

(/apps
/redirect?url
banner-clic

浅析Linux中的零拷贝技术



卡巴拉的树 (/u/e1ae71f0499c) [+ 关注](#)

2017.03.17 22:20* 字数 2645 阅读 6917 评论 3 喜欢 27

(/u/e1ae71f0499c)

本文探讨Linux中**主要的几种零拷贝技术**以及零拷贝技术**适用的场景**。为了迅速建立起零拷贝的概念，我们拿一个常用的场景进行引入：

引文##

在写一个服务端程序时（Web Server或者文件服务器），文件下载是一个基本功能。这时候服务端的任务是：**将服务端主机磁盘中的文件不做修改地从已连接的socket发出去**，我们通常用下面的代码完成：

```
while((n = read(diskfd, buf, BUF_SIZE)) > 0)
    write(socketfd, buf, n);
```

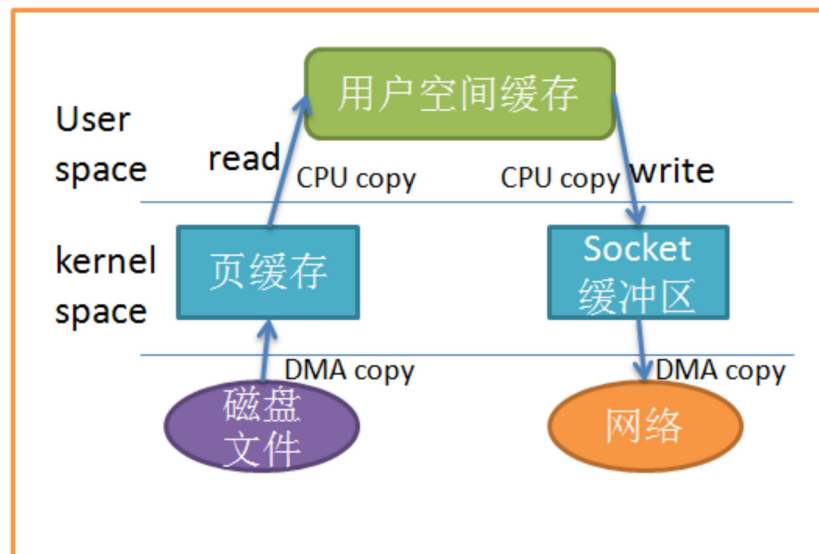
基本操作就是循环的从磁盘读入文件内容到缓冲区，再将缓冲区的内容发送到 socket。但是由于Linux的 I/O 操作默认是缓冲 I/O。这里面主要使用的也就是 read 和 write 两个系统调用，我们并不知道操作系统在其中做了什么。实际上在以上 I/O 操作中，发生了多次的数据拷贝。

当应用程序访问某块数据时，操作系统首先会检查，是不是最近访问过此文件，文件内容是否缓存在内核缓冲区，如果是，操作系统则直接根据 read 系统调用提供的 buf 地址，将内核缓冲区的内容拷贝到 buf 所指定的用户空间缓冲区中去。如果不是，操作系统则首先将磁盘上的数据拷贝的内核缓冲区，这一步目前主要依靠 DMA 来传输，然后再把内核缓冲区上的内容拷贝到用户缓冲区中。

接下来，write 系统调用再把用户缓冲区的内容拷贝到网络堆栈相关的内核缓冲区中，最后 socket 再把内核缓冲区的内容发送到网卡上。

说了这么多，不如看图清楚：





数据拷贝

(/apps
/redirect?ui
banner-clic

从上图中可以看出，共产生了四次数据拷贝，即使使用了 DMA 来处理了与硬件的通讯，CPU 仍然需要处理两次数据拷贝，与此同时，在用户态与内核态也发生了多次上下文切换，无疑也加重了 CPU 负担。

在此过程中，我们没有对文件内容做任何修改，那么在内核空间和用户空间来回拷贝数据无疑就是一种浪费，而零拷贝主要就是为了解决这种低效性。

什么是零拷贝技术（zero-copy）？##

零拷贝主要的任务就是**避免** CPU 将数据从一块存储拷贝到另外一块存储，主要就是利用各种零拷贝技术，避免让 CPU 做大量的数据拷贝任务，减少不必要的拷贝，或者让别的组件来做这一类简单的数据传输任务，让 CPU 解脱出来专注于别的任务。这样就可以让系统资源的利用更加有效。

