**25. Centos的启动过程**

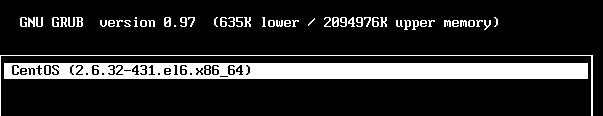
前面好几章都只讲了windows的相关知识，主要还是为了方便大家理解，由简入难。今天我们就开始学习Linux的相关知识。

Linux的发行版本实在是太多了，作者也不可能一一讲解了，因为Linux系统主要还是用在服务器上，而且好像使用Redhat派系的比较多，本章就以Centos6和Centos7为例讲一下其具体启动过程。（大概的流程和windows差不多）

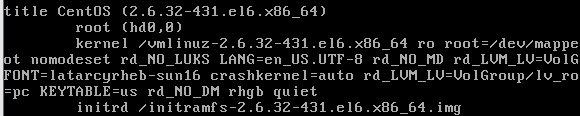
**Centos6系统启动过程**

Bios启动模式下，BIOS预启动程序先读取启动磁盘上的MBR扇区，确定分区表没问题后，再执行mbr扇区里的引导代码，里面的代码是grub引导代码。对，Centos6使用grub引导去引导系统。

grub引导在bios启动模式下，分三个阶段（stage），stage\_1的代码放在MBR扇区的446字节引导区里，主要作用是判断分区表的正确性和加载下一步的引导代码，第二步的引导代码称为stage\_1.5，位于MBR扇区之后的几个扇区里（mbr\_gap），第二步的引导stage\_1.5代码不多，大概不到20KB，主要作用是识别分区里的文件系统，方便加载文件系统里的其他引导文件，比如加载/boot分区里的stage\_2引导程序和grub.conf配置文件，最终提供给用户几个启动菜单，

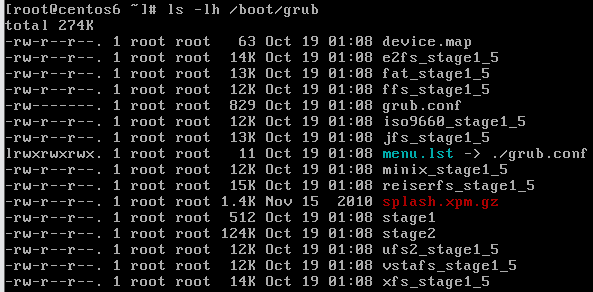


用户选中某菜单后，加载菜单里指定的系统内核和initramfs镜像。



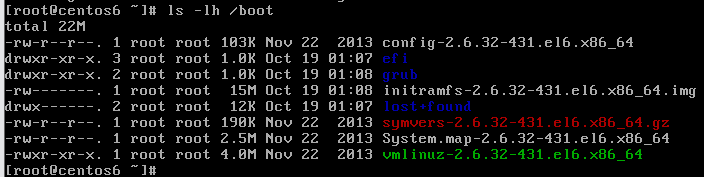
最后由initramfs的dracut系统去找真正的系统的/根目录，再切换根目录到真正的根上，这样就完成了正式系统的启动。

下图可见stage\_2文件位于/boot分区的grub子目录下，名为stage2，大小约124KB

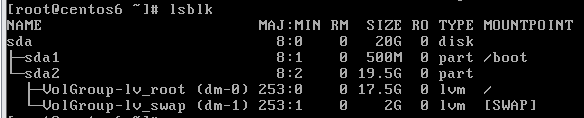


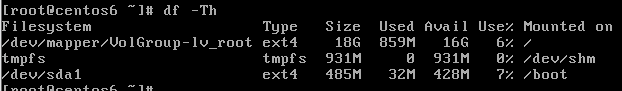
内核文件位于/boot分区里，名为vmlinuz-\*\*\*

initramfs文件和内核文件位于同一位置，名为initramfs-\*\*\*.img



Centos6在bios启动模式，一般的分区结构如下：





一般分2个分区，其中一个为标准分区，挂载在/boot目录下，我们称之为/boot分区，使用ext4文件系统，

而另一分区默认使用lvm逻辑卷去管理，默认卷组名VolGroup创建lv\_root和lv\_swap 2个逻辑卷，分别挂载在/根目录下和作为swap分区。

/boot分区必须为标准分区，即直接在分区上使用ext4或xfs等文件系统，不能用lvm的逻辑分区，为什么呢？因为centos6是使用grub引导的，grub的stage\_1和stage\_1.5的引导比较小，不能识别lvm逻辑卷，而grub还要到/boot分区里去找stage\_2及内核等文件，所以/boot分区不能为lvm分区。

内核启动后，内核再加载initramfs里的某驱动和工具，就能识别lvm分区，所以/根分区可以为Lvm分区（当然也可以是标准分区）

**内核的启动**：

内核vmlinuz和initramfs文件被grub引导加载到内存后，grub把引导权交给内核，内核名为vmlinuz时，末尾字母为z表示此内核文件是使用gzip压缩的，内核文件头部的代码会自动把此内核解压缩，内核解压后就开始做正式的工作了

主要是初始化系统中的各设备并做相关配置，如cpu，I/O，存储等，配置过程也会加载一些内核里带有的部分设备驱动，调用这些驱动去初始化相关设备。内核启动后会创建一个rootfs文件系统。

