在上一节中讲了Pipeline的运行时创建。

我们会打开一个已经包含音视频的文件。

而负责打卡这个容器文件的element称为demuxer（对应于ffmpeg的AVFormat），常见的容器格式为MKV、QT、MOV、Ogg和ASF、WMV、WMA等等。

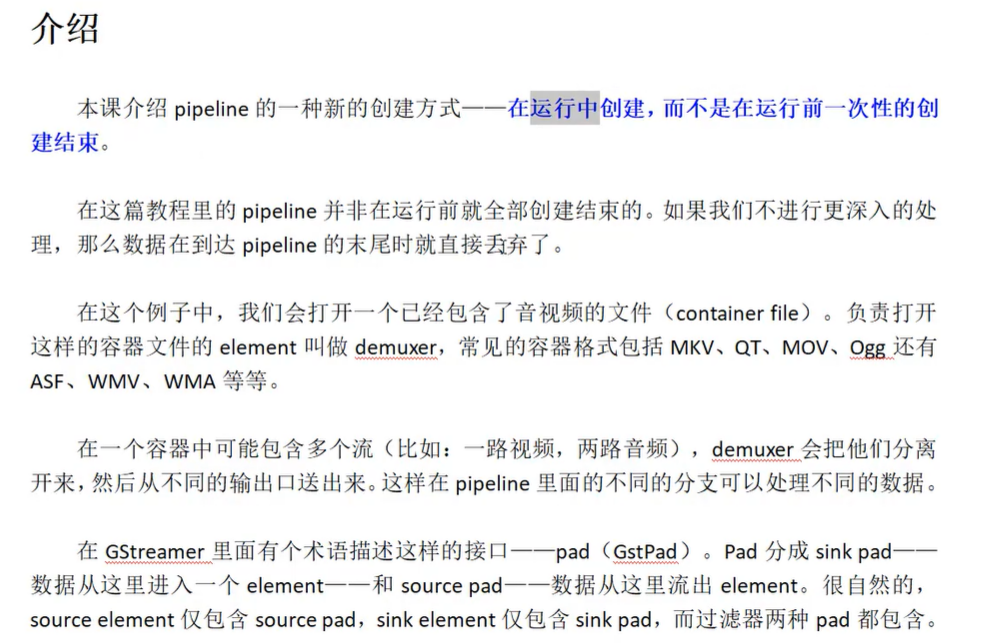
一个容器可能会包含多个流（对应于ffmpeg的AVStream）（如：一路视频，两路音频），demuxer就会把他们分离开，然后从不同的输出口送出来，这样在Pipeline里面的不同的分支就可以处理不同的数据。

Gstreamer里面有一个pad（GstPad）,Pad分成sink pad（数据从这里进入一个element）和source pad（数据从这里流出element）。

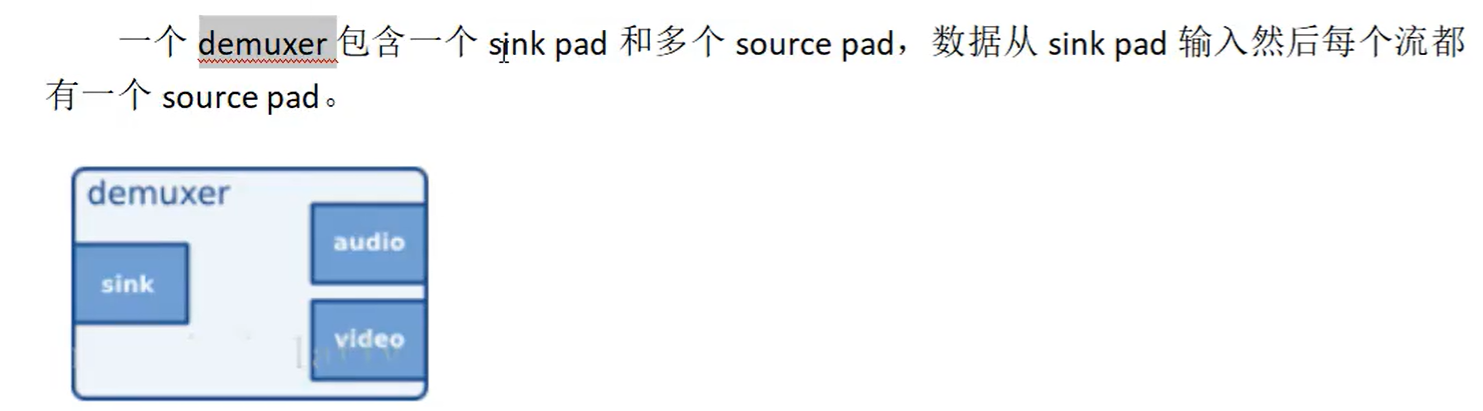
Source element仅包含source pad，sink element 仅包含sink pad，而过滤器两种pad都包含。

