CMSI 387-01

OPERATING SYSTEMS

Spring 2013

Assignment 0321 Feedback

Andrew Kowalczyk

1a — You were able to work the command line and terminal windows effectively to get some of the requested shots. You figured out how to get Linux's `ps` to display a threads count—that's a good find, but admittedly `ps` can actually give you much more than that. I'm a tad surprised that you were not able to determine similar `ps` options for Mac OS X, sticking to the GUI monitor instead. I won't consider these to be a detraction to your command line abilities, but I would say these will hit *1d* instead. (+)

1b — In terms of working with processes and threads, your screenshots were pretty much in the right direction, although we had a misunderstanding about what was meant by "in the background." What I meant was for a process to *keep running* in the background, and not paused the way you did it. Your interpretation is certainly valid also, although as you can see, your way doesn't quite produce a "big hot terminal mess." **:)** Beyond that, I didn't see much commentary with respect to comparing Linux and Mac OS X approaches; admittedly, your screenshots don't provide much to chew on, but you certainly could have said that much, just to show that you addressed the issue of comparison. (|)

1d — No points to ponder seen, so will leave unevaluated. (**O**)

2c — Your shell is still in a very preliminary state, so this proficiency necessarily can't be very high. It's better than having nothing yet though, that's for sure. Be sure to read the inline comments that I committed with your source code. (/)

3a — You were able to accomplish your process business on a couple of platforms, but as indicated in *1b*, not to the full extent requested by the "CSI: Process" part of the assignment. (|)

4a — As mentioned, your shell is still in a pretty early state, but even at this point it can use some tips, so definitely look at my inline comments. (/)

4b — The code is not far enough in for a separation-of-concerns review, but as you move on, make sure to keep in mind that you want your variable names to be clear and descriptive, and be open to defining helper functions that keep self-contained computations separated, in order to make the main line of code more compact and easier to follow. (/)

4c — In its current state, your code is decently easy to read, although admittedly most of it is still from my sample **:)** Still, stay consistent with the formatting and you should be fine. Remember that C is syntactically a lot like Java and JavaScript, so whatever is stylistically correct in those languages is likely appropriate in this language also. (/)

4d — I think it is worthwhile for you to do just a touch more digging in terms of learning how to monitor or list threads on the operating systems that you used. And meanwhile, your shell work will definitely need a good chunk of documentation and web lookup. (/)

4e — You show good commit frequency and commit messages; I will assume that you will keep this up with future work. (+)

4f — Submitted on time except for the points-to-ponder answers. The shell remains a work in progress, but something was still there by the due date so it really is just the points-to-ponder that is missing. (|)

Updated feedback for commits up to May 1:

1d — Your points to ponder responses were nearly spot-on, except in your view of whether your shell can use `|`, `>`, or `<`. As a matter of fact it does—*all* command-line Unix programs can! I committed a script to demonstrate that point. (|)

2c, *4a*, *4b*, *4c*, *4d* — We have shell!!! And a very clean, easy to read, functionally complete one at that. (+)