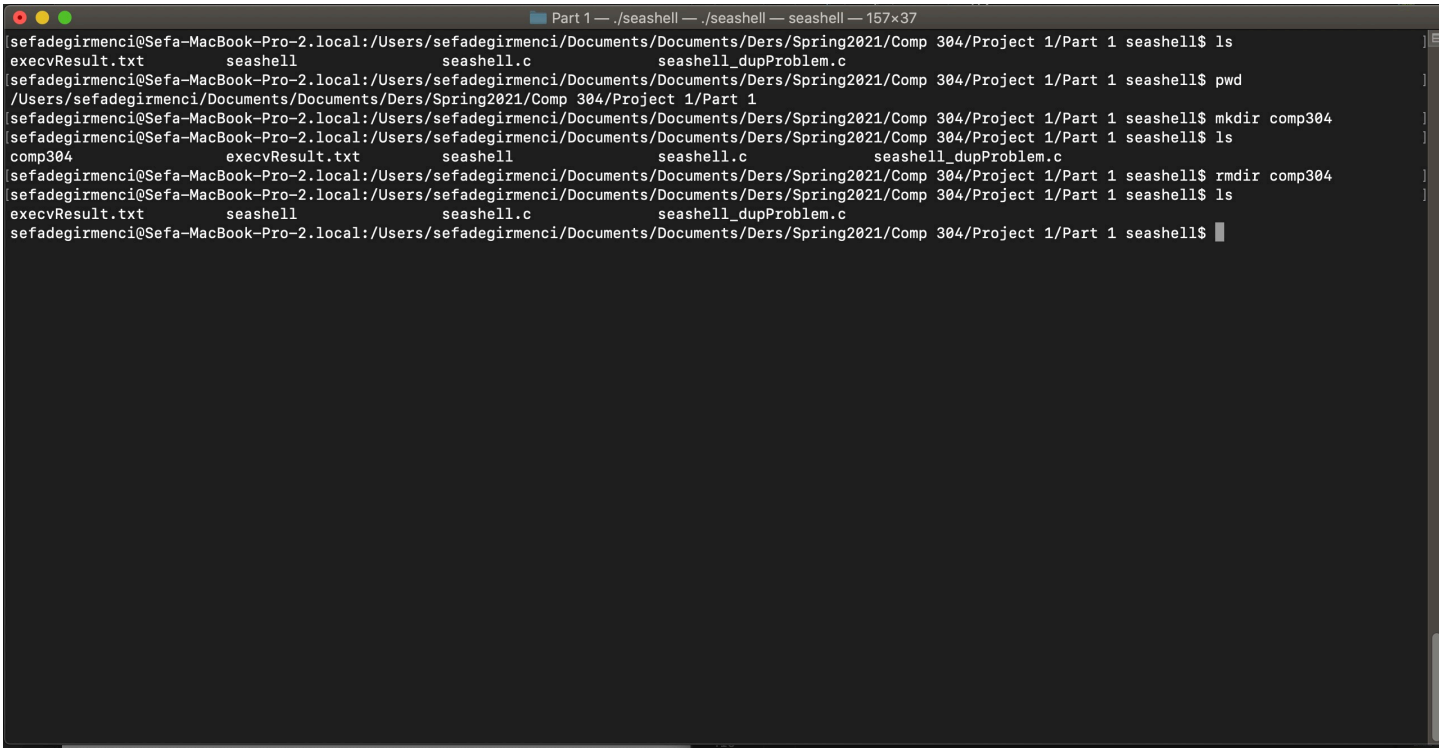
Part 1 : Execv System Call



General Approach: Execv requires path information along with the program name in contrast to execvp. Our approach was creating a child process, finding the path name by executing whereis program in the child process and then using this path information in the parent process to use execv system call.

The Struggle to Send Path to Parent Process: Our first method was to create a pipe between processes. We used dup2 to duplicate our file descriptor and make its file descriptor “1” -> stdout. After dup2 when we call exec, we expected its output to go through pipe’s write end and we could read from the read end in the parent process. Even though in separate practice codes this approach worked, in our project we got random characters like @æΩ from the read end of the pipe. So we changed our approach.

Sending Path Using .txt File: Dup2 can redirect every file descriptor. After creating a child process, we create an txt file and use dup2 to connect it to std_out. When child process executes whereis command, its output directly goes to txt file. In the parent process we read from the txt file using fopen and fgets. Then we split the output of whereis command using strtok. If we reach a token starting with ‘/‘ that means it is the path information.* **We use execv with the path information.**

*In linux whereis gives 3 separate information, in Mac it gives only the path information.Strtok is arranged accordingly.