内核文件本身比较小，带的东西不多，所以把其他具体系统中要用到的工具及驱动放到了initramfs文件里，内核把initramfs文件解压到rootfs里，然后执行rootfs里面的/sbin/init程序，init程序加载完最后的一些驱动后，会去寻找/根分区，把/根分区挂载到/sysroot目录下，最后init会释放掉未使用的内存，并执行根切换（chroot /sysroot）

切换到真实的/根（root）上面，同时运行根分区里的/sbin/init程序，根分区里的/sbin/init程序运行后，就是系统的1号进程，真正的系统就启动了，接下来就是系统的事情了。（比如执行/etc/inittab及/etc/init.d/\*等脚本文件，启动指定的服务）

**小结**：

|  |  |
| --- | --- |
| 启动过程 | centos6的 |
| 1 | BIOS预启动程序加载指定磁盘的MBR扇区里的引导：grub-stage1 |
| 2 | stage1加载位于MBR之后的stage1.5引导 |
| 3 | stage1.5去加载/boot分区里的grub子目录下的stage2引导程序 |
| 4 | stage2读取同目录下的menu.lst（即grub.conf）配置菜单 |
| 5 | 当用户选择指定菜单项后，grub引导指定的内核及initramfs，若用户不选择，则启动默认的菜单项 |
| 6 | 内核文件解压后，初始化设备，加载驱动，解压initramfs到rootfs文件系统里，执行里面的/sbin/init |
| 7 | rootfs里的/sbin/init程序加载最后的驱动，寻找根分区，挂载到/sysroot目录下，清理内存，把当前的根目录切换到/sysroot下 |
| 8 | 系统切换到真正的根目录下后，执行真正根目录里的/sbin/init程序，这也是系统里pid为1的程序，它再去完成系统最后的启动工作 |

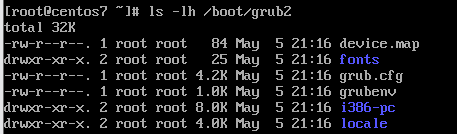
UEFI启动模式下，都差不多，就是grub引导的三个部分stage1，stage1.5和stage2都放在EFI分区下的/EFI/BOOT/bootx64.efi文件里，它们合并为一个文件了，uefi预启动程序加载这个efi文件（在x86\_64pc里名字为bootx64.efi，它是grub引导）然后grub引导接下来的工作就和bios里的一样了。具体的放到其他章节里讲吧。

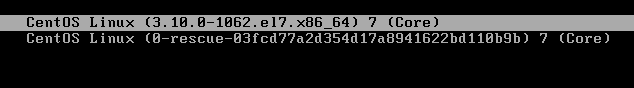
**Centos7系统启动过程**

centos7使用grub2作为引导，BIOS启动模式下，过程如下：

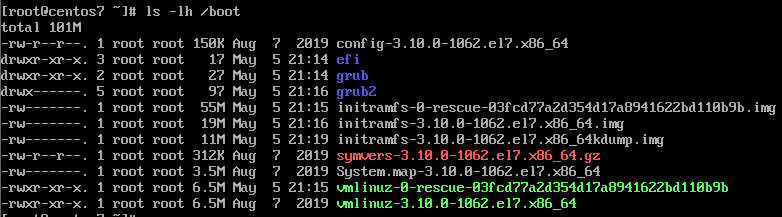
BIOS预启动程序读取启动磁盘上的MBR扇区，执行里面的引导代码（grub2的第一部分代码），mbr里的代码再加载mbr扇区后的第二阶段的代码。第二阶段的代码能识别文件系统，所以它能加载/boot分区里的某些模块及内核等文件

比如加载/boot分区的grub2子目录下的i386-pc目录里的其他模块，以及fonts文件等，再读取grub.cfg配置菜单，最后加载菜单项里指定的内核及initramfs文件，





内核及initramfs等文件位于/boot分区下，



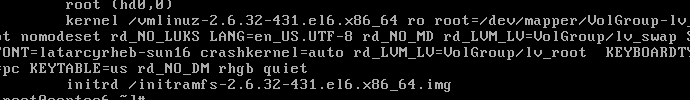
接下来的事情就和centos6的差不多了，只是启动的第一个程序不再是/sbin/init，而是/sbin/systemd程序

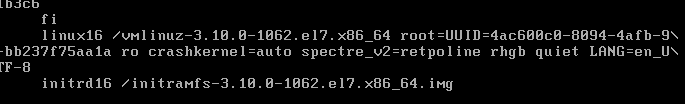
**小结**：

|  |  |
| --- | --- |
| 启动过程 | centos7的 |
| 1 | BIOS预启动程序加载指定磁盘的MBR扇区里的引导：grub2 |
| 2 | mbr里的引导加载位于MBR之后的第二阶段引导 |
| 3 | 第二阶段引导去加载/boot分区里的grub2子目录下的i386-pc子目录里的其他模块文件，grub2正式加载完毕 |
| 4 | grub2读取同/boot/grub2下的grub.cfg配置菜单 |
| 5 | 当用户选择指定菜单项后，grub2引导指定的内核及initramfs，若用户不选择，则启动默认的菜单项 |
| 6 | 内核文件解压后，初始化设备，加载驱动，解压initramfs到rootfs文件系统里，执行里面的/sbin/systemd |
| 7 | rootfs里的/sbin/systemd程序加载最后的驱动，寻找根分区，挂载到/sysroot目录下，清理内存，把当前的根目录切换到/sysroot下 |
| 8 | 系统切换到真正的根目录下后，执行根目录里的/sbin/systemd程序，这也是系统里pid为1的程序，它再去完成系统最后的启动工作 |

