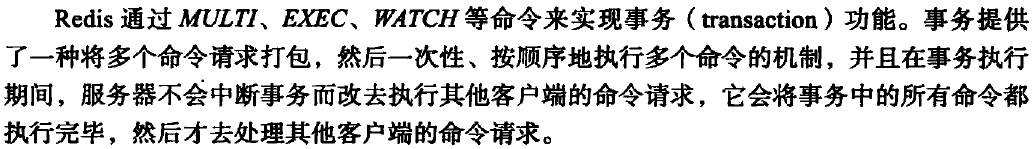
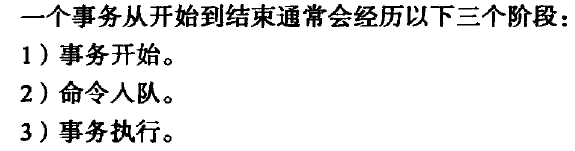
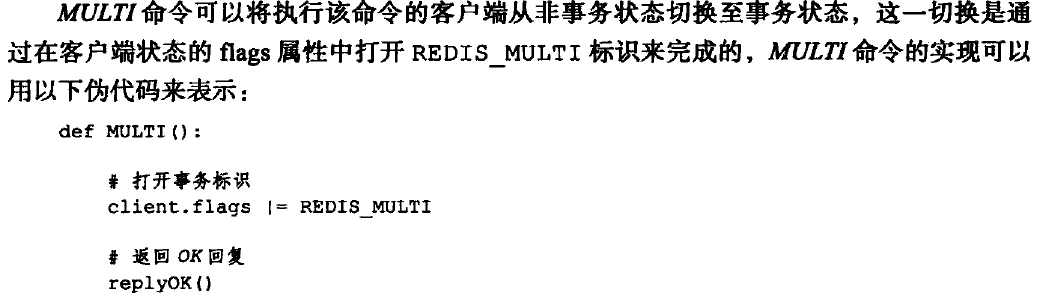
事务



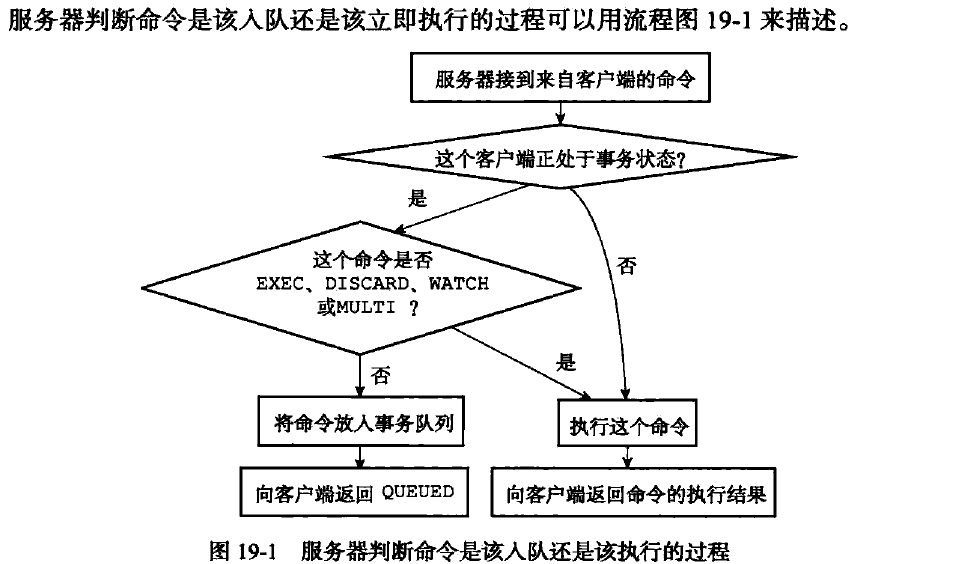
# 事务的实现



## 事务开始



## 命令入队



## 事务队列

typedef struct redisClient{

//…

//事务状态

multistate mstate; /\*MULTI/EXEC state\*/

//…

}redisClient;

//--------------------------------------

typedef struct multistate{

multiCmd \*commands; //事务队列，FIFO

Int count; //已入队命令数

}multistate;