我们继续回到引文中的例子，我们如何减少数据拷贝的次数呢？一个很明显的着力点就是减少数据在内核空间和用户空间来回拷贝，这也引入了零拷贝的一个类型：

让数据传输不需要经过 user space

使用 mmap

我们减少拷贝次数的一种方法是调用 `mmap()` 来代替 `read` 调用：

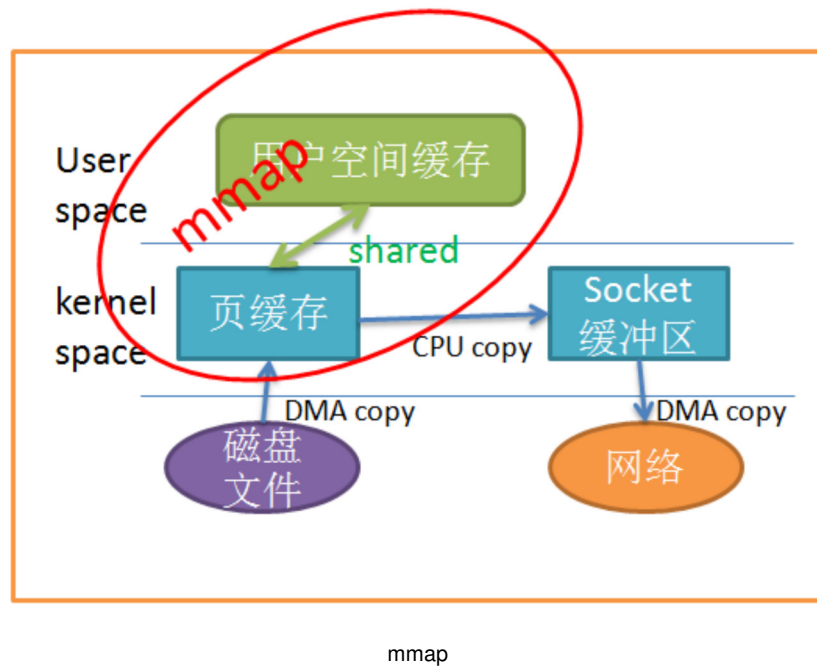
```
buf = mmap(diskfd, len);  
write(sockfd, buf, len);
```

应用程序调用 `mmap()`，磁盘上的数据会通过 DMA 被拷贝的内核缓冲区，接着操作系统会把这段内核缓冲区与应用程序共享，这样就不需要把内核缓冲区的内容往用户空间拷贝。应用程序再调用 `write()`，操作系统直接将内核缓冲区的内容拷贝到 socket 缓冲区中，这一



一切都发生在内核态，最后，socket 缓冲区再把数据发到网卡去。

同样的，看图很简单：



(/apps
/redirect?ui
banner-clip

使用mmap替代read很明显减少了一次拷贝，当拷贝数据量很大时，无疑提升了效率。但是使用 mmap 是有代价的。当你使用 mmap 时，你可能会遇到一些隐藏的陷阱。例如，当你的程序 map 了一个文件，但是当这个文件被另一个进程截断(truncate)时，write系统调用会因为访问非法地址而被 SIGBUS 信号终止。SIGBUS 信号默认会杀死你的进程并产生一个 coredump，如果你的服务器这样被中止了，那会产生一笔损失。

通常我们使用以下解决方案避免这种问题：

1. 为SIGBUS信号建立信号处理程序

当遇到 SIGBUS 信号时，信号处理程序简单地返回，write 系统调用在被中断之前会返回已经写入的字节数，并且 errno 会被设置成success,但是这是一种糟糕的处理办法，因为你并没有解决问题的实质核心。

2. 使用文件租借锁

通常我们使用这种方法，在文件描述符上使用租借锁，我们为文件向内核申请一个租借锁，当其它进程想要截断这个文件时，内核会向我们发送一个实时的 RT_SIGNAL_LEASE 信号，告诉我们内核正在破坏你加持在文件上的读写锁。这样在程序访问非法内存并且被 SIGBUS 杀死之前，你的 write 系统调用会被中断。write 会返回已经写入的字节数，并且置 errno 为success。

我们应该在 mmap 文件之前加锁，并且在操作完文件后解锁：



```
if(fcntl(diskfd, F_SETSIG, RT_SIGNAL_LEASE) == -1) {
    perror("kernel lease set signal");
    return -1;
}
/* L_type can be F_RDLCK F_WRLCK 加锁*/
/* L_type can be F_UNLCK 解锁*/
if(fcntl(diskfd, F_SETLEASE, l_type)){
    perror("kernel lease set type");
    return -1;
}
```

