Coby Cockrell
11/20/2022
OS Part2

GitHub: https://github.com/CobyEC/OSSim.git

The Memory addition to the operating system is being added to the specific processes instead of the linked list like in the first definition, there are many more working pieces to add before this is complete. This is no justification, but in the past 2 weeks my processor pipeline implementation with introducing new data hazards for my instruction set has taken up much time, and then trying to implement a joint signal processing 5G network utilizing MRC and Channel estimation for capstone has been eating the other half of my brain. Processor design being over with thanksgiving will be the grindstone for my operating system.

I demo the linked list after the initial process demo, I also updated the implementation to move the process out of the initial state after it is added to the linked list. Thus, implementing my linked list as my container for my overall processes.

PREV Instruction:

In order to use what is left of my Operating System, in the main, there are displayed two example prints of the XML files. My UI had originally been an interrupt menu that would stall until the next user input (The options included –help, –log, –RunCycles, –loadFile). In order to run a process, after initializing an object of the StateMachine class, all that is needed would be to call the RunSim function to run a single cycle of the operating system. I have not yet implemented a scheduler, as I spent most of the time rebuilding the Process's connection to the State Machine.

Note, My implementation uses the "rapidxml.hpp" library, this was a standard library included at this source: https://rapidxml.sourceforge.net/rapidxml.hpp

The Two XML files I have included have the following input format:

```
<TemplateData>
        <Operation>... Being (CALCULATE, I/O, or FORK)
                <name>...  Being(Calc, IO, Fork)
                <min>...   Being(any positive integer)
                <max>...  Being(any positive integer)
        </Operation>
        …
</TemplateData>
```