《Key-Value 数据库设计》作业报告

实现方法

在 Lab 1 原有的基础上，我加入了 Redo 的日志记录功能，实现了事务恢复。

具体实现方法为，在每次 commit 时，将当前事务修改后的新值写入 Redo 文件中。这样假如出现中断情况，程序可以根据 Redo 文件判断出最后一次成功 commit 的事务是哪一个，然后根据 Redo 的内容反推得到最后一次成功 commit 之后数据库内数据的值（这部分可以直接从文件末尾读入处理，减少 IO 时间开销），并且可以接着执行剩余的未执行的事务。同时由于数据每更新一次都会储存一份在 Redo 文件内，也可以做到数据的持久化。

实验结果

因为这里用的是 Lab 1 的测试工具，所以程序还额外处理了输出结果的重做功能，即未 commit 成功的事务的 `READ` 语句全部作废，从最近的一次 END 后开始输出。

实验结果为在多次强行中断程序的情况下，程序仍然能做到恢复未完成的事务并接着执行后面的事务，且输出全部正确。具体见网络学堂提交的视频。

不足

目前方法只适用于数据量较小的情况，当数据量大的话 Redo 文件可能会变得非常大，这时候我觉得可以定期清理 Redo，因为 Redo 前面的数据对后面的事务恢复其实没有很大的帮助。

代码文档

- `main.cpp`：主程序（含注释）
- `inputs/*.txt`：测试样例

可直接执行 `make` 进行编译。

执行 `./main x` 运行多线程版本，线程数为 `x`，若该参数为空则默认为 1。