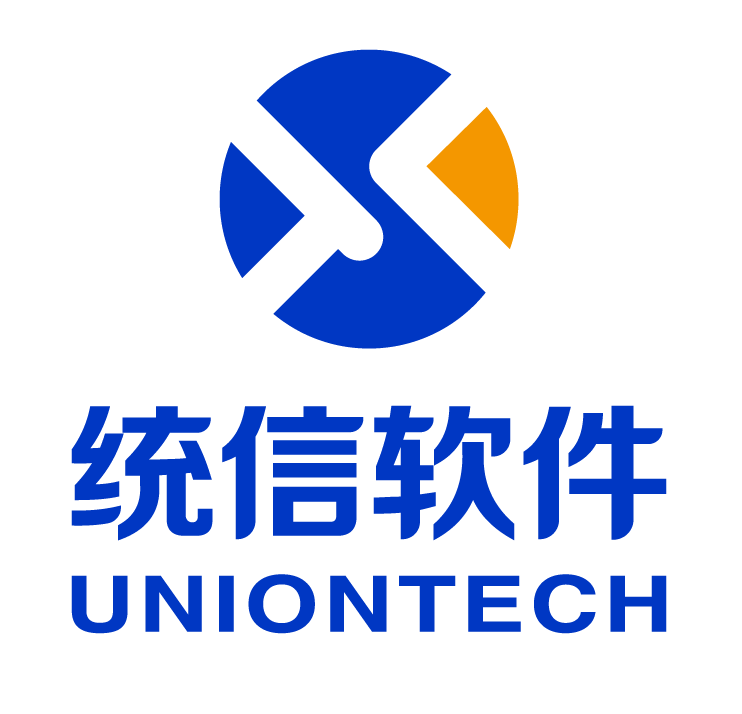
在overlayroot环境下nvidia闭源切开源驱动版本回退失败问题根因分析报告



统信软件技术有限公司

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 |  | 版本 | 1.０ |
| 日期 | 2021年05月27日 | 密级 | 内部 |
| 使用范围 | 研发中心 | | |

版本变更记录

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 版本 | 说明 | 修改人 |
| 2021/04/12 | 1.0.0 | 创建 | 胡晓东 |
| 2021/05/27 | 1.0.1 | 修改 | 胡晓东 |
| 2021/05/31 | 1.0.2 | 修改 | 胡晓东 |
| 2021/06/04 | 1.0.3 | 修改 | 胡晓东 |

目录

[1. 问题概述](#_Toc1008691611_WPSOffice_Level1) [2](#_Toc1008691611_WPSOffice_Level1)

[1.1. 显卡驱动管理器介绍](#_Toc1372973055_WPSOffice_Level2) [2](#_Toc1372973055_WPSOffice_Level2)

[1.2. 问题现象](#_Toc2113419558_WPSOffice_Level2) [3](#_Toc2113419558_WPSOffice_Level2)

[2. 问题分析](#_Toc1372973055_WPSOffice_Level1) [3](#_Toc1372973055_WPSOffice_Level1)

[2.1. overlayroot背景知识介绍](#_Toc2026243684_WPSOffice_Level2) [3](#_Toc2026243684_WPSOffice_Level2)

[2.1.1. overlayfs](#_Toc1372973055_WPSOffice_Level3) [3](#_Toc1372973055_WPSOffice_Level3)

[2.1.2. overlayroot](#_Toc2113419558_WPSOffice_Level3) [4](#_Toc2113419558_WPSOffice_Level3)

[2.2. 问题分析步骤](#_Toc1142744387_WPSOffice_Level2) [5](#_Toc1142744387_WPSOffice_Level2)

[3. 实验验证](#_Toc2113419558_WPSOffice_Level1) [7](#_Toc2113419558_WPSOffice_Level1)

[3.1. 验证步骤](#_Toc1984300566_WPSOffice_Level2) [7](#_Toc1984300566_WPSOffice_Level2)

