Alex Olteanu Lab 2

Question 1: Here goes my answer to question one. My default terminal prompt is "(base) alexolteanu@Alexs-MacBook-Pro-2 ~ %".

Question 2: Here goes my answer to question two. The full path to my home directory is "/Users/alexolteanu"

Question 3: Here goes my answer to question three. The files and folders inside my home directory are "Applications", "Music", "Desktop", "Downloads", "Library", "Movies", "Music", "OneDrive - Tulane University", "Pictures", "Public", "venv".

Question 4: Here goes my answer to question four. The full path to my cmps2300 directory is "/Users/alexolteanu/CMPS2300".

Question 5: Here goes my answer to question five. The command to do this is "mv lab1.txt cmps2300/".

Question 6: Here goes my answer to question six. At first, the variable value in both the parent and child process is the same. After the fork(), any changes to this variable is now independent because the processes do not share the same memory address space. Changing one var will not affect the other.

Question 7: Here goes my answer to question seven. Both the child and parent processes are able to access the file descriptor. They are also both able to write to the file descriptor although the order is randomly selected. This is unless a wait() function is implemented which ensures that the child process will write first.

Question 8: Here goes my answer to question eight. We need multiple varieties of the exec() function because each has a specialized functionality that requires differing parameters to accomplish the objective at hand. This multitude of options gives us greater control over interacting with our machine.

Question 9: Here goes my answer to question nine. The wait() function returns the PID of an available process. In the child it will return -1.

Question 10: Here goes my answer to question ten. The waitpid() function becomes useful when we wish to reference a specific process.

Question 11: Here goes my answer to question eleven. If the descriptor is closed, the child will not be able to write to the file and printf() will fail to execute.

Question 12: Here goes my answer to question twelve. The fork() and exec() functions will create new processes while wait() and waitpid() would cause the parents to wait for the child to complete. Every child must perform each subsequent function following fork.