# 分布式中间件考试试题

1. **选择题**

1：ABCDE

2: A

3:B

4.D

5.AB

6.A

7.AB

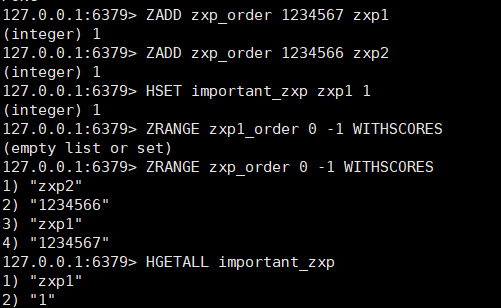
8.C

9.ABCD

10.D

**二、编程题**

1.代码截图：



2.代码截图：

|  |
| --- |
| 60秒过后： |

**三、简答题**

**1.答：Redis支持以下数据类型：**

**字符串（String）：**最基本的数据类型，值可以是字符串、整数或浮点数。

可以用SET、GET、INCR、DECR、APPEND等。

最大存储容量为512MB。

**哈希（Hash）：**用于存储对象，值以键值对形式存储。

可以用HSET、HGET、HGETALL、HDEL等。

适用于存储对象的属性。

**列表（List）：**有序的字符串列表，可以从两端插入和移除元素。

可以用LPUSH、RPUSH、LPOP、RPOP、LRANGE等。

常用于消息队列。

**集合（Set）：**无序的字符串集合，集合中的元素是唯一的。

可以用SADD、SREM、SMEMBERS、SINTER等。

适用于需要存储唯一元素的场景。

**有序集合（Sorted Set）：**和集合类似，但每个元素都会关联一个分数，元素按分数排序。

可以用ZADD、ZRANGE、ZREM、ZRANGEBYSCORE等。

适用于排行榜、带权重的队列等。

**地理位置（Geo）：**基于有序集合实现，用于存储地理位置信息并进行地理位置操作。

可以用GEOADD、GEORADIUS、GEODIST等。

适用于地理位置相关的应用。

**2.答：**原因：简单性：单线程模型使代码逻辑简单，避免了多线程编程中的锁定和竞争条件问题。

性能：大多数Redis操作是在内存中完成的，单线程可以通过非阻塞I/O和事件循环机制高效地处理大量请求。

一致性：单线程避免了并发写操作带来的数据一致性问题。

优点：易维护：代码逻辑简单，没有多线程编程的复杂性。

高效：对于大部分操作（尤其是读操作），单线程模型性能非常高。

数据一致性：避免了并发写操作带来的数据一致性问题。

缺点：CPU瓶颈：不能充分利用多核CPU的优势，对于CPU密集型任务，单线程可能成为瓶颈。

阻塞问题：某些命令（如慢查询）可能会阻塞整个Redis实例，影响其他操作的执行。

**3.答：**Redis提供两种主要的持久化机制：RDB快照和AOF日志。

**RDB快照：**定期将内存中的数据生成快照并保存到磁盘。手动执行SAVE或BGSAVE命令，或者通过配置自动触发。

优点：适合大规模数据备份，恢复速度快。

缺点：数据持久化过程中可能会丢失最近几分钟的数据。

**AOF日志：**将每个写操作记录到日志文件中（追加模式）。通过配置持久化策略（每秒、每修改、手动）。

优点：更高的数据安全性，数据丢失量较少。

缺点：日志文件可能会变得很大，需要定期重写压缩。

**4.答：事务：**Redis事务通过MULTI、EXEC、DISCARD和WATCH命令实现。

事务保证一组命令按顺序执行，且在执行过程中不会被其他命令打断。

MULTI开启事务，EXEC提交事务，DISCARD取消事务，WATCH用于乐观锁机制。

**用途：**保证一组操作的原子性，即要么全部执行成功，要么全部不执行。

实现复杂的原子操作，确保数据一致性。

通过WATCH实现乐观锁，防止并发修改导致的数据不一致问题。

**5.答：发布订阅模式：**是一种消息通信模式，允许消息的发送者（发布者）不直接将消息发送给特定的接收者（订阅者），而是将消息发布到特定的频道（channel）。订阅者订阅特定的频道，并接收发布到该频道的所有消息。

**操作命令：**PUBLISH：将消息发送到指定频道。

SUBSCRIBE：订阅一个或多个频道，接收其发布的消息。

PSUBSCRIBE：订阅符合给定模式的所有频道。

**用途：**实现实时消息通知。

构建发布订阅系统，例如聊天室、实时数据推送等。