[3.2. 问题根因](#_Toc1916470173_WPSOffice_Level2) [7](#_Toc1916470173_WPSOffice_Level2)

[4. 解决方案](#_Toc2026243684_WPSOffice_Level1) [8](#_Toc2026243684_WPSOffice_Level1)

[5. 小结](#_Toc1142744387_WPSOffice_Level1) [10](#_Toc1142744387_WPSOffice_Level1)

# 问题概述

## 显卡驱动管理器介绍

显卡驱动管理器切换驱动大致流程如下图所示：

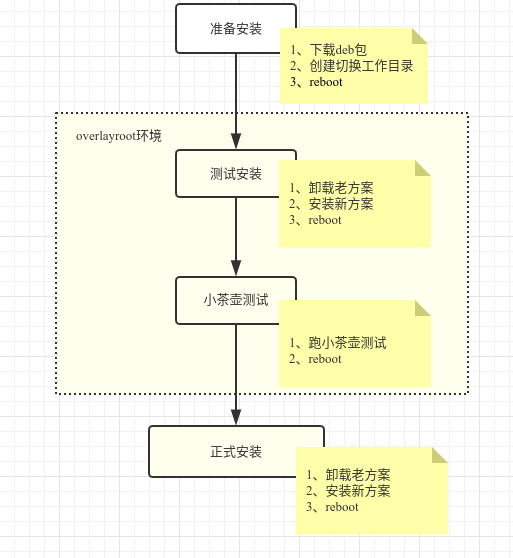


图1 驱动切换流程图

显卡驱动工具中使用到了一个关键的技术，用来保证当显卡驱动切换失败时，系统还能完好无损的回退到切换前的状态，这个关键技术就是overlayroot。

按照最初对overlayroot的理解，它应该是在整个系统上覆盖了一层(可以抽象理解为PS的图层)，然后在系统上做的任何修改，最后都可以撤销，不留下任何痕迹。但是其实不然，在遇到下面问题之后才发现overlayroot并非能够对整个系统起到"保护"作用，具体情况请看后面讲解。

## 问题现象

该问题是由测试那边发现的，他们在测试nvidia闭源驱动向开源驱动切换时，测试安装完毕，并且小茶壶测试也跑成功，在最后需要选择点确认或者取消时，点了取消，然后重启系统，系统起来后就退出了overlayroot环境，这时系统应该回退到闭源驱动，但是结果和期望的情况不太一样，系统加载的却是开源驱动。

# 问题分析

## overlayroot背景知识介绍

overayroot是overlayfs的一种特殊应用，所以要弄懂overlayroot就必须先了解overlayfs，由于篇幅限制，这里都只做简单介绍。

### overlayfs

overlafs挂载：

mount -t overlay overlay -o lowerdir=lower,upperdir=upper,workdir=work merged

lowerdir:是只读层，对用户不可见；

upperdir:是可读写层，对用户不可见；

merged:是合并层，对用户可见。

overlayfs是一种堆叠文件系统，它的作用是把底层目录（包括lower和upper目录）“合并”到一起，它具有如下特性：

（1）如果用户修改某个文件，并且这个文件来自于lower层，那么系统就会自动把该文件拷贝一份到upper层再做修改；

（2）如果用户删除某个文件，并且这个文件来自于lower层，系统会自动在upper层创建一个同名的字符设备文件，overlay文件系统通过这个字符设备文件把lower层的同名文件覆盖，此时用户就看不到这个文件；

（3）如果用户修改或删除的文件来自upper目录，那么系统就会和直接修改或删除这个文件。

从这些特性可以看出，lower是被overlayfs真正保护的内容，upper层保存的是修改差异内容。

### overlayroot

overlayroot挂载：

mount -t overlay -o lowerdir=/media/root-ro, upperdir=/media/root-rw/overlay,workdir=/media/root-rw/overlay-workdir overlayroot /

