| 操作     | 传统方式    | 零拷贝方式                    |
|--------|---------|--------------------------|
| 数据拷贝次数 | 4次      | 2次 (mmap) 或1次 (sendfile) |
| 上下文切换  | 4次      | 2次                       |
| CPU占用  | 高(频繁复制) | 极低(DMA主导)                |

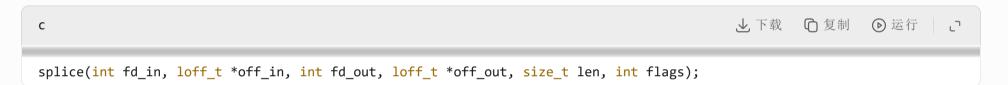
## 四、零拷贝的局限性

- 1. 内存占用:
- mmap会占用虚拟内存空间(文件大小),大文件可能导致进程地址空间紧张。
- 2. 数据修改复杂度:
- 修改mmap映射的文件需处理脏页回写(RocketMQ的CommitLog只追加不修改,规避此问题)。
- 3. 硬件依赖:
  - sendfile需网卡支持Scatter-Gather DMA(现代网卡普遍支持)。

## 五、扩展:现代优化技术

1. splice()

Linux 2.6+ 支持管道+内存页转移,实现两个文件描述符间零拷贝(无需磁盘参与)。



- 2. 硬件加速
- RDMA(远程直接内存访问): 网卡直接读写对方内存,彻底绕过CPU(云原生场景应用广泛)。

## 六、总结

- 核心价值:零拷贝通过减少CPU拷贝次数和规避用户态/内核态切换,将I/O性能推向极限。
- 适用场景: 消息中间件(RocketMQ/Kafka)、Web服务器(Nginx)、大数据传输(HDFS)。
- 实现选择:
- 。 需修改数据 → mmap
- o 纯传输 → sendfile/splice
- 。 分布式系统 → RDMA
- 6 C P B E v D

