Shell assignment 1

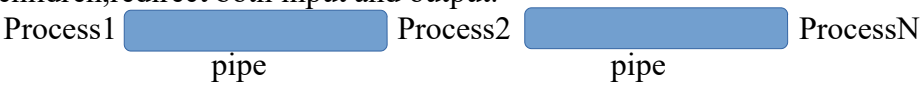
Danciulescu Dragos-Gabriel
s1086730
Andrei Octavian Dan
s1046913

Design of solution:

For the internal commands, we just check for “exit” and “cd” as commands and use chdir() and exit() functions accordingly. We also implemented “cd ..” command.

Next, we create a list of 2 file descriptors and iterate through the number of pipes(nrOfCommands – 1) and create pairs with pipe() system call. We then add the file descriptors of the pipes to pipesRead and PipesWrite vectors.

Next, we create a list for the children processes. If there is more than 1 command,we iterate through all the commands and, in the parent process add children processes to the list. If we’re in the first child process we get input from the file if need be and redirect only output from the first command in case of chained commands. If we are in the last child, we only redirect input. For all other children,redirect both input and output.



In case of only one command, we also handle input and output files and fork a child to execute the one command(otherwise our shell would exit after each command).

In the end, close all pipes in a for loop, otherwise the shell won’t know when to stop reading commands and we wait for all children to terminate, only if the “&” argument is not in the command.

Testing:

When it comes to tests, we did not add any new cases. The given tests do not check the infinite buffer problem (what would happen if 2 processes wrote or tried to read from the same pipe). We do not have this problem in our implementation because every process only reads from or writes to one pipe.