关于Linux系统下Grub启动流程的讨论总结 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/252/2021_2022__E5_85_B3_E4_BA_8ELinu_c103_252979.htm 全世界linuxer都知道grub是什么东西，但对于MBR引导到grub再引导到具体操作系统的这个流程可能有不少朋友就比较迷糊了。这不，cu上一位朋友就发出了这样一个求助贴：假如现在一台电脑上装了WIN2000系统，那么我现在在装上LINUX系统和GRUB，那么假如把GRUB装在主分区的话，GRUB直接引导 LINUX和WIN2000，我是可以理解的，因为MBR中是GRUB的STAGE1（对不对呢？），MBR通过检查DPT分区信息引导系统跳转至DBR （活动分区），我这里想问的活动分区是什么时候设的呢？那么装GRUB到MBR里，那原来MBR中的WIN的引导信息是怎么处理的呢？是不是我们假如说装GRUB到MBR的时候，GRUB就把GRUB所在那个区设置为了活动分区了呢？然后GRUB引导时候，MBR就找到那个活动分区找到所需要的文件，然后继续呢？假如说把GRUB装到其他分区（非主引导区）的话，那是怎么样实现GRUB先启动的呢？不是先MBR吗？因为装到了其他分区，没有改主引导区，因此主引导区还是WIN2000的引导数据啊，怎么会GRUB先启动了呢？这是为什么呢？跟活动分区有关系没有呢？我看资料上写的是哪个系统启动哪个系统就是活动分区，可是那样的话，似乎就解释不通了啊，就是最最开始这个地方一直不懂，理不清楚。 下面就是cu各个玩家对这个问题分析讨论的总结。 首先让我们看看传统的启动流程：加载并运行Master Boot Record(MBR)主引导区内容（如lilo等）。

然后扫描分区表，定位活动分区，并将活动分区上的引导扇区内容加载到内存中执行。 系统引导过程主要由以下几个步骤组成（以硬盘启动为例） 1、 开机； 2、 BIOS加电自检（POSTPower On Self Test），内存地址为0fff：0000； 3、 将硬盘第一个扇区（0头0道1扇区，也就是Boot Sector）读入内存地址0000：7c00处； 4、 检查（WORD）0000：7dfe是否等于0xaa55.若不等于则转去尝试其他介质；如果没有其他启动介质，则显示“ No ROM BASIC ”，然后死机； 5、 跳转到0000：7c00处执行MBR中的程序； 6、 MBR先将自己复制到0000：0600处，然后继续执行； 7、 在主分区表中搜索标志为活动的分区。如果发现没有活动分区或者不止一个活动分区，则停止； 8、 将活动分区的第一个扇区读入内存地址0000：7c00处； 9、 检查（WORD）0000：7dfe是否等于0xaa55，若不等于则显示“ Missing Operating System ”，然后停止，或尝试软盘启动； 10、 跳转到0000：7c00处继续执行特定系统的启动程序； 11、 启动系统。 装grub到逻辑分区，那么就一定把grub装入的逻辑分区设为活动的。不过，这时候，grub接管了11步以后的动作：从stage 1.5读出grub.conf。再由配置和用户选择决定下一步的引导行为。 一般安装grub都有两种情况，对于安装到MBR这种情况而言，GRUB直接覆盖了原来的MBR引导程序。这也是为什么要换回“ 原来的 windows的引导方式 ”，只要用dos引导fdisk /mbr一下就可以的原因。为什么可以这样做，请注意，1-11步中有两个地方出现了0000:7c00。不管是dos boot sector还是nt loader它本身也是从0000:7c00运行的。其实ms当年开发分区管理的这个小程序相当于是在bios引导boot sector中插进去的。grub因为

