# Internals Project #1 Documentation

Jiaqi Liu, Chen Zou

[Jliu6@wpi.edu](mailto:Jliu6@wpi.edu), czou@wpi.edu

## Disk File I/O

**Assumptions**

1. The database is small enough to be read into main memory and wrote into disk in total with satisfactory speed.
2. All values in database (except for metadata) are in byte format.
3. All keys in database are in string format.

**Design Decisions**

1. Every times a client tries to manipulate database, the cs542.db file will be firstly read into the main memory as a whole and be synchronized into disk when a transaction ends.
2. When ever a transaction fails, a roll-back function is invoked.
3. To serialize data, the cPickle module was used. The protocol version one ( the old binary format) was chosen to prevent the sharp increase of the size of database.
4. To establish a more user-friendly and intuitive way of putting, modifying and deleting data in database, we rewrite some methods of dictionary in python using the interface UserDict.DictMixin.
5. Operations on new ‘dictionary’ are further wrapped into Put() and Get() functions.

## Threads Control

**Assumption**

1. It is possible that there are several clients roughly at same time to operate same data resource in the database, so that some of them will get a outdated data or nonexistent data.
2. The only way to prevent this kind of problem is that set lock to certain data when one of clients first get the accessibility of the data.

**Design Decision**

1. To implement concurrency, we introduce three threads, which are going to get, put, and remove the same database recourse. Every time one of the three threads is get the ability of operate the data, the data will be locked, so that other two thread cannot continue but waiting. Only can one of the rests continue, when the lock is released.
2. In the process of finish the three movements, if there is no element, which is going to be gotten and be removed, the system will show information about the action is failed.

## Program Operation

1. To run shell program, users should input “sh show.sh main.py”
2. the result includes firstly the process of file I/O with the sequence which is showed in fragmentation requires; secondly, the process of concurrency with four clients.