1. 事件
2. 概述

Redis服务器是一个事件驱动程序，服务器需要处理两类事件：

1. 文件事件

Redis服务器通过套接字和客户端进行连接，而文件事件就是服务器对套接字操作的抽象，服务器与客户端的通信会产生相应的文件事件，而服务器通过监听并处理这些事件完成网络通信操作。

1. 事件时间

服务器中的一些操作如serverCron函数需要在给定的时间点执行，时间事件就是服务器对这类定时操作的抽象。

1. 文件事件
2. 概述

Redis基于Reactor模式开发了自己的网络时间处理器，称为文件事件处理器，使用I/O多路复用程序来同时监听多个套接字，并根据套接字目前执行的任务为套接字关联不同的事件处理器。当被监听的套接字准备执行连接应答， 读取，写入，关闭等操作时，与操作相对应的文件事件就会产生，这时文件事件处理器就会调用套接字之前关联好的事件处理器来处理这些事件。

文件事件处理器以单线程的方式运行，使用I/O多路复用程序监听多个套接字，实现了高性能的网络通信模型，保持了单线程设计的简单性。

1. 文件事件处理器的构成

文件事件由四部分组成：套接字，I/O多路复用程序，文件事件分派器，事件处理器。

1. 套接字

文件事件是对套接字操作的抽象，每当一个套接字准备执行操作，就会产生一个文件事件，因为一个服务器会连接多个套接字，所以多个文件事件可能会并发地出现。

1. 多路复用

监听多个套接字，并向文件事件分派器传递那些产生了事件的套接字，虽然文件事件会并发地产生，但是复用程序会将这些事件放到一个队列中有序分发。当上一个套接字产生的事件被处理完毕之后，才会继续向分派器传送下一个套接字。

1. 分派器

接受套接字，根据套接字产生的事件类型，调用相应的事件处理器。

1. 处理器

服务器会为执行不同任务的套接字关联不同的事件处理器，这些处理器是一个个函数，定义了应该执行的操作。

1. I/O多路复用的实现

通过包装常见的select, epoll, evport和kqueue这些I/O多路复用函数库来实现的。

1. 事件的类型
2. AE\_READABLE事件

当套接字变得可读时，或者有新的可应答套接字出现时，套接字产生AE\_READABLE事件。

1. AE\_WRITEABLE事件

当套接字变得可写时，产生该事件。

I/O多路复用程序会同时监听这两种事件，并优先处理读事件。

1. 文件事件的处理器
2. 连接应答处理器

用于对连接服务器监听套接字的客户端进行应答，当服务器初始化时，会将这个连接应答处理器和服务器监听套接字的读事件关联起来，只要有客户端通过connect函数连接服务器监听套接字，套接字就会产生读事件，引发连接应答处理器执行。

1. 命令请求处理器

负责从套接字中读入客户端发送的命令请求内容，在客户端连接服务器的整个过程中，服务器会一直为客户端套接字的读事件关联命令请求处理器。

1. 命令回复处理器

负责将服务器执行命令后得到的命令回复通过套接字返回给客户端。

1. 时间事件
2. 概述

时间事件分为两类，一类是定时时间，一类是周期性事件。一个时间事件由三个属性组成：

全局唯一ID，毫秒精度的UNIX时间戳，记录了时间事件的到达时间，时间事件处理器。

1. 实现

服务器将所有时间事件都放在一个无序链表中，当时间事件执行器执行时，它就遍历整个链表，查找所有已到达的时间事件，并调用相应的事件处理器。

1. 事件的调度

对于文件事件和时间事件，服务器需要对这两种事件进行调度，由于时间事件是固定在某个事件执行的，而文件事件是随机执行的，因此为了避免对时间事件频繁的轮询操作，总是会先处理文件事件，再处理时间事件，两个事件的处理都是同步的。