Lab Two

Eric Stenton

Eric.Stenton1@Marist.edu

September 15, 2019

1 Problem One

Question: How is your console like the ancient TTY subsystem in Unix as described in `https://www.linusakesson.net/programming/tty/`?

Answer: My console shares some commonalities with the ancient TTY subsystem in UNIX described in the link such as line editing in the line discipline, session management, and sending OP code signals to the CPU. While the canvastext.ts file handles the graphical component of the console, the console.ts file deals with the functions of the line discipline similar to the TTY subsystem. Here, the backspace, erasing on the graphical console, and reprinting of text is managed. The file receives input from the keyboard driver as it would from the physical device and handles it as if it is in 'cooked' mode. In the next few labs, I will be implementing session management in the form of commands to control processes with system calls and support for OP codes to carry out provided user programs. While it is more akin to simple mimicry, the console in our project performs some processes the ancient TTY subsystem did in the past that were essential in the human and machine relationship.

2 Problem Two

Question: LaTeX?

Answer: The similarities between my console and LaTeX are fewer than the TTY subsystem. LaTeX works similar to a user program that needs to be compiled or a job that is executed rather than commands that are interpreted line by line on a console. It does, however, introduce a type of I/O process that is somewhat similar. I compile a TeX file with all the structure notation and strings of text that I wish to display pre-written, then I am given an output of a PDF file. It is akin to my status command that takes my input and displays it within a graphic task bar on the web page. It is important to note, however, that LaTeX is very unlike a shell or bash script which function more similarly to how the console interprets commands in the project due to LaTeX's nature of needing to be compiled.