(/apps
/redirect?ui
banner-clic

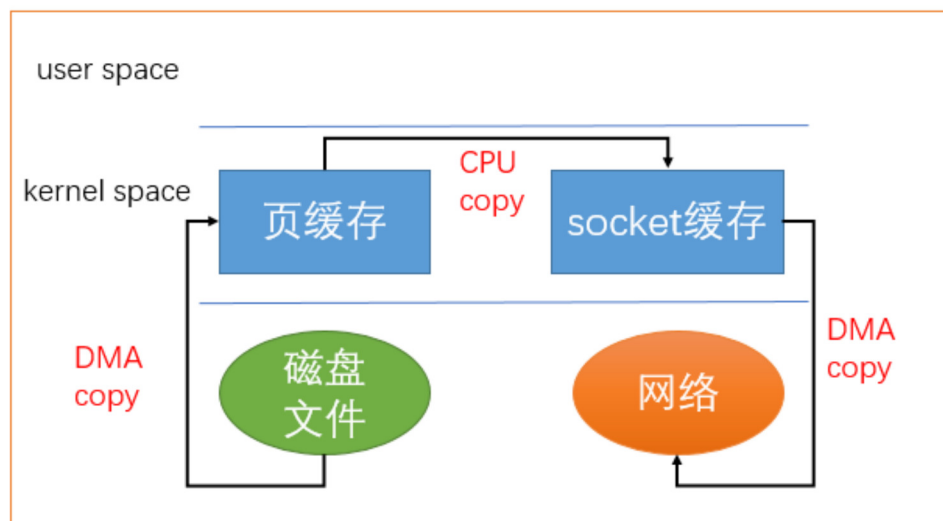
使用sendfile#####

从2.1版内核开始，Linux引入了 sendfile 来简化操作:

```
#include<sys/sendfile.h>
ssize_t sendfile(int out_fd, int in_fd, off_t *offset, size_t count);
```

系统调用 sendfile() 在代表输入文件的描述符 in_fd 和代表输出文件的描述符 out_fd 之间传送文件内容（字节）。描述符 out_fd 必须指向一个套接字，而 in_fd 指向的文件必须是可以 mmap 的。这些局限限制了 sendfile 的使用，使 sendfile 只能将数据从文件传递到套接字上，反之则不行。

使用 sendfile 不仅减少了数据拷贝的次数，还减少了上下文切换，数据传送始终只发生在 kernel space 。



sendfile系统调用过程

在我们调用 sendfile 时，如果有其它进程截断了文件会发生什么呢？假设我们没有设置任何信号处理程序，sendfile 调用仅仅返回它在被中断之前已经传输的字节数，errno 会被置为success。如果我们在调用sendfile之前给文件加了锁，sendfile 的行为仍然和之前相同，我们还会收到RT_SIGNAL_LEASE的信号。