**26.传给内核的参数及initramfs**

grub及grub2的启动菜单项如下：





grub为：

kernel /vmlinuz-xxx 参数1 参数2 等其他参数

initrd /initramfsxxx.img

grub2为：

linux16 /vmlinuz-xxx 参数1 参数2 等其他参数

initrd16 /initramfsxxx.img

在BIOS启动模式下，

grub使用kernel命令来指定要加载的内核，在内核后可传递其他参数，

grub使用initrd命令来指定要加载的initramfs文件

grub2使用linux16命令来指定要加载的内核，在内核后可传递其他参数，

grub2使用initrd16命令来指定要加载的initramfs文件

要传递给内核的参数，只和具体的内核版本有关，而跟grub的版本或其他引导无关，无论我们用grub去引导centos6还是grub2去引导centos6，只要内核版本是一样的，其传入的参数就是同一个写法，与grub版本无关！

**传给centos6内核（2.6.32）的参数如下**：

root=\*\*\* //表示真正的系统的根分区，可以使用UUID，可以使用

磁盘号，可以使用分区的LABEL

ro或rw //加载真正的根分区时的读写属性，默认为ro

rd\_LVM\_LV=\*\*\* //要激活的lvm逻辑卷

LANG=\*\*\* //设置系统的语言环境

rhgb //以图形界面方式启动系统

quiet //以文本方式启动系统，且禁止输出大多数的log信息

crashkernel=\*\*\* //给kdump服务用的一个参数，值为auto时表示

根据系统内存自动reserve一些内存给kernelcrash用

其他的参数就不列举了

**传给centos7内核（3.10.0）的参数如下**：

root=\*\*\* //表示真正的系统的根分区，可以使用UUID，可以使用

磁盘号，可以使用分区的LABEL

ro或rw //加载真正的根分区时的读写属性，默认为ro

rd.lvm.lv=\*\*\* //要激活的lvm逻辑卷

LANG=\*\*\* //设置系统的语言环境

rhgb //以图形界面方式启动系统

quiet //以文本方式启动系统，且禁止输出大多数的log信息

crashkernel=\*\*\* //给kdump服务用的一个参数，值为auto时表示

根据系统内存自动reserve一些内存给kernelcrash用

**其实**initrd或initrd16命令后的initramfs文件也是传递给内核的参数，告诉内核要使用这个**initramfs文件**。

initramfs可以在内核启动时提供一个用户态环境，借助它可以完成一些在内核启动阶段不易完成的工作，比如：

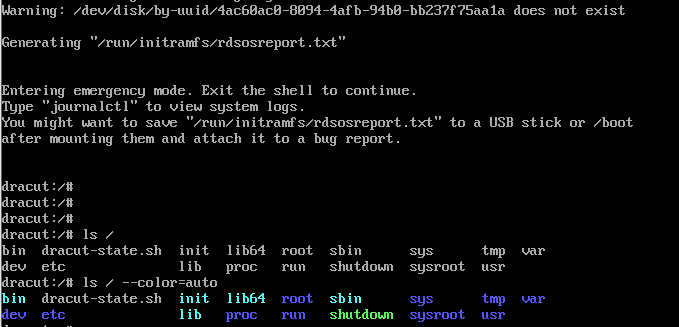
\*加载第三方驱动，

\*定制化启动过程，

\*制作一个极简的rescue shell救援系统

\*执行某些命令，比如/sbin/init或systemd程序

从centos6开始不再使用原来的mkinitrd工具，而是使用**dracut工具**去生成initramfs文件，所以，当我们的grub配置菜单项写错了或磁盘分区的uuid变了，则在加载内核及initramfs后，很可能因为找不到真正的根分区或其他错误而停留在dracut的极简救援系统界面：



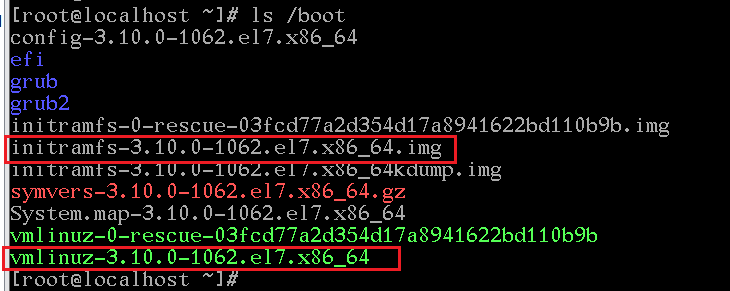


这个极简的系统里只有100多个命令/工具，且根目录下的所有文件都是由initramfs-xx.img解压得来的，

**dracut工具**会尝试尽可能少地将正式系统中的必备工具放入initramfs文件中，在centos6或7系统里，执行以下命令可以生成当前系统的initramfs文件：

# dracut -f /某dir/initramfs-$(uanme -r).img $(uname -r)

表示生成适配当前运行的内核的 initramfs文件。当我们更新系统内核后，也会生成新的initramfs文件。所以内核的版本和initramfs的版本要一致。



因为initramfs文件是根据正式的系统去生成的，所以在我们安装系统时，最后一步就是生成initramfs文件。当我们升级系统后（比如从centos6升级到7）如果正式的系统中缺少某些lib链接文件，则生成的initramfs文件里也会缺少这些链接文件，导致某些功能无法正常启动，比如lvm功能，这时，如果系统的根目录分区为lvm逻辑卷，而initramfs里的lvm工具又无法正常启动，则系统的根目录也无法被激活挂载，系统启动失败。