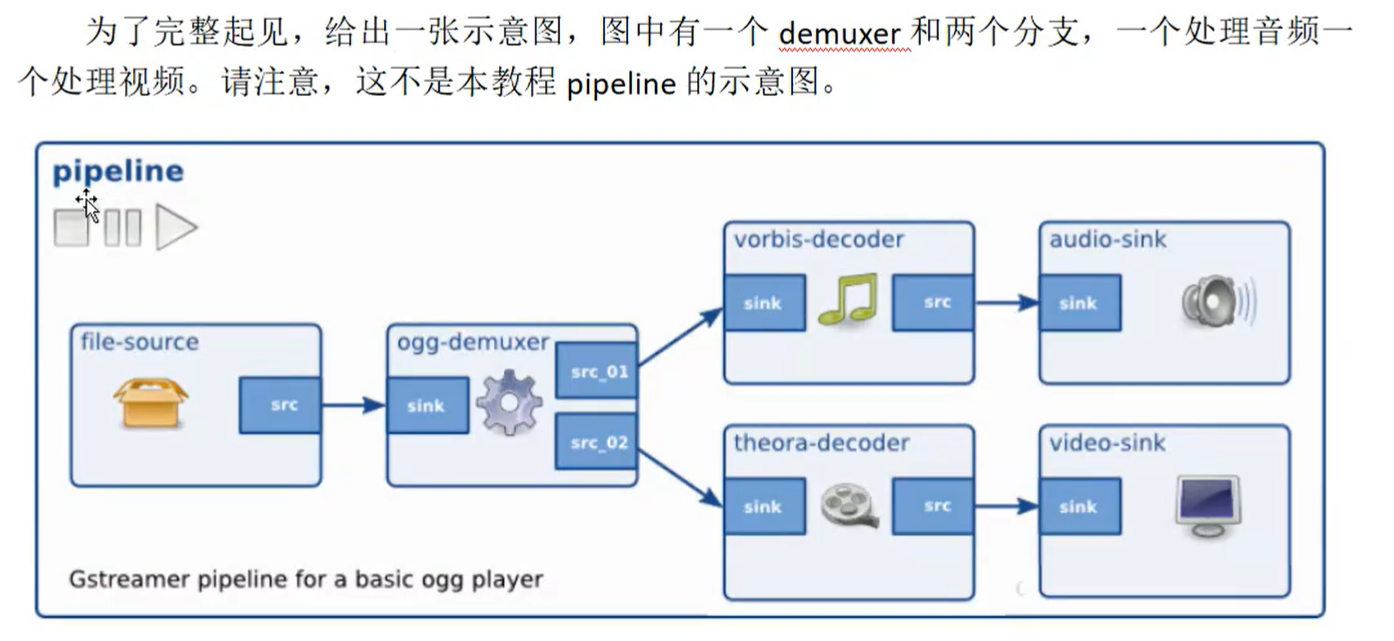
一个demuxer包含一个sink pad和多个source pad，数据从sink pad输入，然后每个流都有一个source pad。