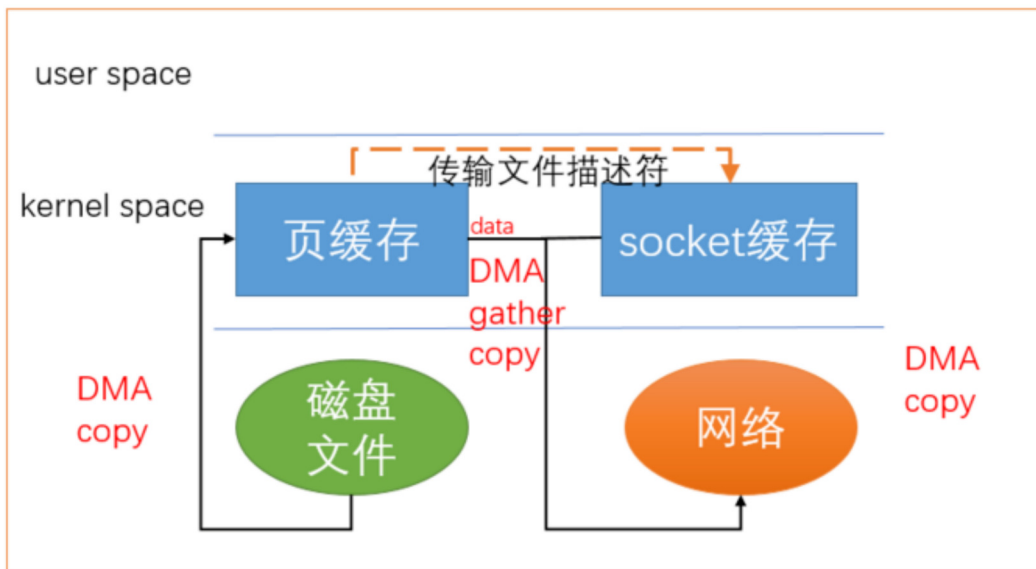
目前为止，我们已经减少了数据拷贝的次数了，但是仍然存在一次拷贝，就是页缓存到s

socket缓存的拷贝。那么能不能把这个拷贝也省略呢？

借助于硬件上的帮助，我们是可以办到的。之前我们是把页缓存的数据拷贝到socket缓存中，实际上，我们仅仅需要把缓冲区描述符传到 socket 缓冲区，再把数据长度传过去，这样 DMA 控制器直接将页缓存中的数据打包发送到网络中就可以了。

(/apps
/redirect?url
banner-clip

总结一下，sendfile 系统调用利用 DMA 引擎将文件内容拷贝到内核缓冲区去，然后将带有文件位置和长度信息的缓冲区描述符添加socket缓冲区去，这一步不会将内核中的数据拷贝到socket缓冲区中，DMA 引擎会将内核缓冲区的数据拷贝到协议引擎中去，避免了最后一次拷贝。



带DMA的sendfile

不过这一种收集拷贝功能是需要硬件以及驱动程序支持的。

使用splice#####

sendfile只适用于将数据从文件拷贝到套接字上，限定了它的使用范围。Linux在 2.6.17 版本引入 splice 系统调用，用于在两个文件描述符中移动数据：

```
#define _GNU_SOURCE          /* See feature_test_macros(7) */
#include <fcntl.h>
ssize_t splice(int fd_in, loff_t *off_in, int fd_out, loff_t *off_out, size_t len, unsigned
```

splice调用在两个文件描述符之间移动数据，而不需要数据在内核空间和用户空间来回拷贝。他从 fd_in 拷贝 len 长度的数据到 fd_out，但是有一方必须是管道设备，这也是目前 splice 的一些局限性。flags 参数有以下几种取值：

- **SPLICE_F_MOVE**：尝试去移动数据而不是拷贝数据。这仅仅是对内核的一个小提示：如果内核不能从 pipe 移动数据或者 pipe 的缓存不是一个整页面，仍然需要拷贝数



- 据。Linux最初的实现有些问题，所以从 2.6.21 开始这个选项不起作用，后面的Linux版本应该会实现。
- **** SPLICE_F_NONBLOCK**** : splice 操作不会被阻塞。然而，如果文件描述符没有被设置为不可被阻塞方式的 I/O ，那么调用 splice 有可能仍然被阻塞。
 - **** SPLICE_F_MORE**** : 后面的 splice 调用会有更多的数据。

(/apps
/redirect?u
banner-clip

splice调用利用了Linux提出的管道缓冲区机制， 所以至少一个描述符要为管道。

以上几种零拷贝技术都是减少数据在用户空间和内核空间拷贝技术实现的，但是有些时候，数据必须在用户空间和内核空间之间拷贝。这时候，我们只能针对数据在用户空间和内核空间拷贝的时机上下功夫了。Linux通常利用**写时复制(copy on write)**来减少系统开销，这个技术又时常称作 **cow** 。

由于篇幅原因，本文不详细介绍写时复制。大概描述下就是：如果多个程序同时访问同一块数据，那么每个程序都拥有指向这块数据的指针，在每个程序看来，自己都是独立拥有这块数据的，只有当程序需要对数据内容进行修改时，才会把数据内容拷贝到程序自己的应用空间里去，这时候，数据才成为该程序的私有数据。如果程序不需要对数据进行修改，那么永远都不需要拷贝数据到自己的应用空间里。这样就减少了数据的拷贝。写时复制的内容可以再写一篇文章了。。。

除此之外，还有一些零拷贝技术，比如传统的Linux I/O中加上 **O_DIRECT** 标记可以直接 I/O ，避免了自动缓存，还有尚未成熟的 **fbufs** 技术，本文尚未覆盖所有零拷贝技术，只是介绍常见的一些，如有兴趣，可以自行研究，一般成熟的服务端项目也会自己改造内核中有关I/O的部分，提高自己的数据传输速率。

