Introduction

To test my code, I created a shell script to test it against each attack case provided in the assignment. There were two main issues I found in my code: Exceptions not raised in certain situations and threads causing deadlocks.

Exceptions not raised

The first vulnerability I noticed was that I was not checking if the file was closed to raise an exception in that case. To fix it, I defined the is_closed attribute to check if the file is closed before writing, raising an error if the write function is called when the file is closed.

After that, I also noticed that I was not checking if the offset was greater than 0 or if the data was a string, so I also raised exceptions in those cases, adding these checks to the initial write check along with the is_closed attribute, which will call the original function and consequently raise these errors.

I had also not checked if the given offset is greater than the size of the file after the pending write is completed. To fix this, I had to create two attributes, pending_size and pending_size_backup. These are instantiated with the initial length of the original file and updated whenever a write occurs, with pending_size_backup being the previous value of pending_size. Moreover, when an undo happens, previous_size will change to the value of previous_size_backup, so that the undo reflects the change in size of the file. After implementing these attributes, I added another check to write, raising an error if the offset is greater than pending_size.

Deadlocks when running threads.

Although I had added locks to prevent race conditions, I had not ensured that the locks would be closed in case an exception happened, which caused deadlocks in some attack cases. To fix this, I added a try-except-else block to the code, so that the pending_lock is released with or without exceptions happening, reraising the exception after if necessary.

Another mistake that was creating deadlocks was that I was that I had used two different locks by mistake, pending_lock, and rw_write, which is unnecessary and could create deadlocks on some test cases. To fix this, I simply deleted the rw_lock while keeping the pending_lock.

Conclusion

After fixing the issues mentioned above, the code was able to pass almost all of the correct test cases.