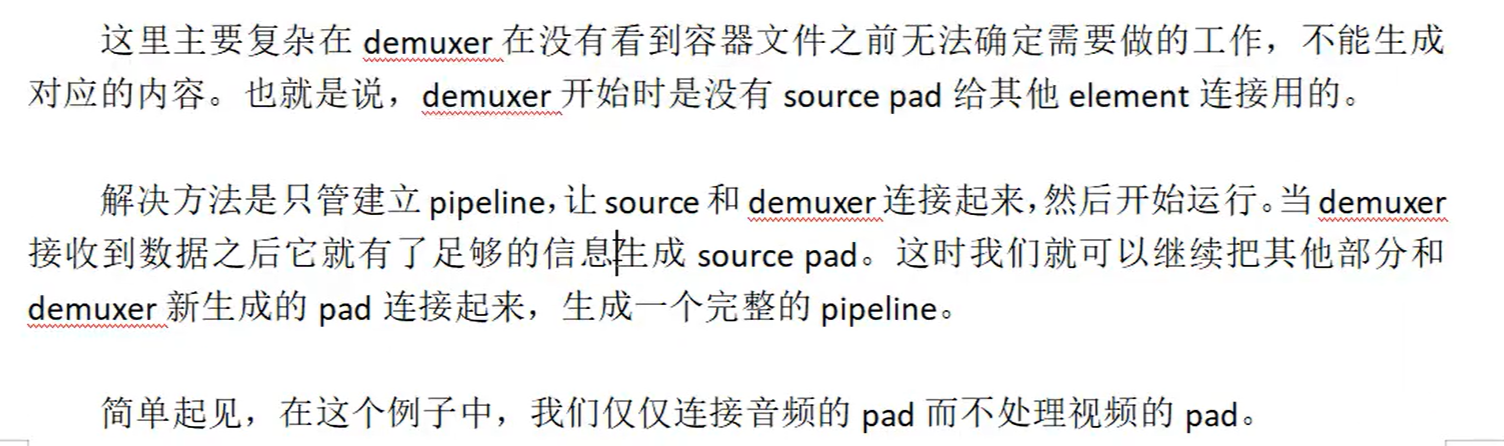
比如：一个解析mp4的demuxer有一个入口（入口传递mp4文件/网址）,然后有多个出口（出口对应音频流、视频流等。）