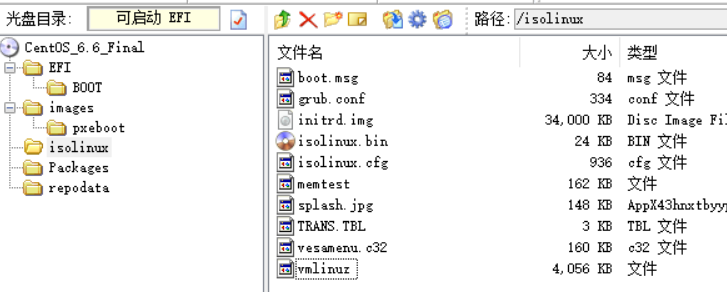
**linux内核对initramfs-xxx.img文件的处理流程**：

首先内核先生成一个rootfs文件系统，rootfs是内核里的一个特殊的文件系统，基于内存的文件系统，是内核启动的初始根文件系统。

内核再判断initramfs文件的格式，若为cpio格式的文件，则把initramfs的内容解压释放到由内核生成的rootfs中，最后执行其中的/sbin/init或/sbin/systemd程序。

**27. Centos系统的安装原理**

这得先看一下centos的安装光盘镜像文件的结构：（以centos6为例）



在centos6的安装光盘里就几个主要的目录：

/EFI目录是给UEFI启动时准备的，里面主要是grub启动相关的文件

/images目录里面有pxeboot子目录，里面有内核及initrd文件

/isolinux目录下也有内核及initramfs文件，

/Packages目录里是各rpm软件包， /repodata目录里是关于Packages目录的yum仓库信息文件。

安装光盘里的mbr引导为isolinux引导。

当使用BIOS启动模式时，这个isolinux引导会去读取/isolinux目录下的配置文件isolinux.cfg ，并在用户选择install centos时加载内核vmlinuz及initrd.img文件。

如果是UEFI启动模式，会去寻找光盘里的/EFI目录，然后找/EFI/BOOT/Bootx64.efi引导，这个bootx64.efi引导是grub引导。加载引导后，grub再读取配置文件grub.conf，当用户选择要安装系统时，grub会加载光盘里的/images/pxeboot/目录下的内核及initrd文件。

（/isolinux目录下的内核与/images/pxeboot目录下的内核其实是一样的，是同一个文件，initrd.img文件也是一样的。只是目录不一样而已）

安装光盘里的initrd文件和正式系统中的initramfs文件原理差不多，只是功能不一样，或者说它们执行的程序不一样，正式的系统中的initramfs文件被加载后，会去找真正的系统的根分区，并挂载到/sysroot下，最后切换根。

而安装光盘里的initrd文件在加载后，会执行 安装向导 程序。并在用户的操作下完成正式系统的安装。

安装向导会找到安装源，默认为当前的光盘，然后根据安装的版本（如最小化版本或桌面版）去安装指定数量的rpm软件包，软件包位于光盘的/Packages目录下。安装完成后，再切换根到正式的系统下，生成initramfs文件。

**28. Syslinux引导**

在进行centos系统的安装 之前，我们有必要先了解一下Syslinux引导。

Syslinux是一个项目，官网为syslinux.org，该项目包含以下启动引导程序：

|  |  |
| --- | --- |
| 引导名称 | 应用场景 |
| Syslinux | 安装于MS-DOS的FAT文件系统分区里， |
| Isolinux | 用于光盘介质时的引导 |
| Pxelinux | 用于从网络启动 |
| Extlinux | 安装于Ext文件系统分区里 |
| Memdisk | 装载镜像文件到内存里再引导里面的系统 |

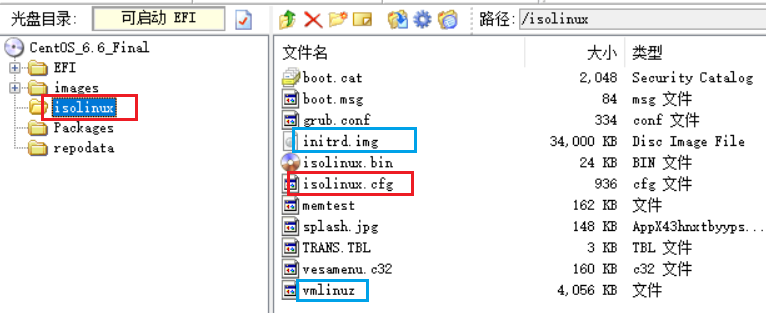
也就是说以上5种引导的区别是应用的场景不同，不同的文件系统或介质 使用不同的引导，它们都是属于Syslinux这个项目，由Syslinux团队开发的。