//--------------------------------------

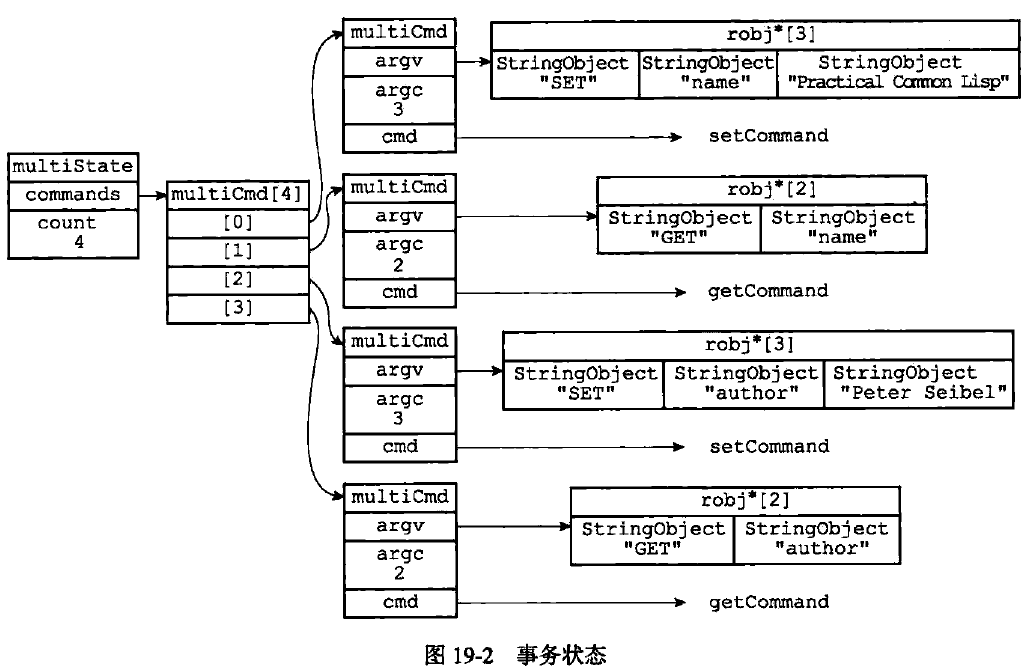
typedef struct multiCmd{

robj \*\*argv; //参数

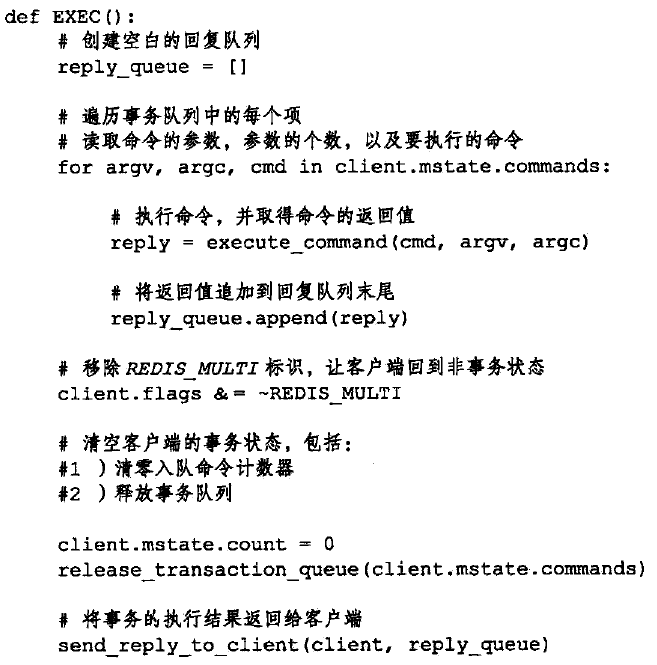
int argc; //参数数量

struct redisCommand \*cmd; //命令指针

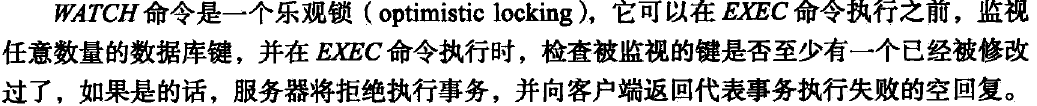