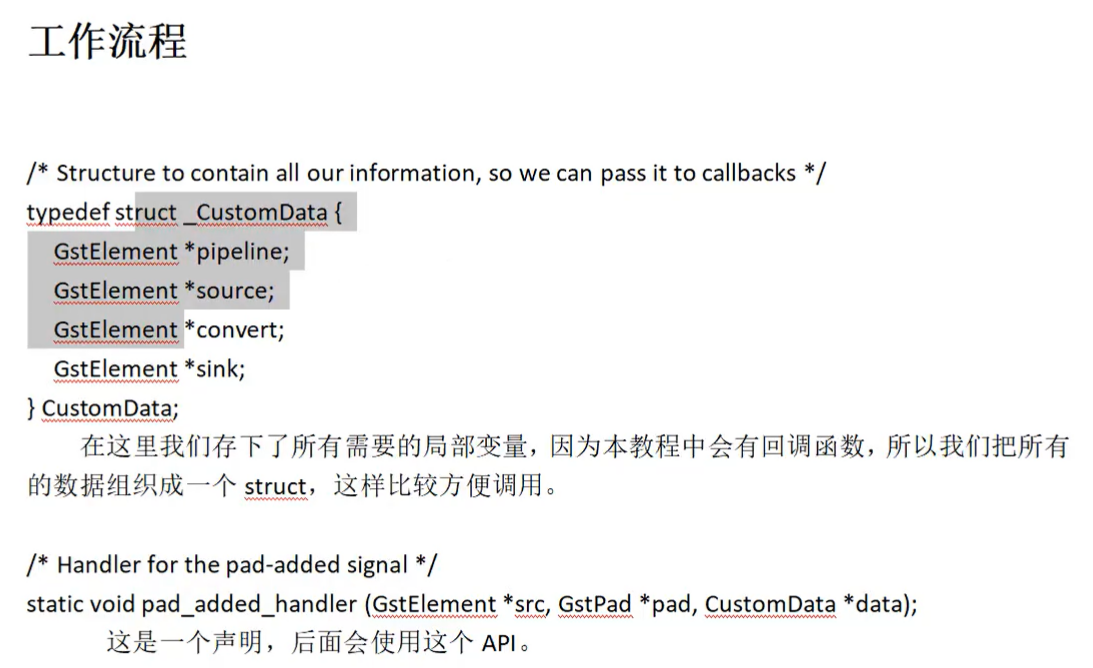


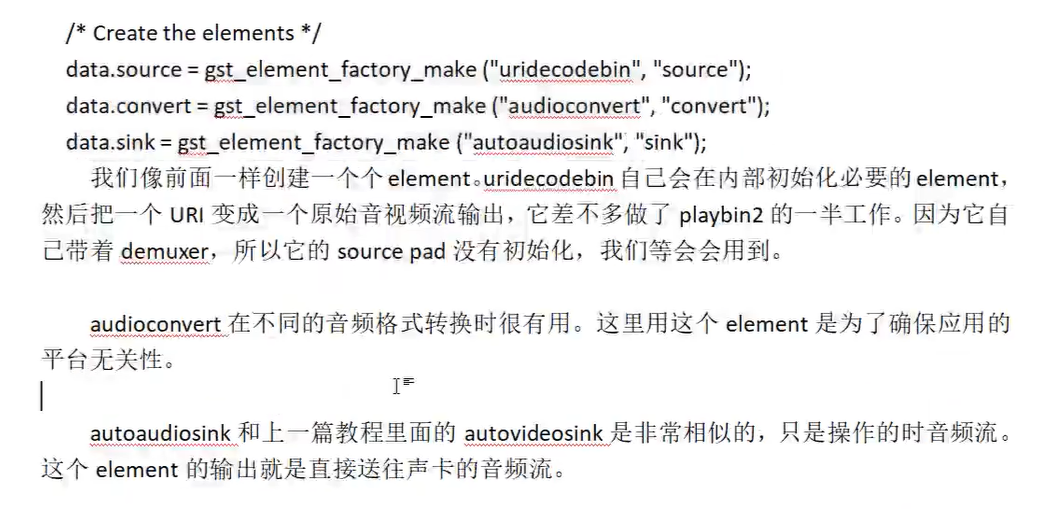


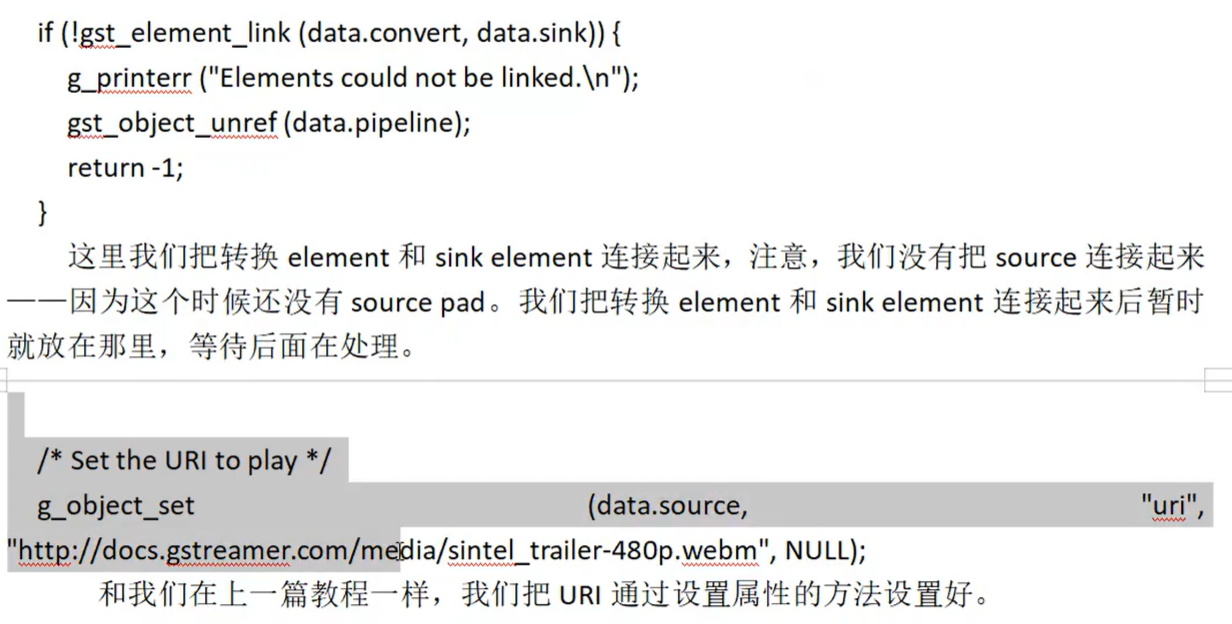




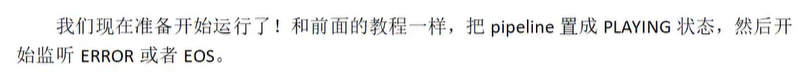


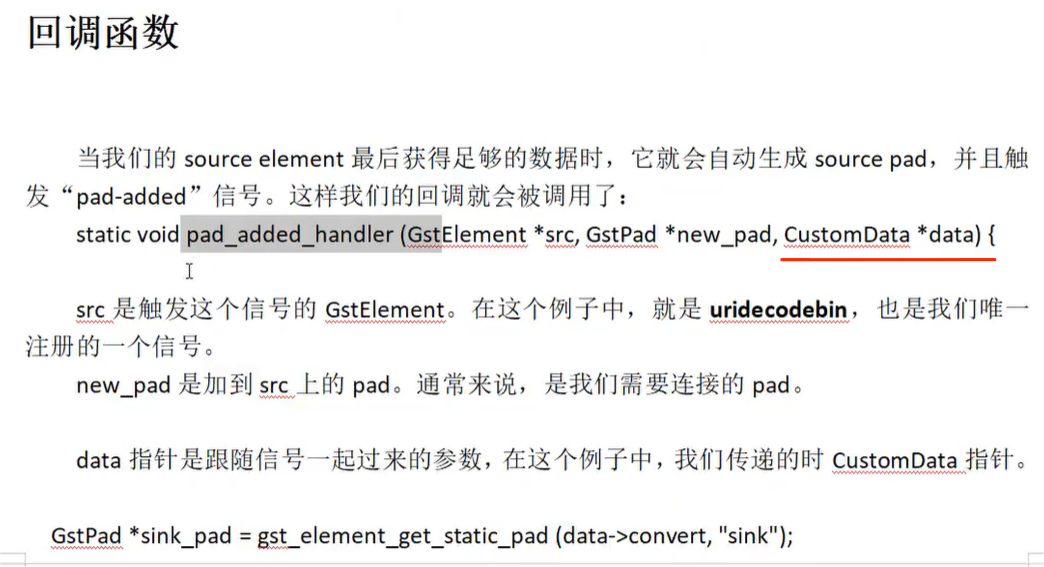




