

Justin (Yuting) Li

Cyber Scurity Assignment 2.3

My original reference monitor is not working because I'm using a "Try:" function to open the file but I do not end the try statement there. Personally, I'm not familiar with the python and for the try statement, and when I first wrote the assignment in 2.1, I'm thinking to "try opening this file", which later on generate this CodeUnsafe Error.

I'm also encountered another problem of the terminal is not output anything when I following the instructions in assignment 2.1, and originally I thought that is a signal of no error is showing up, but later on figure out that it might contain error in my code. For some reason, this questions happened again in my assignment 2.2, where I'm using LPopenfile() instead of openfile(), which made all of my 9 attackcase invalide.

After regenerating this reference monitor in 2.3, I findseveral things that can be updated based on what I did in assignment 2.2. What I'm doing in 2.1 is considering two different cases: EOF and write while accessing read. The solution I'm come up with is have many EOF checking statements before executing the functionality, but the code there is redundant and no efficient. I'm also include a lock and a double-check-temp value to handle the second concern, however I figure that double-check value is not useful since the algorithm included there already ensure no bugs. Hence, for my assignment 2.3, I'm more focused on fixing my bugs in EOF with a more efficient way and resolving the double-check things.

Most of the work is on the EOF, where I originally is calling a readat-function to access the maximum length of the data, but I'm also ignoring one case: since the data is not writeat immediately, maybe after writeat the length will increase, which will make the readat offset valide, and in this case I'm thinking to create two different values under self to represent the actual length and the previous length. In this way, I only need to check it is between 0 and max length, and when performing undo, I'll just use the preivous value as a reference.

Besides, I also delete the double-check value since that is redundant and not working. Moreover, I also notice another concern in this 2.3. If I'm performing the close file, and do writeat again, I should not be able to perform that, hence I also include another boolean under self to represent the file is close or not. If the file is closed, then all the functionality regarding that file will be diable: to implement that, just add a if statement to check the isClose equals to true or not and decided execute the functions or not.