}



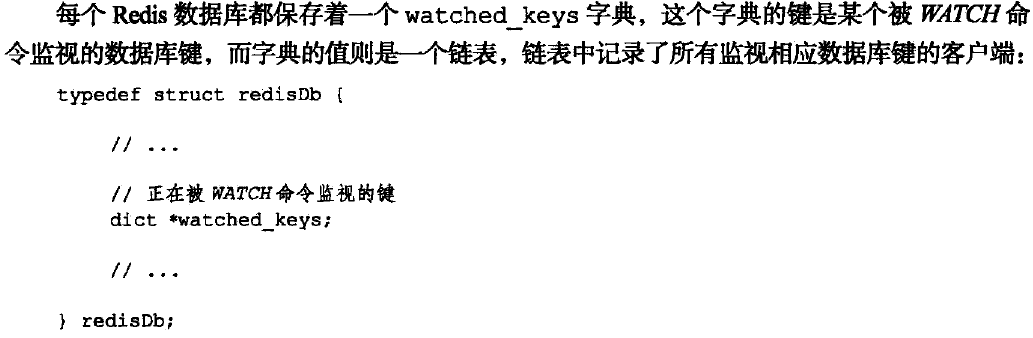
## 执行命令



# WATCH命令的实现

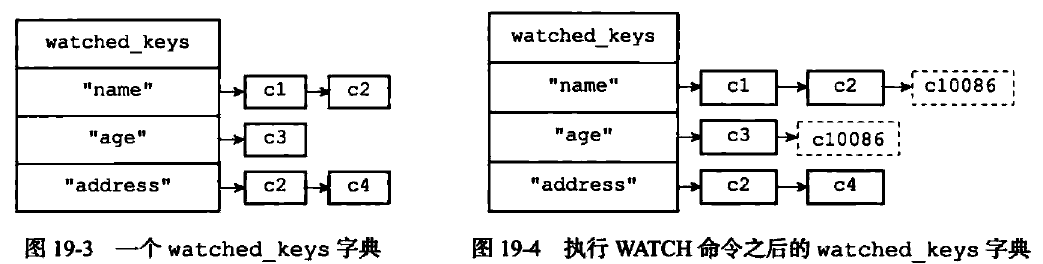


## 使用WATCH监视数据库键

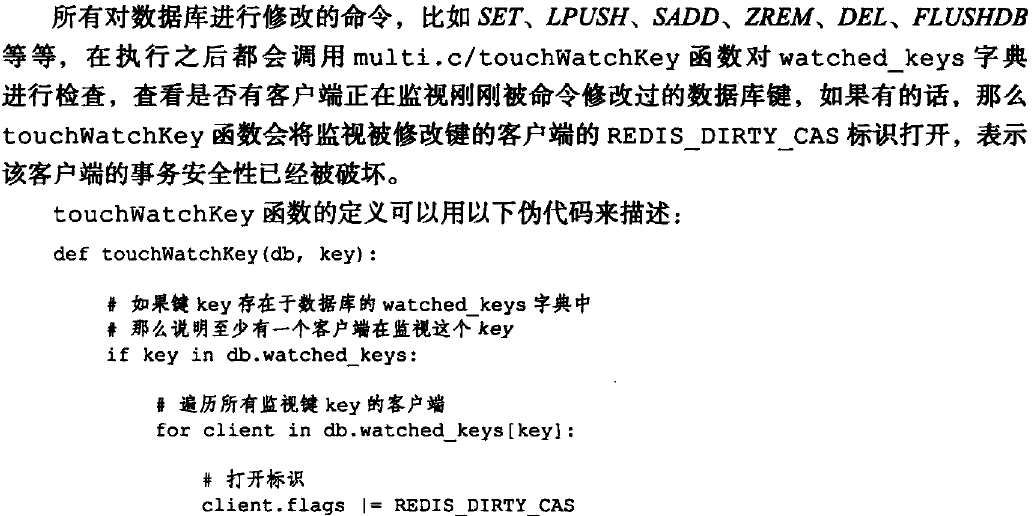


客户端执行命令：



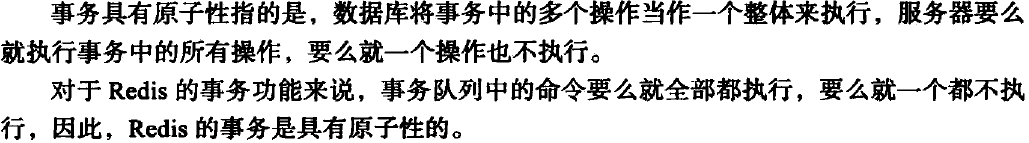


## 监视机制的触发

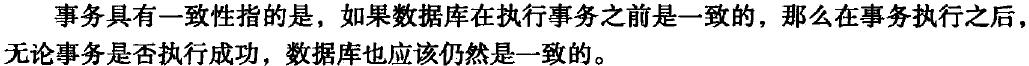


