entered in the terminal under the lab1 directory, the terminal correctly printed out the available files and folders (seen in file finaltestvmshell.txt, lines 5-11). As the terminal correctly listed the contents of the current directory, one per line, the test confirms that the command *ls* works.

The second test performed was the *cd* command. *cd* is deemed functional if the terminal swaps the current directory to the one given by the parameter. When the command was entered in the terminal under the lab1 directory, the directory was correctly changed from “student@ELEC377-Student:~/ELEC377-Group-132/lab1$” to “student@ELEC377-Student:~/ELEC377-Group-132$” (seen in test1.txt, between lines 4-8). As the directories was changed successfully, the test confirms that the command *cd* works.

The third test performed was the *pwd* command. *pwd* is deemed functional if the terminal correctly prints out the current pathway. When the command was entered in the terminal under the lab1 directory, the terminal correctly printed out the current folder open (seen finaltestvmshell.txt, lines 2-3). As the terminal correctly printed the current pathway, this test confirms that the *pwd* command works.

The final test performed was the *exit* command. *exit* is deemed successful if the terminal successfully exits the shell with an exit code of 0. When the command was entered, the shell was successfully exited with the script end message being printed to show proof of exit (seen in test1.txt, lines 11-12). As the terminal successfully exited the shell, the test confirms that the *exit* command works.



Figures 1 and 2: As some of the script files do not show the special characters, the above photos have been provided to show proof of test results. The validity of the photos can be seen at the top of the second photo with our group ID. 132.