- **四、持久化操作**
  - RDB
    - RDB和AOF差异
      - 共同：redis服务器通过RDB或AOF文件还原数据库状态。
      - 不同：
        - 本质区别：
          - RDB通过保存redis服务器中数据库的所有键值对记录数据库状态；
          - AOF通过保存redis服务器执行的所有写命令记录数据库状态；
        - AOF文件的更新频率通常比RDB文件更快，当redis服务器开启了AOF持久化功能时，优先通过AOF文件还原数据库状态；只有当AOF持久化功能处于关闭时，才会通过RDB还原数据库状态。
    - 1. RDB文件的创建与载入
      - 通过执行SAVE命令或BGSAVE命令生成RDB文件，SAVE命令或BGSAVE命令都是通过调用rdbSave()函数来实现，两个命令的区别是SAVE命令会阻塞当前服务器进程，而BGSAVE命令会创建当前服务器进程的子进程。
      - redis服务器在启动时会对每个被载入的文件进行检查，如果某个文件的前五个字节为"REDIS"，就认为该文件为一个RDB文件。并且当AOF持久化功能处于关闭时，通过该文件还原数据库状态。
      - 补充：
        1. 子进程在执行BGSAVE命令期间，如果接收到客户端发送的SAVE命令，则拒绝执行（因为这两条命令都是通过调用rdbSave函数实现的，会产生竞争条件）；
        2. 子进程在执行BGSAVE命令期间，如果接收到客户端发送的BGSAVE命令，同样拒绝执行（也是因为会产生竞争条件）；
        3. 子进程在执行BGSAVE命令期间，如果接收到客户端发送的BGREWRITEAOF命令，会被阻塞，直至BGSAVE命令执行结束（虽然不会产生竞争条件，但是两个子进程同时执行，会导致不必要的内存写入动作）；
        4. 子进程在执行BGREWRITEAOF命令期间，如果接收到客户端发送的BGSAVE命令，会被直接拒绝执行；
    - 2. RDB文件的自动间隔性保存
      1. 因为BGSAVE命令可以在不阻塞当前服务器进程的情况下执行，redis服务器可以定期的执行该命令以此完成对RDB文件的更新。通过redis服务器提供的服务器配置选项save选项设置保存条件。
      2. save保存条件对应保存在redis服务器状态redisServer结构中的save数组，该数组中的每个元素包含两个属性：1、seconds秒数；2、changes修改次数。
      3. redis服务器状态redisServer结构中还有另外两个属性辅助BGSAVE命令的触发：1、dirty计数器，用于记录数据库中键被修改的次数；2、last_update用于记录上一次执行BGSAVE命令的时间。
      4. redis服务器的周期性时间事件serverCron函数默认每100毫秒执行一次，用于对正在运行的服务器进行维护，其中一项检查工作就是检查save选项所设置的保存条件是否满足，如果满足则触发BGSAVE命令的执行。
    - 3. RDB文件内存结构
      - 完整的RDB文件由五部分组成：
        1. REDIS。前五个字节为固定内容，保存"REDIS"这五个字节，redis服务器启动时会自动检测载入的文件是否为一个RDB文件；
        2. db_version。记录RDB文件的版本号；
        3. databases。记录该服务器中的所有数据库；
        4. EOF。标志RDB文件正文内容的结束；
        5. CHECKSUM。校验和，用于正确性校验，由redis服务器自动计算，无需关心；
      - RDB文件的第三部分保存服务器中的所有数据库，每个数据库由三部分组成：
        1. select_db。告知redis服务器，接下来会读入一个数据库号码；
        2. db_number。数据库号码；
        3. key_value_pairs。数据库中的所有键值对；
      - 每个数据库的key_value_pairs保存了所有键值对，每个键值对由三部分或五部分组成：
        - 不带过期时间的键值对：
          1. type。键值对中value值的数据类型及编码方式；
          2. key。键值对的key值；
          3. value。键值对的value值；
        - 带有过期时间的键值对：
          1. expire_ms。告知redis服务器，接下来会读入一个以毫秒为单位的过期时间；
          2. ms。键值对的过期时间；
          3. type。键值对中value值的数据类型及编码方式；
          4. key。键值对的key值；
          5. value。键值对的value值；
  - AOF
    - 1. AOF持久化功能的实现
      - 实现：
        1. 命令追加 —— redis服务器每执行一条命令，就会将该命令发送至服务器状态redisServer结构中的aof_buf缓冲区中；
        2. 文件写入
        3. 文件同步
      - 大多数现代操作系统为了优化文件写入效率，在执行write系统调用时，会先将需要写入的内容保存到一块内存缓冲区中，等到缓冲区被填满或等待的时间超过时限后再将其刷新到磁盘上。
      - redis服务器进程可看做是一个事件循环，每一个循环结束之前，都会调用flushAppendOnlyFile函数决定是否将aof_buf缓冲区的内容写入并同步到AOF文件中，该函数的具体行为由服务器提供的配置选项appendFileSync决定，而appendFileSync是一个枚举，有三个枚举值：
        1. always。表示总是将aof_buf缓冲区中的内容写入并同步到AOF文件中；
        2. everysec。默认值，表示总是将aof_buf缓冲区中的内容写入到AOF文件，如果距离上次同步已经超过了一秒钟，则再次对AOF文件执行同步；
        3. no。表示总是将aof_buf缓冲区中的内容写入到AOF文件，具体何时同步由操作系统自行决定；
      - redis服务器的appendFileSync配置选项默认值为everysec，即redis服务器最多只丢失最近一秒中执行的所有写命令。
    - 2. AOF文件载入
      - redis规定所有的redis命令必须在客户端上下文中执行，而AOF文件中的redis命令直接来源于AOF文件，所以载入AOF文件的第一步：创建一个没有网络连接的伪客户端；
      - 第二步：从AOF文件分析并读取一条记录交给伪客户端执行；
      - 不断重复第二步直至AOF文件中的所有命令均已被执行完毕，redis通过AOF文件还原数据库状态操作顺利完成。
    - 3. AOF文件重写
      - 通过执行BGREWRITEAOF命令实现AOF文件重写功能。该命令和BGSAVE命令一样都会创建当前服务器进程的子进程。
      - BGREWRITEAOF命令
        - 优点：
          1. 当前服务器进程仍然可以处理客户端发送的命令请求；
          2. 子进程通常带有当前服务器进程的数据副本，使用子进程而不是线程，可以在不使用锁的情况下保证数据的安全性；
        - 缺点：子进程执行期间当前服务器进程执行的写命令无法被重写到AOF文件中；
        - 缺点的解决方案：BGREWRITEAOF命令执行期间，当前服务器进程执行的所有写命令不仅会发送到aof_buf缓冲区中，还会被保存到aof重写缓冲区中；
      - 子进程完成AOF文件重写后，会向当前服务器进程发送一个信号。
      - 当前服务器进程接收到该信号后，调用信号处理器函数，该函数会阻塞当前服务器进程，然后将aof重写缓冲区中的所有写命令写入到AOF文件中。然后将旧的AOF文件删除并释放所占用的存储空间，对新的AOF文件进行改名，替代旧的AOF文件。
      - 至此，AOF文件重写操作完成，当前服务器进程可以继续处理客户端发送的命令请求。