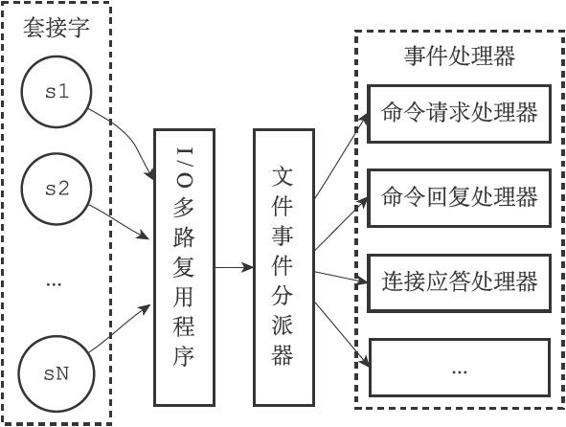
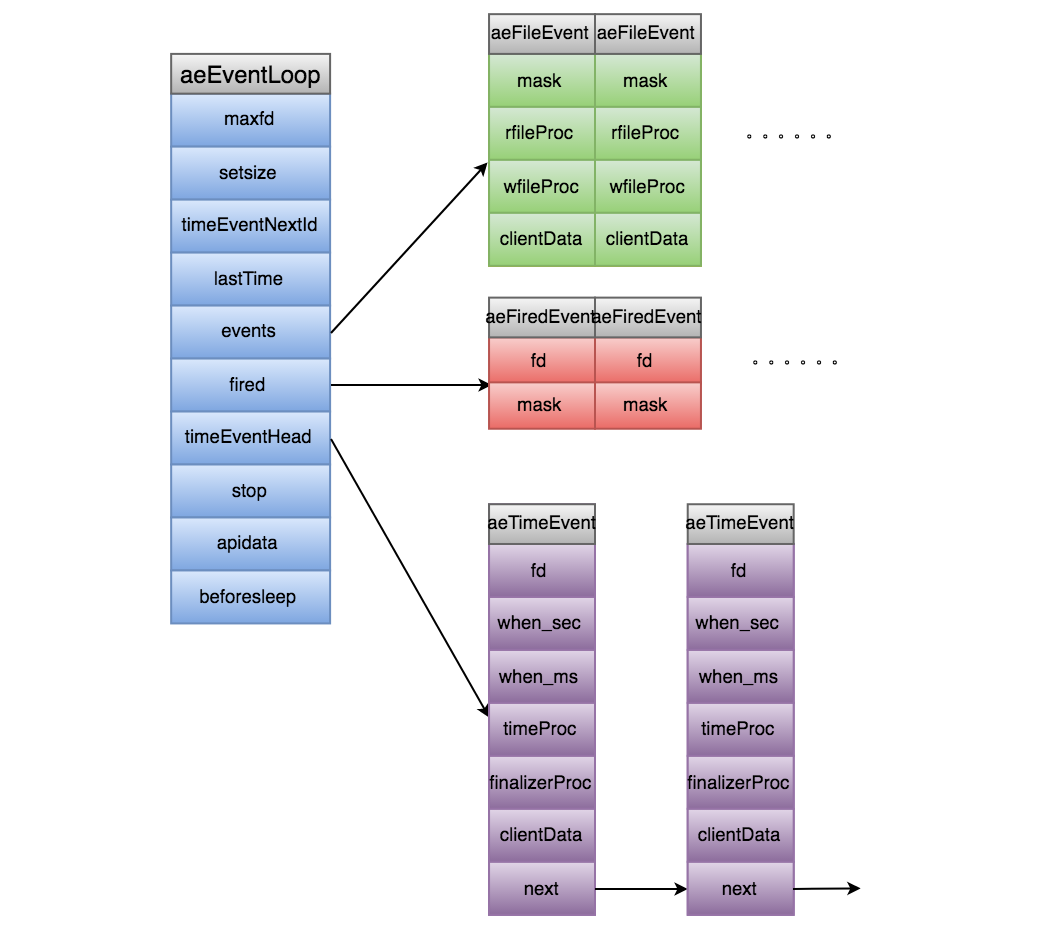
Redis时间循环的主要代码位于ae.c文件。Redis基于Reactor模式开发了自己的文件事件处理。Redis使用I/O多路复用程序来同时监控多个套接字。当被监听的套接字准备执行读取、写入、关闭等操作时，就会产生相应的文件事件，redis会选择相应的文件处理器来处理时间。



事件循环的核心部分是aeEventLoop，下图为数据结构：



1. 初始化

sever.c/initServer函数调用ae.c/aeCreateEventLoop函数创建event loop对象，对事件循环进行初始化。

以epoll为例，ae.c/aeCreateEventLoop函数创建并初始化全局时间循环结构，调用ae\_epoll.c/aeApiCreat函数创建epoll的fd，以及要监听的epoll\_eventae。ApiCreate会初始化这些属性，并将aeApiState结构存放到eventLoop->apidata。

1. 事件的添加

Redis服务器需要处理两类事件：文件事件和时间事件。

文件事件通过ae.c/aeCreatFileEvent函数添加，根据注册的fd(套接字)，获取其event，设置属性，并调用ae\_epoll.c/aeApiAddEvent函数添加至I/O多路复用程序的监听范围内。

时间事件通过ae.c/aeCreatTimeEvent函数添加，接受一个毫秒数milliseconds和一个事件事件处理器proc为参数。

1. 事件触发

在server启动过程中，会调用aeMain启动事件处理循环。

void aeMain**(**aeEventLoop **\***eventLoop**)** **{**

eventLoop**->**stop **=** 0**;**

**while** **(!**eventLoop**->**stop**)** **{**

**if** **(**eventLoop**->**beforesleep **!=** **NULL)**

eventLoop**->**beforesleep**(**eventLoop**);**

aeProcessEvents**(**eventLoop**,** AE\_ALL\_EVENTS**|**AE\_CALL\_AFTER\_SLEEP**);**

**}**

**}**

函数内部为一个while循环，不断调用aeProcessEvents函数，等待事情触发。调用前会调用beforesleep函数，处理异步任务。

aeProcessEvents函数首先会处理时间事件，然后是文件事件，下面介绍这个函数的实现。

首先，声明变量记录处理的事件个数processed，以及触发的事件numevents。flags表示此轮需要处理的事件类型，如果不需要处理时间事件和文件事件直接返回。

接着，找出最近的时间事件，计算出文件事件的阻塞时间。如果最近的时间事件存在，则根据离当前时间的时间差得出文件事件的阻塞时间；如果不存在，则根据 AE\_DONT\_WAIT 是否设置来决定是否阻塞，以及阻塞的时间长度。

之后，阻塞等待就绪的文件事件，处理文件事件，阻塞时间由 tvp 决定。

然后处理已就绪的文件事件。获取的已就绪事件存储在fired中。如果文件事件绑定了读/写事件，进行相应的处理。其中rfileProc和wfileProc就是在文件事件被创建时传入的函数指针。

