- **Improved Locking Mechanism:** In the original code, I noticed that there was a single lock (_write_lock) used for both the writeat and undo methods. This could lead to race conditions when multiple threads or processes tried to access the file simultaneously. To address this, I introduced a separate lock (self.lock) for each method (readat, writeat, undo, and close). This ensures that each method is protected from concurrent access by different threads or processes, preventing race conditions and ensuring data integrity.
- **Handling Negative Arguments:** It became apparent that the original code did not handle negative arguments for num_bytes and offset in the readat and writeat methods. This could lead to unexpected behavior or exceptions. To mitigate this, I added checks to validate these arguments, raising a RepyArgumentError if negative values are provided. This helps prevent invalid or malicious inputs from causing issues.
- **Proper File Size Tracking:** I realized that the original code did not accurately update the file size after writing data. To address this, I added code to ensure that the filesize attribute is correctly updated to reflect the size of the file after writing data. This improvement ensures that the file size tracking is accurate and up-to-date.
- **Buffer Flushing Mechanism:** I observed that the original code did not explicitly handle a buffer of pending data that needed to be written to the file. In the updated code, I introduced a buffer_flushed flag that tracks whether the buffer has been written to the file. Data is only written to the file when the buffer is explicitly flushed, preventing data loss if the program is terminated prematurely.
- **Handling Closed Files:** It was clear that the original code did not check whether the file had been closed before performing read and write operations, which could lead to exceptions or unexpected behavior. To address this, I added checks to each method to verify if the file has been closed. If an operation is attempted on a closed file, a FileClosedError is raised, ensuring that operations are only performed on open files.
- **Data Type Checks:** I noticed that the original code did not check the data type of the data argument in the writeat method. To enhance data integrity, I added type checking to ensure that the data argument is of type string. If it's not, a RepyArgumentError is raised, preventing unintended data types from being written to the file.
- **Handling Seek Past End of File:** It was evident that the original code did not handle cases where the offset was greater than the file's size when writing data. In the updated code, I added checks in the writeat method to ensure that the offset is within the bounds of the file's size. If the offset extends past the end of the file, a SeekPastEndOfFileError is raised. This improvement ensures that data is written within the file's boundaries.