Vincent Maggioli

Lab 1

3.1

c. The ancestor process runs in an infinite loop for the rest of its existence.

d. Nulluser() does not ever return from start.S.

e. Halt() is in intr.S and it stops the CPU.

f. When I removed the halt() call after nulluser(), nothing changed.

g. When I replaced the while loop at the end of nulluser() with a call to halt(), nothing changed.

3.2

a. After a new process is created in Linux using fork(), the parent and child processes both run.

b. Since a new process has been made obviously with the intent to run, I believe the child process will run before the parent.

c. As an app programmer my preference would be that the child process runs first. This is because the parent process may rely on a result from the child process, so that executing first would be optimal.

e. The difference between newProcess() and create() is that create() makes a process from nothing, while newProcess() duplicates the parent process.

f. The create() method is different than clone() in that the process made by create() has its own execution context while the process made by clone() can share it with the parent process.

g. The posix_spawn() function differs from newProcess() in that the new process is created in an image passed in as an argument while newProcess() uses the parent process image to create a child process.

h. There is no best way to create processes because every context is different when creating a process. Different situations may be optimized by a specific kind of process creation, but considering the amount of processes and how quick they all are in general there is no best way to create a process.