Logan Montgomery

CSC 362 Operating Systems

Prof. McCartin-Lim

15 April 2021

To Fork or Not to Fork

Fork! Yes its outdated, its terrible for many reasons, but, when designing my own OS I would include the Fork system call. Hopefully by the end of this short paper you will agree with me. My argument is simple. Compatibility. The issue with stripping fork out of an operating system all together is that too many process already use the fork system call. These programs would be rendered useless until someone wrote a patch or completely re formatted the code to use thread instead. Similarly, windows users would like to run Unix based programs. Yes some are compatible with both operating systems, but this is only when the producers of the program have written different versions for each OS. To eliminate this issue, an ideal OS would not only have Fork, but also thread. This would allow for the best compatibility. For these reason I also believe that Fork should still exist in the new Linux OS. Now, there are still issues to address, like, “Fork doesn’t compose” , “Fork isn’t thread-safe.” , “Fork is slow.”( Baumann). All of the reason that “A fork() in the road” dislikes fork are valid. They are mostly right in their disposition of fork and I tend to agree with them on many of their points. I would not agree with how they plan to move forward. It would be best to keep fork in the OS and allow it to be used for a number of years while encouraging the use of spawn or thread as they mention. This would allow for greater compatibility and eliminate any “growing pains” that are associated with such a drastic change in a well know and loved system.