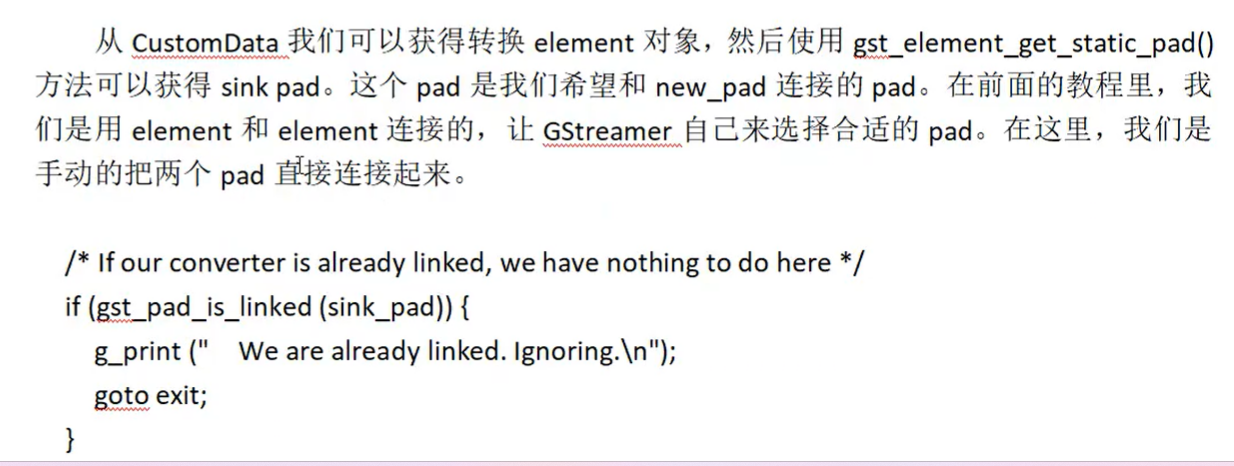








这里的CustomData就是我们自定义的结构体，里面可以取出我们想要的对象。



因为这个回调函数会被调用多次，而每次调用的时候,触发Gstelement的src是不同的。在程序里，我们注册的信号是uridecodebin。

而new\_pad是要加到src上的pad。

data指针是跟随信号一起过来的参数。