syslinux引导可以引导linux系统，也可使用chainloader引导其他的bootloader

配置文件为对应的引导名称加.cfg ，如syslinux引导的配置文件为syslinux.cfg

isolinux引导的配置文件为isolinux.cfg。配置文件的内容和命令都差不多。

我们还是先看一下centos6的安装光盘文件结构吧：



安装光盘的/isolinux目录为isolinux引导的默认工作目录，具体的文件及用途：

isolinux.cfg //isolinux引导的配置文件

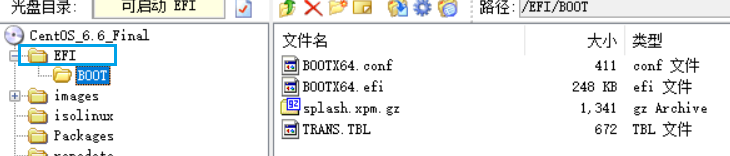
vesamenu.c32 //窗口模块

splash.jpg //启动菜单背景图片

vmlinuz //centos6的内核文件

initrd.img //initrd文件，里面有安装向导程序

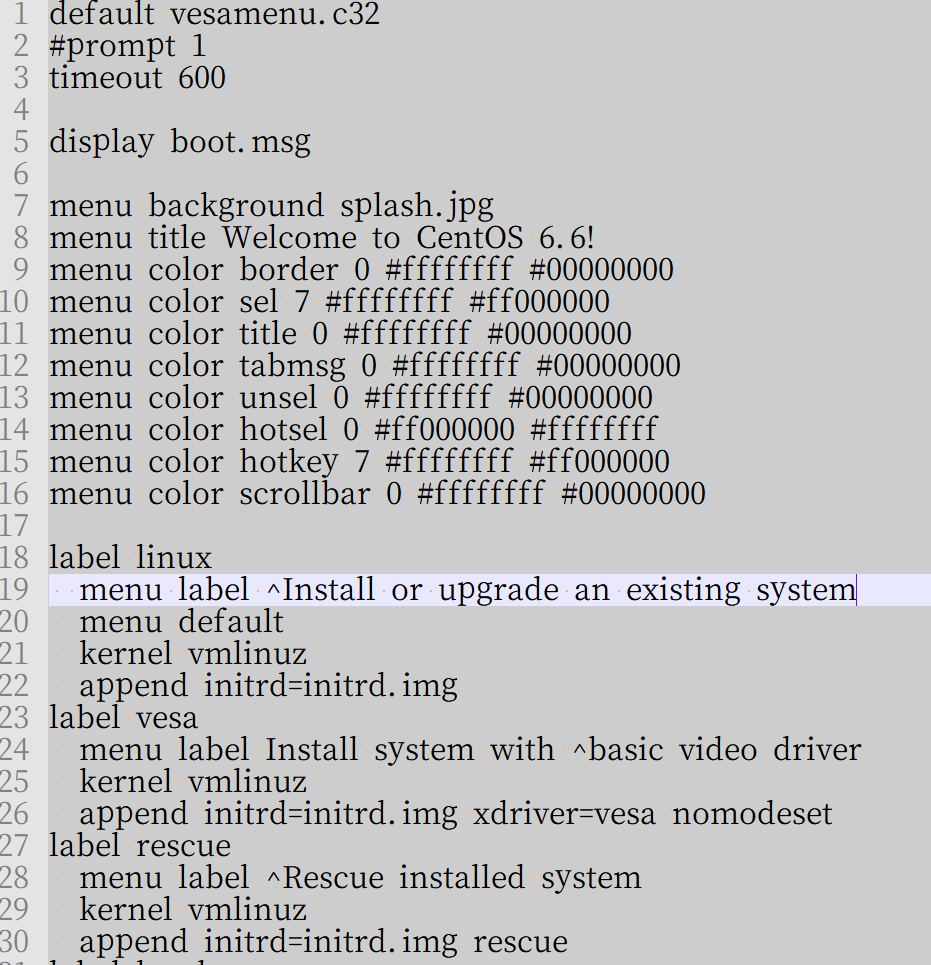
其他的一般用不上，



而光盘里的/EFI/BOOT目录下的bootx64.efi引导为grub引导，配置文件为bootx64.efi，本章先不讲。

BIOS启动模式下使用的是isolinux引导，UEFI启动模式下才使用grub引导。

isolinux.cfg配置文件讲解：



default //命令指定使用的容器模块为vesamenu.c32

timeout 600 //指定菜单停留时间，时间单位为10分之1秒，

600则表示为60秒

menu \*\* //指定菜单的样式，如无特殊要求，可忽略

label \*\* //定义一个菜单项，指定其label

menu label \*\*\* //指定该菜单项的显示名称

menu default //指定该菜单项为默认选择的菜单

kernel vmlinuz //指定要加载的linux内核，相对路径则表示

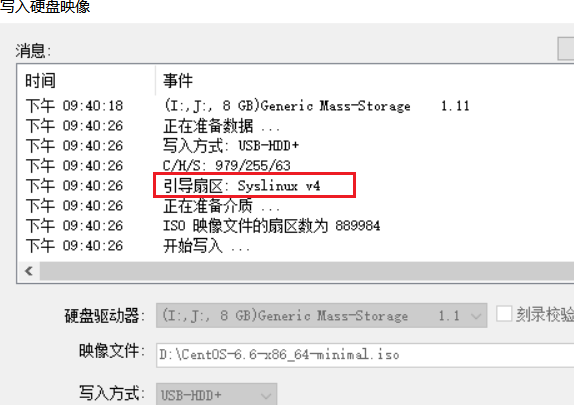
和该配置文件相同的目录下

append \*\*\* //给该Linux内核传入的参数，包含了initrd文件

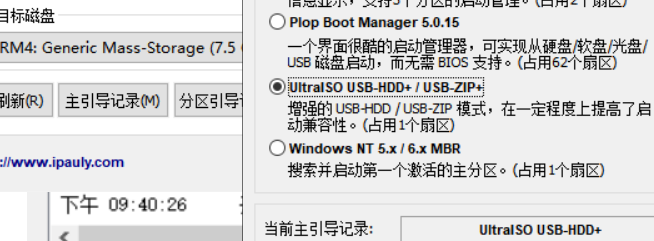
就讲这么多吧，了解就行，真正使用的时候也主要是用来安装系统的，正式的系统一般也不用syslinux作为引导了，而是使用grub和grub2。所以本章不想深入讲解syslinux引导。

