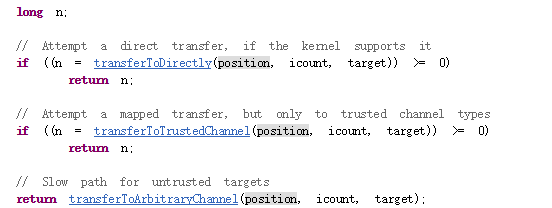
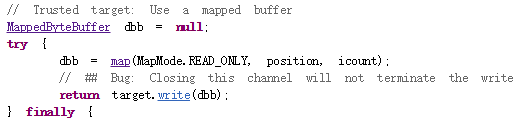
根据之前谭杰学长的分析结果，分析FileChannel类的transferTo方法，它的具体实现在FileChannelImpl类中。FileChannelImpl类中transferTo方法的关键性代码是：



在我的测试中，使用的测试环境不同，执行的程序不同。在windows平台上，执行transferToTrustedChannel方法，这与谭杰学长的分析结果一致；而在ubuntu平台上，执行的是transferToDirectly方法。二者的区别是前者使用内存映射的方式进行文件读取，具体代码如下：



这里的map函数从文件读数据，write函数将数据写入target channel。在map方法中，截取关键代码如下：



这里调用了map0方法。对于transferToTrustedChannel方法的分析暂时进行到这里，接下来看transferToDirectly方法。

transferToDirectly方法中有这样一部分代码：



在这里调用了transferTo0方法，这一方法的作用应该就是我们之前讨论过的zero-copy的具体实现。

分析到这里，遇到了一个问题就是：map0方法和transferTo0方法的声明是这样的：





这两个方法均是native方法，也就是说这两个方法的实现均不是有java语言实现，而是由C/C++实现，被编译成DLL以后由java程序调用的。

下一步是不是要找到transferTo0方法的源码？

如果源码是Linux内核代码，是否继续分析？

在linux内核中，能否实现并行？并行度最大为多少？