

Description of Lab5

XIE JIE

516030910042

1.Design decisions

(1)Deadlock Detection: every time a transaction try to get a lock, I use DFS to search for the cycle if the lock can be got. It says, the transaction can only get the lock after it is committed/aborted. If it is, abort the transaction. To avoid a long-time detection, if the chain is too long, just abort the transaction.

(2)Locking granularity is on page. That is I get the read/write lock on page. The lock is based on transaction, not thread. So it is a little hard to keep the writelock exclusive with other writelocks and readlocks.

(3)Using fair lock in Java to keep sequence of acquiring lock.

(4)Fix some bugs in lab4.

2.Some methods in BufferPool are restrict by synchronized to avoid competition when acquiring and releasing lock.