Project No. 1: Quite a Shell (quash)

Anjali Pare and Victoria Maldonado

In the quash shell code, we created a loop that gets user input and continues executing the specified command until “quit” or “exit” is typed. For each of the command line typed, we first split the same and then called specific functions depending on the keywords in the line read in.

1. Run executables without arguments.

On reading the user input, “launch” function is called with the command parameters. First, an existing process forks itself into two separate ones. Then, the child uses execvp() to replace itself and run a new program. If not a background process, parent waits for the child to finish execution using the system call wait(). It is important to send NULL at the end of the arguments list because execvp requires it to be a null terminated list. The first parameter of execvp is the executable name and the next is the argument list for that executable (In this case it is just NULL, because we are running executables with no arguments). We tested this using a dummy program that required no arguments and also using simple commands such as ls, pwd and quit.

1. Run executables with arguments

This has the same execution as the one above, only difference being in the argument list sent to execvp. In this case, it will contain the arguments required for the executable and complete execution.

1. set for HOME and PATH work properly

For this feature, when the user input contains the keyword “set”, the setPaths function is called. This function is responsible for first splitting the environment variable name from its value (these are passed to the function as a single string). Then setenv function is used to appropriately change the HOME and PATH environment variables. Depending on whether the setenv function was successful in changing the path or not, we print appropriate messages on the terminal. Once quash is done implementing this command, printenv VARIABLE\_NAME can be used to see if the environment variable was updated or not (VARIABLE\_NAME can be either HOME or PATH).

1. exit and quit work properly

Every command entered is first checked if it is equal to “quit” or “exit”. If yes, the quash shell exits successfully.

1. cd (with and without arguments) works properly

For this feature when the user input contains the keyword “cd”, the changeDir function is called. Depending on whether cd is executed with or without arguments, changeDir is given the HOME environment variable of the user entered directory name as input respectively. This function then call chdir with the passed in parameter and updates the PWD environment variable to correctly display the current directory the user is currently in. Since we use PWD to print the working directory of the user each time a new command is entered, this was easy to test.

1. PATH works properly. Give error messages when the executable is not found

For this feature to work, we made use of the execvp function call again in the launch function. If execvp returned a value less than 0, we printed the error message saying the executable was not found. We tested the execution by using a dummy executable that did not exist and the error message was successfully printed.