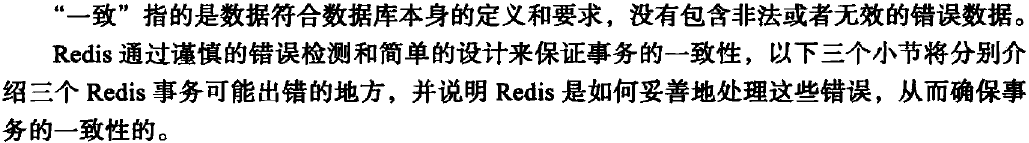
# 事务的ACID性质

## 原子性

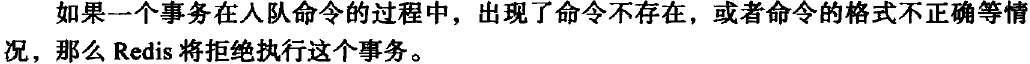


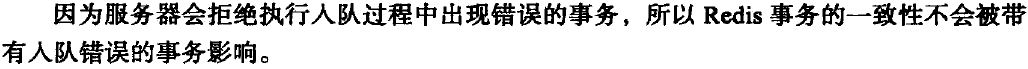
## 一致性



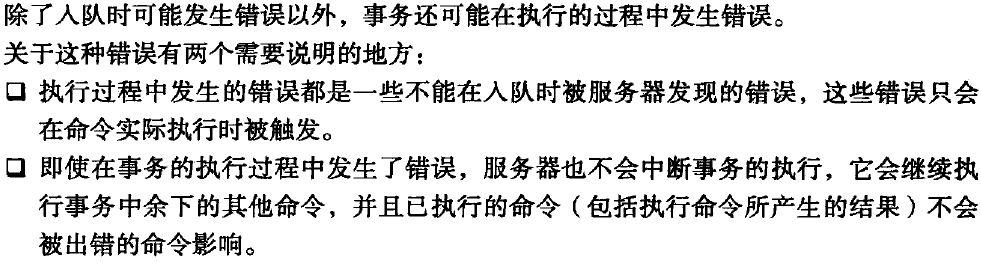


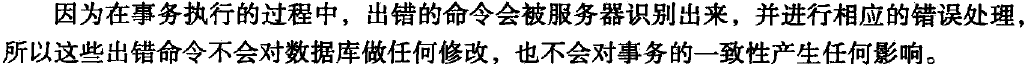
