

Problem 3

3.1

- a. nulluser() is defined starting from line 43 in initialize.c.
- b. create() is the ancestor of all processes and is defined in create.c. It assigns itself with a PID of 1.
- c. nulluser() does nothing for the rest of its existence.
- d. nulluser() does not ever return.
- e. halt() is located in intr.S. halt() is a do nothing infinite loop.
- f. Removing nulluser() doesn't initialize the system.
- g. Replacing the while-loop at the end of nulluser() with halt() doesn't make a difference.

3.2

- a. The fork function is used to create a new process by duplicating the existing process from which it is called. The new child process runs concurrently with the parent and has the same copy of address space and is given a pid of 0.
- b. It can be either parent or child depending on the scheduler.
- c. I would prefer to have child to run first since that is why I called fork. Then the parent being the "owner" of the child process, can clean up or do something after the child has run.
- e. newProcess calls the executable binary in a child process. XINU's create just sets up the address space for the new process. It only handles the memory allocations for stack and return addresses and adds them to the process table.
- f. Linux's clone shares the same address space when it spawns a child process. This is used to implement threads. XINU's create only setup the memory for a new process.
- g. posix_spawn lets you have more control over what memory is shared between the parent and child processes.
- h. No, there is no general best way to create processes. Each of the different ways to create processes have varying degrees of control.