< 上一篇 (/p/65710069d934)	目录	下一篇 > (/p/9c3784d8d8ad)
--	--------------------	--

小礼物走一走，来简书关注我

赞赏支持

 计算机基础 (/nb/5038419)

举报文章 © 著作权归作者所有



计算机基础 (/nb/5038419)

1.8万字 · 2.2万阅读 · 70人关注

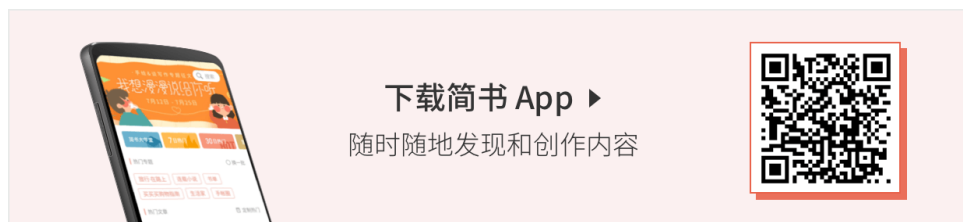
+ 关注



喜欢 | 27



(/apps
/redirect?ui
banner-clic



(/apps/redirect?utm_source=note-bottom-click)



登录 (/sign-in?utm_source=desktop&utm_medium=not-signed-in-comr

3条评论

只看作者

按时间倒序 按时间正序



羽尧 (/u/2b3f808ddf0d)

4楼 · 2018.06.20 10:48

(/u/2b3f808ddf0d)

您好，我想问一下，零拷贝对于大文件和小文件有区别吗？拷贝大文件，比如图片，是否使用零拷贝

赞 回复



张鸣箏 (/u/d209d1ebac64)

3楼 · 2017.11.28 20:57

(/u/d209d1ebac64)

最后还是有一次拷贝呢，为什么不叫one-copy技术

赞 回复



AlferWei (/u/1bf39a87def6)

2楼 · 2017.03.17 22:36

(/u/1bf39a87def6)

好多上层应用都利用零拷贝技术做了些优化，赞

赞 回复

被以下专题收入，发现更多相似内容





程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)



Netty (/c/3dca7e1213f0?utm_source=desktop&utm_medium=notes-included-collection)



污力_Java (/c/b661c20f74f6?utm_source=desktop&utm_medium=notes-included-collection)

(/apps
/redirect?ui
banner-clic

推荐阅读

更多精彩内容 > (/)

七月之痛 (/p/cfe983cd3f2f?utm_campaign=maleski...

那年 用饼干的香 用糖果的甜 用新裙子的美 用小手枪的威风 千方百计哄你卷起衣袖闭上眼眸 亮出接受针尖入肉的勇敢 他们说 从此你身边有道无形的墙 佑你平安健康 今天 他们却说那一针只是逗你玩 筑好的墙其实形同虚设 荨麻疹巫婆 小儿麻痹巫师 还有那红眼珠直尾巴的狗妖 都随时可能破墙而入向你亮出魔爪阴森 可怜的你 早已被变色的良心列为不设防 他们说 从五十万到二个亿耗时两年太慢 他们说 百分之九十的利润太薄 谁来感知 孩子的一生进入永夜是短还是长 盖在冰凉的小小躯体上的白色床单 到底是厚还是薄 七月之痛 是天理给出的拷问 拷问后自有七月烈焰抛下的绞刑 青天白日 五雷轰顶 结束！名号长生的嗜血之王

(/p/cfe983cd3f2f?utm_campaign=malesk
utm_content=note&
utm_medium=pc_all_hots&
utm_source=recommendation)

南飞雨燕 (/u/fab322d55108?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

刺心:就算是药神，也救不了这么多的苦难。 (/p/a19a8...

作者：任争气 活着，生而为人最基本的生存状态。可是，在有些时候却显得无能为力，就像握紧的拳头打进了棉花包里，任凭你再努力也没用。疾病不会因为你贫穷而放过你，不会应为你无力支出而可怜你。《我不是药神》这部电影火了、真的火了。那种面对疾病的无助和束手无策很扯心，那种静静等待死亡的心情谁能懂。可叹。可悲、可怜、可恨，在癌症面前，生面又如草芥、生死不能掌握在自己手里。看了、哭了、心疼了、无力了。未来的不确定性正是生活可期待的精彩之处，唯独灾难、疾病让我们望而却步，那种几万分之一、几十万分之一、几百万分之一的可能性一旦降临就是百分之百，它从不管你有没有做好准备、也不管你是贫穷还是富有，因为灾难...