系统启动之初（init-bottom阶段），会把根分区挂载到/media/root-ro目录，然后再把这个目录做为lower层按照overlay类型的文件系统进行挂载，所以结合上面对overlayfs的理解，可以认为overlayroot其实是对根分区进行保护。

## 问题分析步骤

开始看到上述问题时感觉很懵，整个切换都在overlayfs环境下进行，overlayfs退出时所有的修改应该恢复才对，为什么并没有恢复？接下来针对这个问题来进行了如下分析：

第一步：没有回退回去，说明切换过程中的修改还有“残留”没被撤销，系统不可能和切换前完全一样，这一点是绝对肯定的，否则就不会出现回退不到之前的的状态。

第二步：既然有“残留”修改没有清理干净，那就从修改入手进行分析，切换过程首先要就旧方案卸载，然后在装上新方案，现在的现象是没有回退成功，加载的是新方案驱动，难道是旧方案已经卸载了吗？这里的旧方案是nvidia闭源驱动（见前面现象描述），那就看这个方案的安装包是否还在：dpkg -l |grep nvidia-driver。

第三步： 结果发现nvidia-driver没有卸载，处于安装状态，但是系统显示却使用的是开源方案，难道是驱动模块加载出了问题？那就查看一下内核模块加载情况：查看闭源驱动：lsmod |grep nvidia；查看开源驱动：lsmod |grep nouveau。

第四步：查看后发现加载驱动的是nouveau（开源驱动），开源驱动模块是内核自带的，无需安装其他deb包，闭源驱动是需要从仓库下载安装的，但现在闭源驱动包已经处于安装成功状态，为什么系统加载的不是不闭源驱动，而是开源驱动？难道同时安装开源驱动和闭源驱动后系统优先加载开源驱动，如果是这样，切换前开源驱动和闭源驱动也都同时安装在系统上，为什么就可以加载的闭源驱动？这还是说明肯定还有“残留”修改没有被恢复，否则不会出现这种差异结果。

第五步：查资料后发现，系统内核模块加载有一个黑名单机制，可以通过配置黑名单去阻止特定的内核模块加载。是不是这里也存在一个和这两个驱动加载相关的黑名单？于是就在系统上看了一下，果然有一个/etc/modprobe.d/nvidia-blacklists-nouveau.conf黑名单文件，但是里面内容为:blacklist nouveau，阻止加载的是nouveau，可结果怎么却正好相反呢？

第六步：根据以前的经验，只要在/etc/modprobe.d/目录下添加某个内核模块的黑名单，然后执行update-initramfs -u命令，把黑名单内容同步到initrd中去，重启后一定会生效。现在/etc/modprobe.d/nvidia-blacklists-nouveau.conf这个黑名单没有起作用，难倒是因为没有同步到initrd？如果是这样，只需执行一下update-initramfs -u即可验证。

第七步：按照上面的猜测进行验证，结果和预测完全一致。通过这个测试结果，可以肯定，安装或卸载驱动包过程中对initrd进行了修改，而且最后overlayroot退出时，这个修改没有撤销。对initrd进行了修改很好理解，安装或卸载驱动包是都会执行update-initramfs -u，对initrd进行更新，但是为什么最后这个修改没有撤销呢？

第八步：分析到这前面的猜测已基本证实，在退出overlayroot时，有些“残留”的修改没有被撤销，比如对initrd的修改。为了进一步问题根因，只好再硬着头皮研究一下overlayroot，具体情况见前面背景知识介绍。

第九步：捋清楚overlayroot工作原理后，这个问题就显而易见了，overlayroot保护的是根分区，而initrd存放在boot分区，所以很有可能是这个原因导致了对initrd的修改没有撤销。