### 入队错误



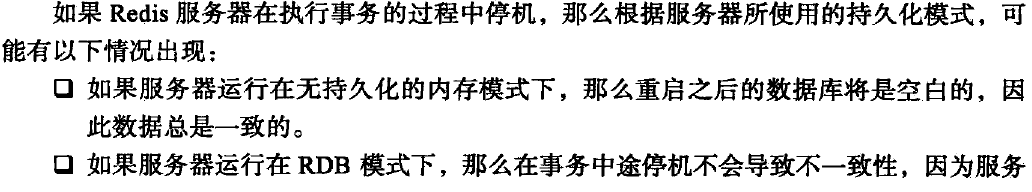


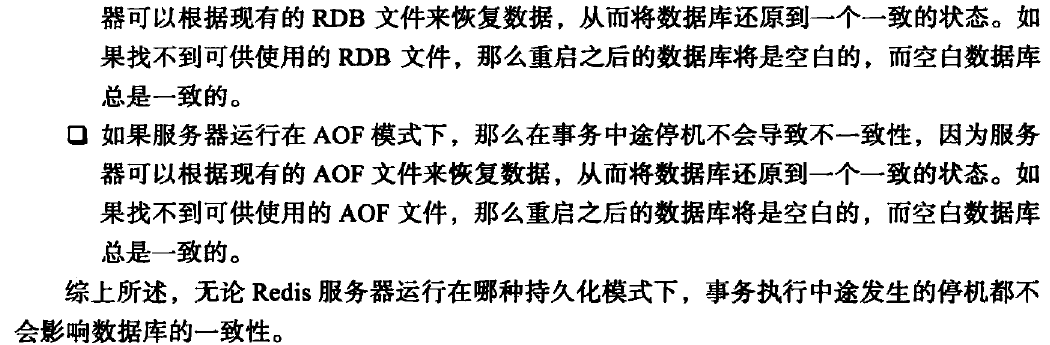
### 执行错误



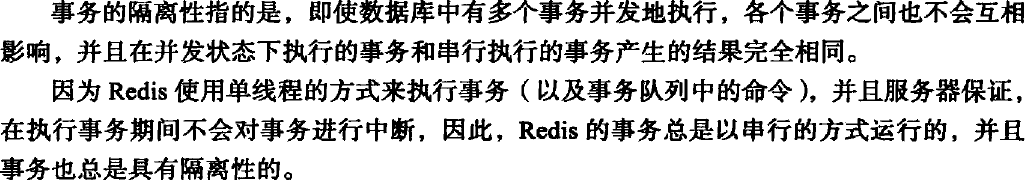


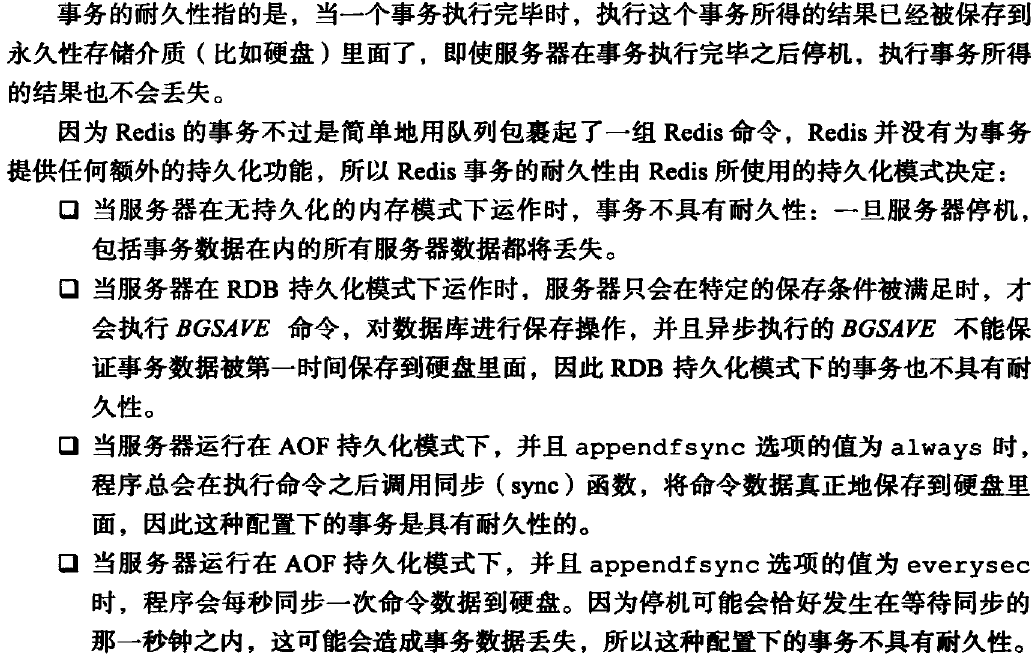
### 服务器停机

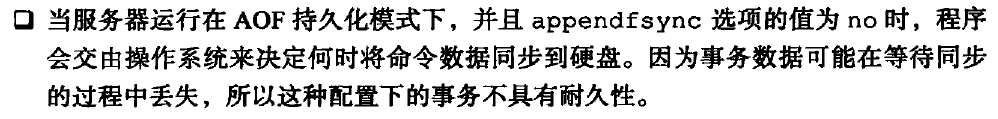




## 隔离性







## 持久性