(/p/a19a87b671f9?utm_campaign=male
utm_content=note&
utm_medium=pc_all_hots&
utm_source=recommendation)

醉美长安 (/u/7886cd7da46e?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

献给最可爱的人 (/p/9639b327e21d?utm_campaign=maleskine&utm_co...

文署雨林季风 用万年花草 编织一副手腕 赠送于你 右手举起忠诚 左手时刻准备着 凭借日月香炉 煮一壶清茶 赐予给你 五湖替你举杯 与狂风对饮 凭借高山流水 弹奏一曲天籁之音 感恩于你 和平之声 因你恪守了一份安宁 大海因你而感慨 惊涛骇浪 你从未改变前进的方向 冰天雪地 你却巡热了这片国土 四季为你记载 和平鸽为你放飞 你们是人民最可爱的人 威武之师 文明之师

雨林季风 (/u/1417ef767446?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)



荷叶上的那滴珠 (/p/57d39018746c?utm_campaign=...

睡眼惺忪的启明星 趟在静谧的水面没有纹波 黑夜流出一声哈欠 凝是倦蝉发出的哑音 摇撻的桨声 摇出一尾婀娜的情影 哗哗的水响 青蛙说这不是我的抖音 是谁在莲叶上 留下那初吻的液珠 竟然跳着蝌蚪的舞姿 风说是星星匆忙卸妆的眼粉 雾说是晨曦馈赠的礼品 谁家早起的铃声 配图却是朦胧的背景 缩起的蝴蝶结 在荷叶间停停飞飞 采藕的姑娘 把自己采成了蝴蝶的翅膀 额头零乱的发丝 就像藕丝一样情长 东方的鱼肚白是你的信套 里面藏着银铃的笑声 荷刺不小心蜇破了封皮 撒落一池标点符号 恰似荷叶上 青蛙的瞳仁

挑夫 (/u/ab7f77abd0bc?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/57d39018746c?utm_campaign=male
utm_content=note&
utm_medium=pc_all_hots&
utm_source=recommendation))
/apps
/redirect?ui
banner-clic

诗||周末（火星文艺主题作业） (/p/c948016f42c5?utm...

我把自己摆成一尾鱼 游荡在多情的西湖 游人穿梭拍照 我却觉得太吵 于是 我把自己化成一缕风 穿梭于声名鹊起的 晓书院 假装成文人畅享书海 后来 我又把自己装扮成快乐精灵 去欣赏宋城千古情 沉醉歌舞升平 梦回西子 华灯初上 微风轻拂 西湖风光 绮丽尽展 乘船出行 醉在荷花 一曲琵琶 梦里来把 不知归路 遇见人佳

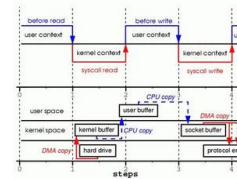
相由心生MM (/u/ce00e90002e6?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)


(/p/c948016f42c5?utm_campaign=males
utm_content=note&
utm_medium=pc_all_hots&
utm_source=recommendation)

(/p/e76e3580e356?utm_campaign=maleskine&
utm_content=note&utm_medium=seo_notes&
utm_source=recommendation)

浅谈 Linux下的零拷贝机制 (/p/e76e3580e356?utm_c...

什么是零拷贝 维基上是这么描述零拷贝的：零拷贝描述的是CPU不执行拷贝数据从一个存储区域到另一个存储区域的任务，这通常用于通过网络传输一个文件时以减少CPU周期和内存带宽。零拷贝给我们带来的好处：减少甚至完全避免不必要的CPU拷贝，从而让CPU解脱出来去执行其他的任务 ...

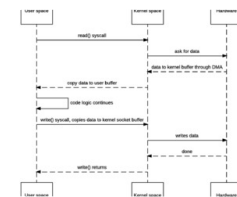



 tomas家的小拨浪鼓 (/u/c4a967f15149?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/f3bea2f6c0b7?utm_campaign=maleskine&
utm_content=note&utm_medium=seo_notes&
utm_source=recommendation)

关于Linux的零拷贝技术详解 (/p/f3bea2f6c0b7?utm...

