Project 3: GT FileSystem

Authors: Eva Grace Bennett, Gopesh Singal

Design Justification

When creating the system, the logging mechanism was based upon the `write_t` structure; whenever a process attempts to write to a file it has opened, a `write_t` object is created, serving as an in-memory log of the write attempt. This `write_t` object holds the data it wrote to the file, as well as the data it overwrote. In doing so, if the write was chosen to be aborted, then the changes made to the in-memory representation of the file, the `file_t` object, can be undone and reverted to just before the write operation was applied. When performing `gtfs_sync_write_file`, the `write_t` is then applied to the disk memory, being applied to the log tracking the write changes and to the file in question. This is to align with the specifications outlined in note @232 on Piazza, where under Method 2 it is stated we may "sync all corresponding pending in-memory logs by moving them to on-disk logs and applying them to the file" for `gtfs_sync_write_file`.

Data Persistence

When `gtfs_sync_write_file` is called by a process, the `write_t` structure passed in is applied to the on-disk logs and file, ensuring that even if the process were to crash, the data is now recoverable from the disk memory. The process would simply need to call `gtfs_open_file` once more to receive the persisted changes. In a sense, for a change written to persist, the process needs to call `gtfs_sync_write_file` or `gtfs_clean` to update the disk memory.

Crash Recovery

Performance for Reading / Writing