我们在使用刻录工具把centos6安装光盘iso文件刻入U盘时，会有什么变化呢？

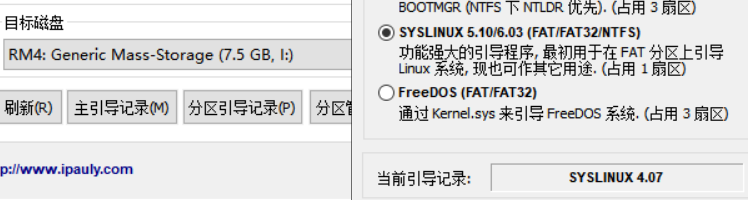
我们试试，使用UltraIso工具把centos6.iso刻录到U盘里，我们在第14章已经知道，直接解压.iso文件时，是只有里面的文件，而不包含引导的，如果刻入到U盘里，会带有引导，那么用Ultraiso写入U盘时，会是什么引导呢？



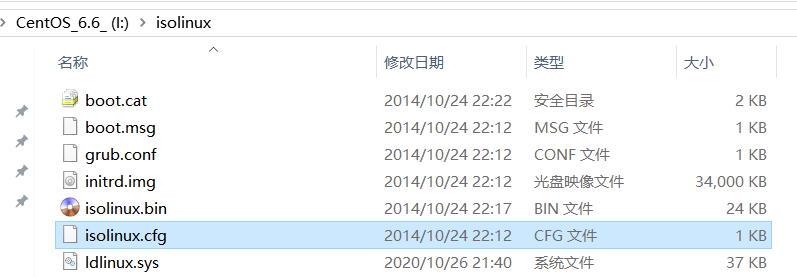
通过刻录时的输出信息，我们大概可以猜到写入到U盘后，U盘上使用的引导为Syslinux 第4版，用Bootice工具验证一下



上图可见，U盘的mbr主引导为Ultraiso USB-HDD+的引导，然后此mbr引导再去引导位于分区上的syslinux 4.07的引导。



也就是说，直接使用光盘安装时，使用的是isolinux引导，刻录到U盘时，由Ultraiso工具把它转为了syslinux引导，然后在U盘的/isolinux目录下多了一个syslinux.cfg配置文件，而不是光盘里的isolinux.cfg



我们当然也可以不使用Ultraiso工具去刻录了，直接使用syslinux引导，把syslinux引导安装到U盘的mbr里或者分区里，再把centos6安装光盘文件解压到u盘上就行了。

怎么**手动安装syslinux引导**到u盘上呢？

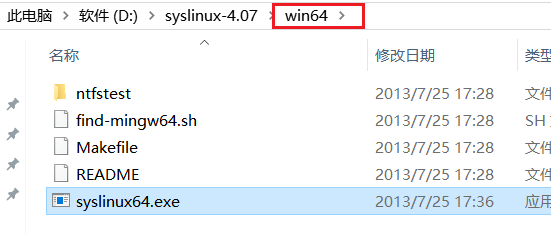
\*先把centos6.iso镜像文件解压到目标U盘里，本例中为i盘

\*下载syslinux工具https://mirrors.edge.kernel.org/pub/linux/utils/boot/syslinux/

\*windows版的下载.zip的包，比如4.07的版本

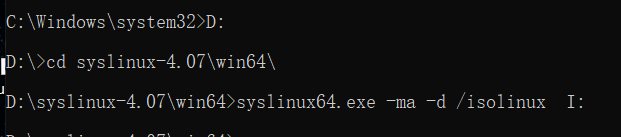


然后解压到windows上的D盘下：



在解压目录里的win64或win32里有syslinux64.exe程序

以管理员身份运行cmd命令行，进入解压目录下的win64目录，执行以下命令



> syslinux64.exe -fma -d /isolinux I: //表示把引导写入I:盘，i盘为目标U盘的盘符，配置文件的目录用-d 指出，本例为/isolinux目录，