也是写的从0000:7c00这个内存开始的子程序，那么既可以被BIOS加载又可以被dos的MBR加载应该好理解了吧。 开机自检后，引导权交给了硬盘的MBR，此时grub就启动了。由grub来引导windows /linux都可以。注意:linux不一定要安装在活动分区，因为引导程序在MBR！但是windows一定要安装在活动分区（可引导的 windows),第二个windows可以不安装在活动分区，但它的引导文件一定在活动分区。 大体顺序是：grub->windows>查找引导文件引导加载启动windows grub->linux>查找引导文件(/boot)>引导加载启动linux 那么，如果把grub安装到了其它的分区上，不是MBR呢？这是grub所装在的那个主分区必须被设为活动分区。因为MBR(物理主引导分区)中其实并没有 OS相关的引导程序的，通常MBR只是扫描并读取随后的分区表，找到相应的活动分区，读取相应活动分区的第一个扇区的512字节程序并运行，该程序负责进一步引导相应分区的相应系统。因此，大概的运行次序是BIOS>MBR->GRUB->菜单。 这样，大体的真实流程就可以总结如下了： 1、 开机； 2、 BIOS加电自检（POSTPower On Self Test），内存地址为0fff：0000； 3、 将硬盘第一个扇区（0头0道1扇区，也就是Boot Sector）读入内存地址0000：7c00处； 4、 检查（WORD）0000：7dfe是否等于0xaa55.若不等于则转去尝试其他介质；如果没有其他启动介质，则显示“ No ROM BASIC ”，然后死机； 5、 跳转到0000：7c00处执行MBR中的程序； 6、 MBR先将自己复制到0000：0600处，然后继续执行；假如先装XP后装LINUX，并且LINUX没有装在MBR，那这个MBR中的数据还是WIN 写的数据，它的作用都是下步中所说的作用，就是搜索主分区表中标志为活动的

分区，那么这个时候就必须把GRUB所在的主分区设置为活动的分区，这个时候才能正常的启动GRUB，然后GRUB的STAGE1在调STAGE1.5和其他的，从而来引导整个系统。假如说先装XP后装LINUX，但是GRUB装在了MBR，那样STAGE1直接调入内存，STAGE1在调STAGE1.5和STAGE2等，从而来引导系统。那这个时候是不需要将GRUB其他文件所在的主分区设为活动分区的，它直接调STAGE1.5等，然后再调STAGE2等，来识别文件系统，从而实现可多启动。 7、 在主分区表中搜索标志为活动的分区。如果发现没有活动分区或者不止一个活动分区，则停止； 8、 将活动分区的第一个扇区读入内存地址0000：7c00处； 9、 检查（WORD）0000：7dfe是否等于0xaa55，若不等于则显示“ Missing Operating System ”，然后停止，或尝试软盘启动； 10、 跳转到0000：7c00处继续执行特定系统的启动程序； 11、 启动系统。 一点资料： 能正常工作的grub应该包括一下文件：stage1、stage2、*stage1_5、menu.lst。 其中stage1的大小一定是512字节，它要被安装（也就是写入）某个硬盘的主引导记录，或者某个活动分区（这个分区要用fdisk标记成可启动的）的启动扇区。stage1的主要的也是唯一的作用就是找到你存放在硬盘上某个地方的stage2文件，来完成后续的工作。 stage2 文件可以存在在某个特定的文件系统中，比如你分了一个linux分区，在上面创建一个ext2文件系统，然后把这个文件拷贝到这个分区的某个目录下。也可以把stage2直接存放在硬盘的某个位置，也就是未分区的某个地方。不过，好像没有多少人会这么做吧。 因为stage1的容量有限（主引导记录MBR和启动扇区的大小只能够是512字节），所以它对文件系统是无法识别

的，那如果你把stage2存放 在 ext2或者fat格式的文件系统上，它如何来找到这个文件呢？这就要用到上面提到的那些stage1_5的文件了，它们负责解释文件系统。你的 stage2放在什么格式的文件系统上，就要调用对应的那个stage1_5文件。比如，你把stage2存放在ext2格式的文件系统上，就需要e2fs_stage1_5；stage2存放在fat格式的文件系统上，就需要fat_stage1_5了。 