

最近，一位之前一直在寻找它的用户在边肖向我们提出了一个问题。相信这也是很多币圈朋友经常疑惑的问题：IONC币值不值得重仓？knc币怎么样？带着这个问题，让专业的边肖告诉你原因。

Collection-Collection-Collectivity

Connection-Connection

Structure-Structure

contradiction-contradiction

correctness-correction

confession-confession

compression-compression

calculation-calculation

celebration-celebration

Cycle-cyclecooperation-cooperationcommemoration-commemoration

communication-communication-communication

compensation-compensation

completion-completion

competition-competition

concentration-concentration

confused-confused

congratulations-congratulations

consolidate-consolidate

Composition-Composition-Composition

Contribution-Contribution

Meditation-Meditation-MeditationCooperation-Cooperation

coordinates-coordination

association-correlation

creation-creation-creativity

culture-culture

cancel-cancel

concentration-concentration

Capitalization,capitalizationcivilization,civilizedcertification

clarification-clarification

classification-classification

combination-combination

compute-calculate

confrontation-confrontation

conservation-conservation-conservation

Consider-consider-beconsiderate-consult-consult-talkComputerizedcomput  
erizedcondemnationcondemnationconfirmationconfirmation

Cohesion-Cohesion-Cohesion

Collision-Collision

Corrosion-Corrosion

Understanding-Understanding-Synthesis

Summary-Conclusion-conclusive

submission-commission

composition-composition

consumption-consumption

comparison-comparison

transformation-transformation

上一篇文章在解析和初始化GraphicBuffer时，会遇到一个ion驱动程序来管理原语。首先，让's来看看ion是怎么用的：

我们按照这个流程来分析ion的源代码。如果你对离子的使用感兴趣，可以去本文底部看

本文基于AndroidLinux内核3.1.8版

。欢迎来本文讨论

离子是什么？如果's的音频和视频相机工程师会熟悉这个驱动程序。最早的GPU和其他驱动合作申请一块内存来画，用的是粗糙的共享内存。安卓系统使用匿名内存。。显示器和摄像头共享内存的问题最早是三星实现的，曾经在Linux社区提出过一段时间。之后在各种大牛的不断改进下，成为了dma\_buf驱动。并在Linux-3.3主线版本中并入主线。。现在它已被广泛应用于各大多媒体开发中。

首先介绍dma\_buf的两个角色，importer和exporter。Importer是dma\_buf驱动程序中的原始使用者。导出器是dma\_buf驱动程序中的原始生产者。

这里's大榭的一张图：

ion是基于dma\_buf设计的。阅读源代码其实很多想法都类似于安卓的匿名内存。

这不#039；如果你不在乎#039；在阅读本文之前，我不知道dma\_buf的设计思想。我赢了#039；不要仔细研究每一行。我会关注它在gralloc服务中的应用过程，看看ion是如何管理共享内存的。为什么要抛弃阿什姆？

Let#039；让我们来看看ion#039sfile\_operation:

只有一个open和ioctl函数。但是没有mmap映射。。因此，当mmap被映射时，其他对象必须正在工作。

我们关注显卡NVIDIA的初始化模块。

文件：/drivers/staging/Android/ion/tegra/tegra\_ion.c

module\_platform\_driver其实就是我之前经常提到的module\_init的一个宏，增加了一个在对应平台注册寄存器的步骤。。这里注册了一个探针方法指针，注册内核模块时调用探针指向的tegra\_ion\_probe。

Let#039；让我们看看相应的结构：

让#039；让我们来看看相应ion中的堆结构：

完成以下步骤：

我们不#039；不要在意调试模式。其实整件事就是我们分析过很多次的方法。在miscdevice中注册此对象。。等到insmod将整个内核模块从dev\_t.

Let#039；让我们来看看这个驱动结构：

file:/drivers/staging/Android/ion/ion\_heap.c

这里有四个不同的堆将被应用。让#039；让我们来看看默认的ION\_HEAP\_TYPE\_SYSTEM对应的堆进程。

其实真正象征ion的内存堆是下面这个结构

，不考虑原来的堆。，会新建三个ion\_system\_heap，顺序分别为8，4，0，大于4的内存较大。这意味着这个堆里有一个ion\_page\_pool页面资源池，只包含顺序对应的2的幂，以及内存块。。其实有点类似合伙人制度。

还会将标志设置为ion\_heap\_flag\_defender\_free，后面会用到。

file:/drivers/staging/Android/ion/ion\_page\_pool.c

isdividedintotwolinkedlistsintheshwimmingpool,oneishigh\_item.另一个是low\_items。此时两者的区分是以2和4的幂为分界线的。

文件：/drivers/staging/Android/ion/ion.c

有一个收缩方法，因为标志位ion\_heap\_flag\_defender\_free和heap是打开的。因此，初始化了两个回收函数。

file:/drivers/staging/Android/ion/ion\_heap.c

此时会创建一个内核线程。，调用ion\_heap\_deferred\_free内核进行连续循环处理。但是，因为这个线程被设置为SCHED\_IDLE，所以这是时间片轮转抢占的最低级别。与处理程序adle相同的处理规则。，即空闲时间处理。

在这个循环中，heap的free\_list中无用的ion\_buffer缓冲区对象被反复破坏。

file:/drivers/staging/Android/ion/ion\_system\_heap.c

注册了heap的内存销毁方法。当系统需要销毁页面时,itwillcallthefunction

registeredthroughtheregistration\_shrinkfile:/drivers/staging/Android/ion/ion\_page\_pool.c.

