Vulnerabilities and my fixes

I've identified several vulnerabilities in my code and have taken steps to address them. Let's delve into these issues and my approach to fixes in more detail:

1. Race Conditions: I've observed that attack cases involving multiple threads were failing due to race conditions. To mitigate this, you've implemented thread synchronization using the `lock.acquire()` and `lock.release()` methods. This ensures that only one thread can modify a shared resource at any given time, preventing concurrent conflicting access and ensuring atomic execution of code between the acquire and release calls.

2. Close Function: Another issue I've tackled is the ability to write to a file after a valid close operation. I've improved my code by ensuring that the close function not only marks the file as closed but also prevents any further writes to it. This is essential for maintaining data integrity and preventing data corruption.

3. Error Handling: I've recognized that error handling in my previous reference monitor was inadequately handled. I've started to work on improving this aspect, even though I've faced challenges due to the subjective nature of test cases. Effective error handling is crucial for robust security, and addressing these issues will enhance the reliability and security of your system.