# 实验验证

## 验证步骤

为了证明上面的猜想，这里做了个测试，测试步骤：

第一步：执行sudo overlayroot-enable,然后重启系统；

第二步：系统起来后，分别在/ 目录和/boot 目录各创建一个文件；

第三步：执行sudo overlayroot-disable，然后再重启系统；

第四步：系统起来后，分别去/ 目录和/boot 目录下查看创建的文件是否存在，结果发现/boot 目录下的文件还存在，而/ 目录下的文件不见了。

## 问题根因

根据上面分析以及实验验证结果，可以推断出本文提所描述的问题的根因如下：

overlayroot只对根分区起保护作用，对根分区之外的分区（比如boot分区）不起保护作用，但切换驱动过程中会修改boot分区内容。具体的讲，闭源驱动向开源驱动切换过程中，会把/etc/modprobe.d/nvidia-blacklists-nouveau.conf这个删掉，并且这个修改会被同步到initrd中去，但由于initrd存放在boot分区，最后overlayroot退出时这个修改没有被撤销，所以就导致了问题现象中描述的结果，即小茶壶界面点取消后，没有回退到闭源驱动，而是是开源驱动。

# 解决方案

根据上述找到的问题根因，既然boot分区不被overlayroot保护，那么可不可以仿照根分区把boot分区也挂载成overlayfs，后来深入想了想，发现这个方案行不通，因为前面已经讲过overlayroot是在系统启动之初（init-bottom阶段）进行挂载的，所以boot分区也必须在这个阶段挂载，但系统启动在比这更早的阶段就需要挂载好boot分区，因为需要找到内核镜像文件和initrd文件，所以如果把boot分区也挂载成overlayfs，就陷入到鸡生蛋还是蛋生鸡的难题里面去了。

既然boot分区不能走overlayroot这条路，那就只能再想别的办法了，剩下的办法里面最简便易行的就是切换之初把initrd文件拷贝出来，存放到根分区某个目录下，切换过程中如果需要回退，就再把备份的这个initrd文件拷贝回boot分区，覆盖当前的initrd文件，这样就达到了撤销initrd修改的效果了。对于boot分区里面的其他文件比如内核镜像或者grub，切换驱动是不会修改到这里的，所有也不用考虑撤销的问题。

以下是备份和恢复initrd的代码：

|  |
| --- |
| export OVERLAYROOT\_IMAGE=/usr/share/overlayroot.img  export isInOverlayRoot=$(grep -Em1 "^overlayroot / overlay |^overlay-root / overlay " /proc/mounts) || isInOverlayRoot=  ## 备份initrd  backup\_initramfs(){  if [[ -n "${isInOverlayRoot}" ]]; then  cp -f /boot/initrd.img\* ${WORKING\_DIR\_G}  fi  }  ## 恢复initrd  recovery\_initramfs(){  if [[ -n "${isInOverlayRoot}" ]]; then  cp -f ${WORKING\_DIR\_G}/initrd.img\* /boot/  fi  } |

# 小结

根据前面分析，overlayroot用在版本回退这个地方有个短板，那就是overlayroot只能对根分区进行保护，无法对其他分区保护，对其它分区的修改就无法撤销。而且对于别的分区还可以尝试模仿根分区一样挂载成overlayfs，但对boot分区绝对不可能这么做，所以对boot分区的修改如果需要回退的话，就需要单独处理。

前面一直在讨论overlayroot覆盖不到除根分区之外的其它分区，这对需要撤销对这些分区修改时，确实是个麻烦事儿。但在另外一个场景，去显示出了它的优点，比如log保存，就拿显卡驱动管理器为例吧，前端日志保存在 ~/.cache/deepin/deepin-graphics-driver-manager/deepin-graphics-driver-manager.log，后端日志保存在/var/log/deepin/deepin-graphics-driver-manager-server.log, 按照我们系统常规的安装方法，这两个文件那都存放在data 分区，与根分区独立，所以在显卡驱动切换过程中，无论实在overlayroot开启阶段，还是没有开启阶段，这两个地方的日志都是完整被保存的。