**ECE 421: Assignment 2 - Systems Programming in Ruby- Final Reflection**

**By: Reem Maarouf, Calvin Ho, and Aaron Philips**

1. Augmented decisions with regard to the design questions:

- Syntax error checking were not used in error conditioning in part 1 as was mentioned in the proposal. Our ShellWrapper class delegates the commands to the system class in ruby and therefore uses its error conditions for invalid inputs.

- Rubys $SAFE value was not adjusted to level 4. Permission to adjust the value to a safe level between 2-4 was denied. Our Shell has a safe level of 1 instead.

- We did not use optionParser as discussed in the proposal. We found that the Regexp and the shell class were valid enough to parse through the commands. They were simpler to implement that the optionParser.

- For part 2 of the project, the interrupt exception was not implemented. If control C is entered in the system then the process will not exit but will continue to operate until the timeout has occurred.

- For more accurate timing, the time resolution is changed to the nearest nanosecond instead of millisecond.

- For part 3, a modified file is determined when the modified time changes from when the program is initially executed. The file class was used in this part as it contained a method that returns the timestamp of when a file was last modified.

1. There are no deviations from the contracts that were specified in part 2.
2. See attached files for a copy of the codes
3. There are no additional contracts that were not specified in the part 2.
4. There are a couple of limitations in part 3 of the assignment. If, for instance, the FileWatcherCreation is called and is watching for a specific file creation anywhere in the system, and a file with that name was deleted before the creation of that file, the watcher will not be able to notify the user that this file has been created. This problem also occurs in our FileWatcherDestroy watcher. If a file with the same name has been created first and then deleted then it would not be able to capture that deletion and notify the user. Another limitation in our system occurs when a files name is changed. The create watcher would not be able to watch that file anymore. Also, if a file name is changed to another filename that is watched, the create file watcher would trigger the create action that is specified by the user. This should not happen since the file was only modified and not created.