零拷贝机制原理分析之前，我们先来看下传统IO在数据拷贝的基本原理，从数据拷贝(I/O拷贝)的次数以及上下文切换的次数进行对比分析。传统IO: 传统IO方式，一共在用户态空间与内核态空间之间发生了4次上下文的切换，4次数据的拷贝过程，其中包括2次DMA拷贝和2次I/O拷贝（ ...



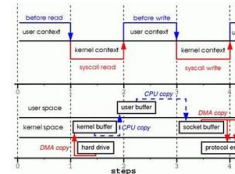
 东升的思考 (/u/62b7335d44e6?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)




(/p/e9f422586749?utm_campaign=maleskine&utm_content=note&

utm_medium=seo_notes&utm_source=recommendation)
Linux 零拷贝技术 (/p/e9f422586749?utm_campaign=...

目录 [TOC] 简介 零拷贝（zero-copy）技术可以减少数据拷贝和共享总线操作的次数，消除通信数据在存储器之间不必要的中间拷贝过程，有效地提高通信效率，是设计高速接口通道、实现高速服务器和路由器的关键技术之一。数据拷贝受制于传统的操作系统或通信协议，限制了通信性能。...




(/apps
/redirect?ui
banner-clic

 炫酷生活 (/u/682a766e000b?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


计算机操作系统（汤小丹）第四版 (/p/be2d3e35c5d7?utm_campaign=ma...

第一章1. 设计现代OS的主要目标是什么？答：（1）有效性（2）方便性（3）可扩展性（4）开放性
2. OS的作用可表现在哪几个方面？答：（1）OS作为用户与计算机硬件系统之间的接口（2）OS作为计算机系统资源的管理者（3）OS实现了对计算机资源的抽象3. 为什么说OS实现了...

 吴业鹏 (/u/9b24e7db193a?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


读书笔记 - 《程序员的自我修养》 (/p/78a1e8d85e5f?utm_campaign=mal...

一、温故而知新 1. 内存不够怎么办 内存简单分配策略的问题地址空间不隔离内存使用效率低程序运行的地址不确定 关于隔离：分为 虚拟地址空间和 物理地址空间 分段：把一段程序所需要的内存空间大小映射到某个地址空间 分页：把地址空间人为地等分成固定大小的页，每一页...

 SeanCST (/u/7283ea6ff5b6?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

先活着，再生活 (/p/b55cbc7da659?utm_campaign=maleskine&utm_co...

刚工作那会，从开始找实习工作，没有会计证都不敢投会计相关的工作，只好找文员类的。没有工作经验的我们投的简历大多石沉大海没有动静，晚上大概都是想东想西翻来覆去睡不着的时光，那时大多憧憬着未来生活的样子。2014年十一月是我们学校准备实习的时候，以前周末喜欢逛武汉江...


 luck愿愿 (/u/ff09dd98fa5b?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/7a064764dae8?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

人因互相“麻烦”而亲近 (/p/7a064764dae8?utm_camp...

俗话说水至清则无鱼，水里太清澈了鱼则无藏身之处，人又何尝不是，人至清则无友。以前接受的教育都是自己的事情自己干，尽量不要去麻烦别人，时间久了，才发现，事情自己都做了，但周边的朋友却越来越少，在亲戚朋友眼中，你就是无需帮助的。最近发生的两件事情让我深有感触。由于天气越来越暖...




 方雅性格完善 (/u/0f511768f363?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)



(/p/4d353a432648?utm_campaign=maleskine&utm_content=note&

utm_medium=seo_notes&utm_source=recommendation)
微雨微阴 (/p/4d353a432648?utm_campaign=malesk...

 愿和你一起飞


(/u/6e8948cf93d9?utm_campaign=maleskine&
utm_content=user&utm_medium=seo_notes&utm_source=recommendation)



(/apps
/redirect?ui
banner-clic

(/p/a914fd2fb4be?utm_campaign=maleskine&
utm_content=note&utm_medium=seo_notes&
utm_source=recommendation)
池边散步 (/p/a914fd2fb4be?utm_campaign=maleski...

凝望荷花淀 潺潺一水深 清风执新蕾 微雨濯青莲

 塞北清寒 (/u/a02c058bfd82?utm_campaign=maleskine&utm_content=user&
utm_medium=seo_notes&utm_source=recommendation)

