

Joseph Shumway

830003458

3/2/2022

PA-2 Report

This PA I would say is by far the most difficult one I have ever worked on. The thought process and design required for implementation totally went against how I naturally think. However, with lots of help from the TAs in lab, I was able to complete the assignment.

Initially I was planning on using main only for all the coding work. I added in various parts such as the prompt, cd functionality, etc. without too much of a hitch. By the middle of the assignment though it became clear that a modular approach would have benefited me greatly since a lot of the confusion in dealing with pipes would have been minimized. By this time, it was too late to refactor the code and move everything into functions (believe me, I tried), so I continued forward. Instead, I sectioned off each requirement with comments and visual breaks. This helped a lot but could have still benefitted from using functions.

As far as the sections themselves, I started with the prompt. I used both time() and ctime() to grab the current time and date, then using getenv() to grab the user, and finally using getcwd() to grab the working directory of the shell. All of this was parsed and concatenated to create the lovely yellow prompt before each command in the shell. Pretty simple.

Next, I focused on single commands and cd. Using fork() and execvp() was much easier than last PA and ran without hitches. At this point the shell was functioning, but I wanted to make sure cd worked well before anything more advanced. I initially wanted to save the last directory and use it for "cd -", but instead for the sake of simplicity and frankly to just get another passing test, chose to just use substring to set the cwd to the current path minus the current folder. This was a mistake in the long run, but it somehow passed the tests, so I at least got some serotonin from it.

After this point, I was feeling ready to challenge redirection. This is where things get ugly. I didn't understand until later how simple dup2 is conceptually and overthought it way too much. I tried a couple of approaches but until lab I was in a coder's block. The TA's helped me to understand dup2 and file descriptors and I was able to finish redirection. I applied this knowledge to piping and was able to set up a working pipe. However, for reasons I still don't understand it wasn't passing on the output and was instead outputting to the console. I rewrote the same code about 3 times so I honestly can't remember much from this section. I once again took my code to lab and was able to wrestle with it with my TA Mary for about an hour until it finally passed the tests. The issue seemed to be how and when I closed the file descriptors. The logic was mostly there, it was just a couple of lines that were troubling the rest of the code. In the end, I would like to say I have learned a lot about pipe() and file descriptors, but I am beginning to think that my forte is not deep in the Linux OS.