Bug编号：240705

标题：“epoll”在关闭套接字后多次触发引起程序崩溃

**发现日期：**2024-06-18

**影响范围：**网络模块，设计服务器与客户端的连接管理

**状态：**已解决

Bug描述：  
**·现象：**在进行多次重新建立PPP链路后，当服务器主动断开连接会导致程序崩溃。但在PPP重新建立链路前，服务器主动断开连接不会导致该Bug产生。（待解决）

**·重现步骤：**

1. 缩短PPP定时器事件，方便重现Bug产生条件。
2. 客户端连接到服务器。
3. 分别在重新PPP建立前和PPP后服务器关闭连接，重现两种不同情况。
4. 观察程序运行情况。

**·预期结果：**‘epoll’应该在对端关闭连接时触发一次，并继续阻塞在epoll\_wait函数位置。

**·实际结果：**在重新PPP前，符合预期结果。在多次PPP后，对端服务器fd又重新触发，多次触发关闭事件，导致程序崩溃。

## 分析过程：

**·初步分析：**怀疑在对端关闭连接后，‘epoll’没有正确处理已关闭的文件描述符。

**·日志与数据：**查看程序日志，发现多次PPP后‘epoll’在关闭套接字后仍多次触发事件。

**·排查过程：**

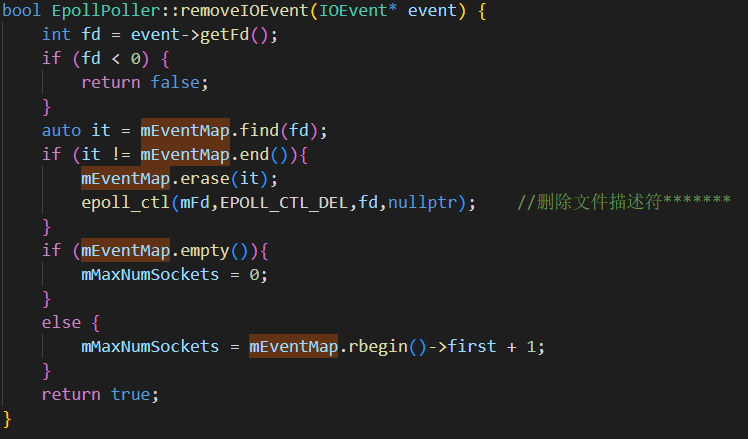
1. 检查’epoll’事件处理逻辑。EpollPoller类通过维护一个‘eventmap‘，其中键为文件描述符，值为‘IOEvent‘函数指针，利用该关联容器来找到对应的需要处理和执行的函数，采用‘epoll‘监听其键，实现事件处理逻辑。
2. 检查与服务器连接的逻辑。当与服务器建立连接时，会建立‘ISRConnection‘对象管理该连接，并且会实例化‘ISRConnection‘父类‘TCPConnection‘，在该父类中，创建了‘IOEvent‘事件，并添加到‘epoll‘中进行监听，利用该文件描述符和服务器进行通信。
3. 检查与服务器断开连接的逻辑。当服务器断开连接时，执行连接断开的回调函数，首先是在内部维护的‘servermap‘映射表中删除该连接记录，并在重连表中添加该服务器地址，然后‘delete‘该‘conn‘对象，此时会触发子类和父类的析构函数。在父类的析构中，会调用removeIOEvent来删除该IOEvent，并关闭该IO事件的文件描述符。
4. 排查EpollPoller中的对于eventmap的管理和epoll监听事件的管理。EpollPoller类中采用addIOEvent新增IO事件，调用removeIOEvent删除IO事件，在删除时仅删除了成员变量EventMap中的映射，此时没有在epoll监听事件中同步删除对应的Fd，当连接父类中关闭该IO事件的描述符时会触发可读，使得epoll监听到该可读事件，但此时 eventMap 中已经没有该 fd，双方不匹配，引发该Bug。

## 解决方案：

**·根本原因：**程序在处理’epoll’事件时，没有正确和关闭文件描述符，导致程序崩溃产生。Epoll监听事件和eventmap不同步。

**·解决方案：**修改removeIOEvent函数，在删除IO事件时同步删除epoll监听事件中的文件描述符。

**·代码变更：**



## 待解决问题：

1. 为什么没有重新PPP时，不会引起该bug产生。（未找到原因）
2. 为什么会二次触发fd=8的可读事件。（好像不是由于close（mFd）引起的）。
3. 修改代码后，不会产生二次触发问题。因为在第一次断开时就已经不会再监听该fd了。
4. 频繁ppp会导致和服务器的连接断开，测试时，一分钟ppp一次，三次以上就会断开连接。