整个过程很简单，其实就是遍历需要销毁的页数，然后如果是8的幂，就是去掉high\_items中的页面缓存。4和0销毁low\_items中的页面缓存。。至于为什么是2的幂，其实很简单，为了简单的破坏和应用。\_\_free\_pages可能会破坏整个页面。

file:/drivers/staging/Android/ion/ion.c

ismainlyusedtoinitializevariousparametersofion\_client.最后，将ion\_client插入到ion\_device的客户机中。让&#039；让我们来看看ion\_client的结构：

内核仍然调用ion\_alloc来申请ionbuffer的句柄。。最后，将数据复制到用户空间

。

这其实就是寻找可以携带的最小尺寸，申请内存。申请8kb内存的话，会在0-4kb，4kb-16kb，16kb-128kb之间拆分分数。。恰好dma只能应用在128kb以内。超过这个数字，申请被禁止。8kb将被拆分为2个4kb，并存储在第一个池中。

最后，所有应用的页面都被添加到pages集合中。

文件：/drivers/staging/Android/ion/ion\_page\_pool.c

可以看出，此时相应大小区域的空闲页会从ion\_page\_pool中被冲刷出来，返回到上层。如果最早没有，将调用ion\_page\_pool\_alloc\_pages来申请新页面。。由于引用最终来自于ion\_page\_pool，所以在应用之后仍然在ion\_page\_pool中。

这里的处理是为了避免DMA直接内存造成的缓存差异(一般应用默认情况下带有DMA标志)。换句话说，是否打开缓存，其实就是关闭时使用池缓存，打开时使用DMA缓存而不是池缓存。

我们可以看到另一个dma的堆。Howdoesitachievetheconsistencyofdirectmemoryaccesstomemory?

file:drivers/staging/Android/ion/ion\_CMA\_heap.c

可以看出，为了达到dma缓存的一致性，它使用dma\_alloc\_coherent创建一个全部强制同步的地址，即没有dma缓存的地址。这里有几个新的结构。、sg\_table和scatterlist

File:/lib/scatterlist.c

这里实际做的是一件事：初始化sg\_table。

。

sg\_table有一个核心对象，即散射列表链表。。如果pages应用的对象数是PAGE\_SIZE/sizeof(散点列表)，那么每个sg\_table只有一个散点列表。但是在这个数字之外，将添加一个散射列表。

公式方面：

换句话说，生成散点表的链表会尽量直接占用一页，这样内存更容易管理。

返回sg\_table。

初始化ion\_handle，记录对应的ion\_client是当前打开文件的进程，将ion\_buffer设置到句柄中。使句柄与缓冲区相关联。

每当需要销毁ion\_buffer时，

先看看提示的程序是什么。可能是这个程序有问题。如果它被卸载并重新加载。如果出现引导提示，您可以删除该程序的引导。如果可以；如果找不到，你可以回忆起来。出现此故障之前，您下载了哪些程序、软件、插件或驱动程序？如果你先想到卸载它们，如果故障排除了，说明是程序软件或者插件导致的。你重新下载并安装。如果故障依然存在，说明这些程序或者插件有问题。，不能再用了。此外，程序和驱动程序的启动也有错误。可以去创业公司看看。如果有提示，可以去掉(故障不会无缘无故的发生，最重要的是在出现问题之前想好操作。如果是玩游戏时的提示，可能是游戏本身有问题。而且这个游戏和你的系统不兼容，显卡驱动也不合适。如果你能；找不到，建议先杀木马，尝试修复系统。

建议大家下载windows清理助手清理系统，查杀木马。、清理启动错误等。

1. 请使用系统自带的系统进行恢复，直到不；不要有这种故障(或者使用恢复软件来恢复系统。如果可以；进入系统，按F8进入安全模式恢复系统)。

2. 如果故障依旧，用系统盘修复，打开命令提示符输入SFC/SCANNOW(SFC和/)插入原系统盘修复系统，系统会自动对比修复。

3. 如果故障仍然存在。将BIOS中的光驱设置为第一个引导设备，插入原始系统安装盘，按R键选择“修复并安装”。

4. 如果故障仍然存在，建议重新安装操作系统。

如果真的没有；工作，建议选择修复安装，让系统得到修复。，同时也使C程序和文件免受损失。

系统自带的系统还原使用方法：

系统自带的系统还原：开始/程序/附件/系统工具/系统还原。单击“将我的计算机还原到较早的时间”然后单击下一步。你会看到日期页面有一个深色的日期，也就是还原点。选择后，单击下一步恢复(Win7恢复系统，然后是控制面板中的设备和安全选项，然后是备份和恢复选项)。

都看过了吗？相信现在你对IONC币是否值得重仓已经有了初步的了解！也可以收藏页面获取更多关于knc币的知识！区块链，虚拟货币，我们是认真的！