如果是6.x版本的syslinux，则使用以下命令：

解压目录的\bios\win64子目录下：

> syslinux64.exe --mbr --active --directory /isolinux/ --install I:



然后要安装系统时，插入U盘，选择从目标U盘的分区启动就行了。不过，这个不保证在每台机器上都能正常引导。（一定要选择从目标U盘的分区启动）

**最后的问题**：

当我们使用使用Ultraios刻录镜像.iso文件到U盘里后，该U盘能在bios模式下正常安装系统，而在UEFI启动时无法安装，无法进入到安装界面，这是为什么？

因为使用Ultraios工具刻录时，它只转换了isolinux引导为syslinux，而没有修改EFI/Boot/目录下的引导配置文件，导致这里出了问题，Ultraios在刻录之前已经把目标U盘格式化为FAT32文件系统，而根据第10章的知识，FAT32分区的卷标最多只有11个字符，



而如果原来的grub配置文件里如果使用的卷标名长度大于11字符时，就会有缺失，所以会找不到目标U盘，无法正常进入安装界面。

**29. 初识grub引导**

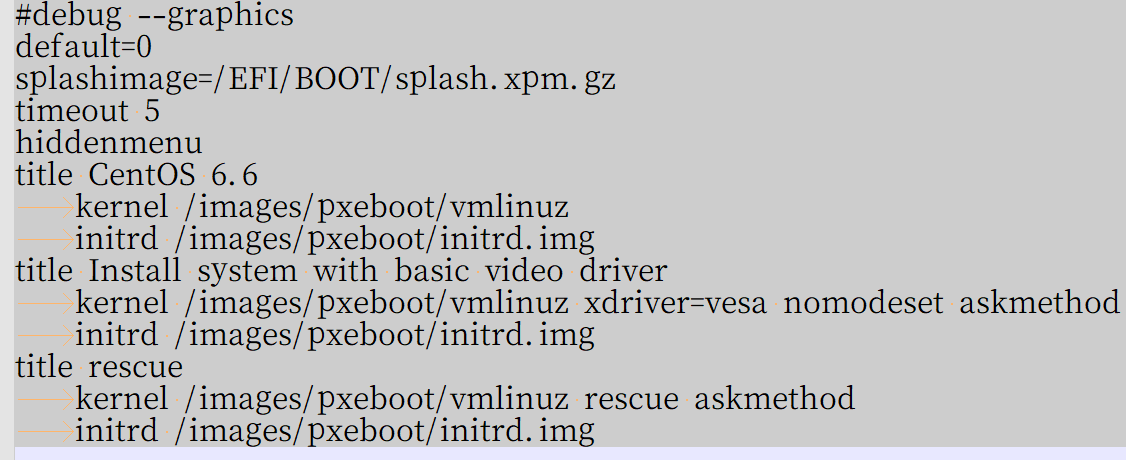
GRUB全称GRand Unified Bootloader，是由GNU组织相关人员开发的一个功能强大的引导程序，它支持多种文件系统，能引导多种操作系统。

centos6使用的引导正是grub，在Centos6的安装光盘里，grub引导只在UEFI启动模式下才用到，装完系统后，正式的Centos6系统就只使用grub引导，而无论是哪种启动模式。



如上图，当使用UEFI启动时，预启动程序就去安装介质里找/EFI/BOOT/bootx64.efi文件，这个文件在这里就是grub引导，它的配置文件为同目录下的bootx64.conf

grub的配置先不讲太多，主要配置如下：



default=0 //表示默认选择第一个菜单项，从0开始编号

splashimage= \*\*\* //表示菜单使用的背景图片，图片格式为.xpm，

用gz压缩的

timeout 5 //菜单停留的时间，单位为秒

hiddenmenu //隐藏菜单，只有当用户按下按键时才显示

title XXX //菜单项名称

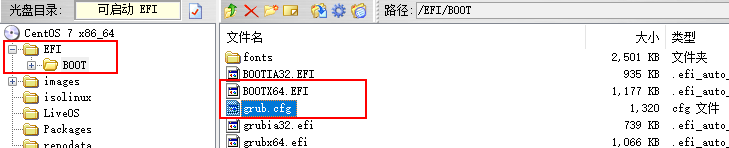
kernel \*\*\* //要加载的内核文件，可带要传给内核的参数

initrd \*\*\* //要加载的initrd文件，功能和第26章的initramfs类似

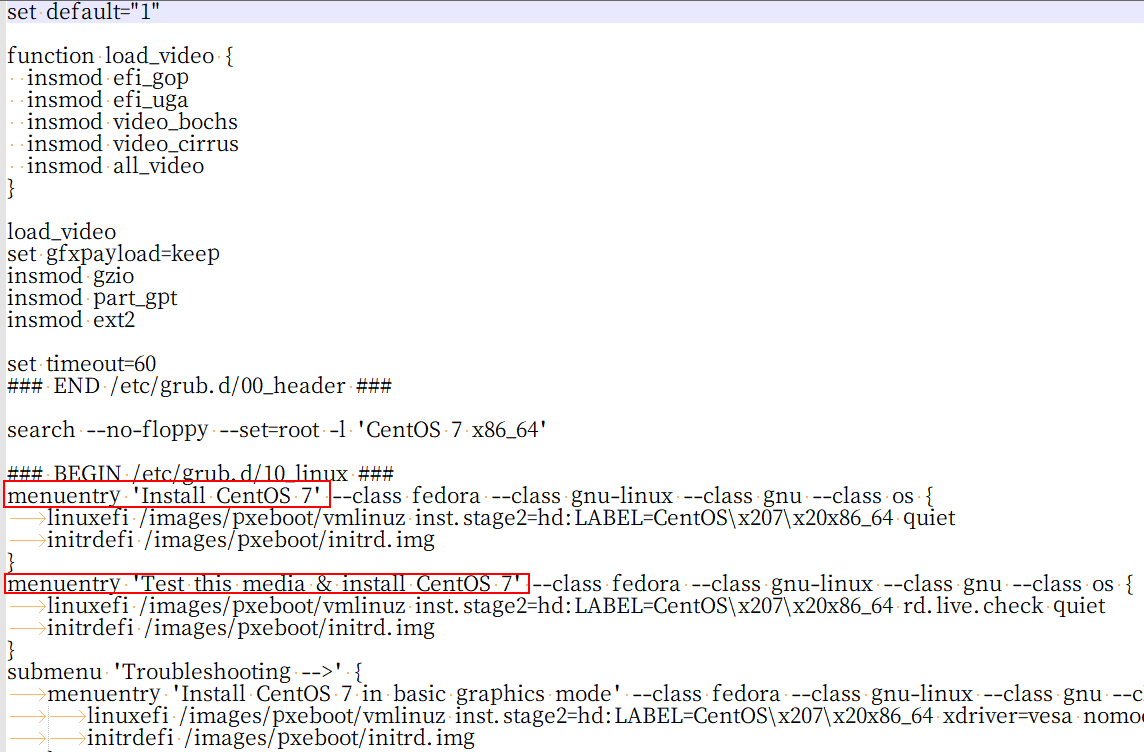
这里给内核传递的参数好像和第26章讲的不一样啊，对，不一样，这里的内核及initrd文件主要功能是用来安装操作系统的，而不是正常的启动正式系统。所以参数不太一样。

**31.初识grub2**

Centos7的安装光盘在BIOS模式下使用的是isolinux引导，刻录到U盘后，转为了syslinux引导，可以查看第28章的相关知识。当使用UEFI模式启动时，使用的是/EFI/BOOT/下的Bootx64.efi引导，这个就是grub2引导，默认配置文件为/EFI/BOOT/grub.cfg



配置文件讲解：



set default="1" //表示默认选择第1个菜单项，从1开始编号

function load\_video { ... } //定义一个函数load\_video

load\_video //使用前面定义的函数

set gfxpayload=keep //设置gfxpayload样式

insmod xxx //加载grub2的模块

//比如安装介质为gpt分区时，要加载part\_gpt模块

//安装源文件所处文件系统为ext2则要加载ext2模块

set timeout=60 //设置菜单停留时间，单位秒

search --no-floppy --set=root -l 'Centos 7 x86\_64'

//-l表示根据label搜索，搜索到label为'Centos 7 x86\_64'的分区就设置为当前的root，从这开始，下文的相对路径的文件就以此root为根目录

menuentry 'xxx' //声明一个菜单项，之后的定义用{ }花括号括起来，菜单项声明后面的--class表示菜单风格

{

linuxefi /xxx/vmlinuz 参数xx //加载linux内核

initrdefi /xxx/initrd.img //加载initrd.img文件

}

//在文件路径前面如果不加上(hd0,1)之类的磁盘及分区信息，则表示使用上文root设置的分区

**传给内核的参数**：

inst.stage2=hd:LABEL=CentOS\x207\x20x86\_64 表示告诉内核，接下来要安装系统或其他操作的镜像源文件所在位置，使用hd:LABEL=表示根据分区的label去找，grub2的label不能含空格，所以当分区的label有空格时要用转义符号\x20表示，

所以CentOS\x207\x20x86\_64就表示CentOS 7 x86\_64

quiet //表示以文本方式启动系统，且禁止输出大多数的log信息

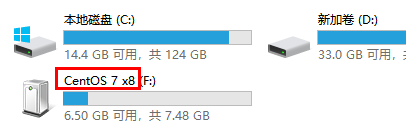
rd.live.check //表示进入live OS系统并进行介质检查

rescue //表示进入救援系统

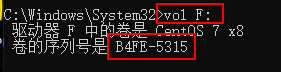
**注意**：

inst.stage2=hd:LABEL=\*\*\* //默认是使用label去找安装源文件所在分区

默认配置里写的是Centos 7 x86\_64，也就是安装光盘的Label名称，如果使用刻录工具将安装光盘.iso文件刻录到U盘后，可能Label就变了或者有残缺，因为U盘上的分区文件系统可能是FAT的，而FAT文件系统允许的Label卷标最多只能有11个字符，而CentOS 7 x86\_64有15个字符，在U盘的FAT分区里就只显示出CentOS 7 x8 这几个字符，所以grub2会找不到目标分区，无法进入安装界面。



**解决方法**可以有：（修改配置文件/EFI/BOOT/grub.cfg）



cmd命令：vol F: //查看目标分区的卷标UUID

inst.stage2=hd:UUID=B4FE-5315 //使用UUID去查找目标分区

