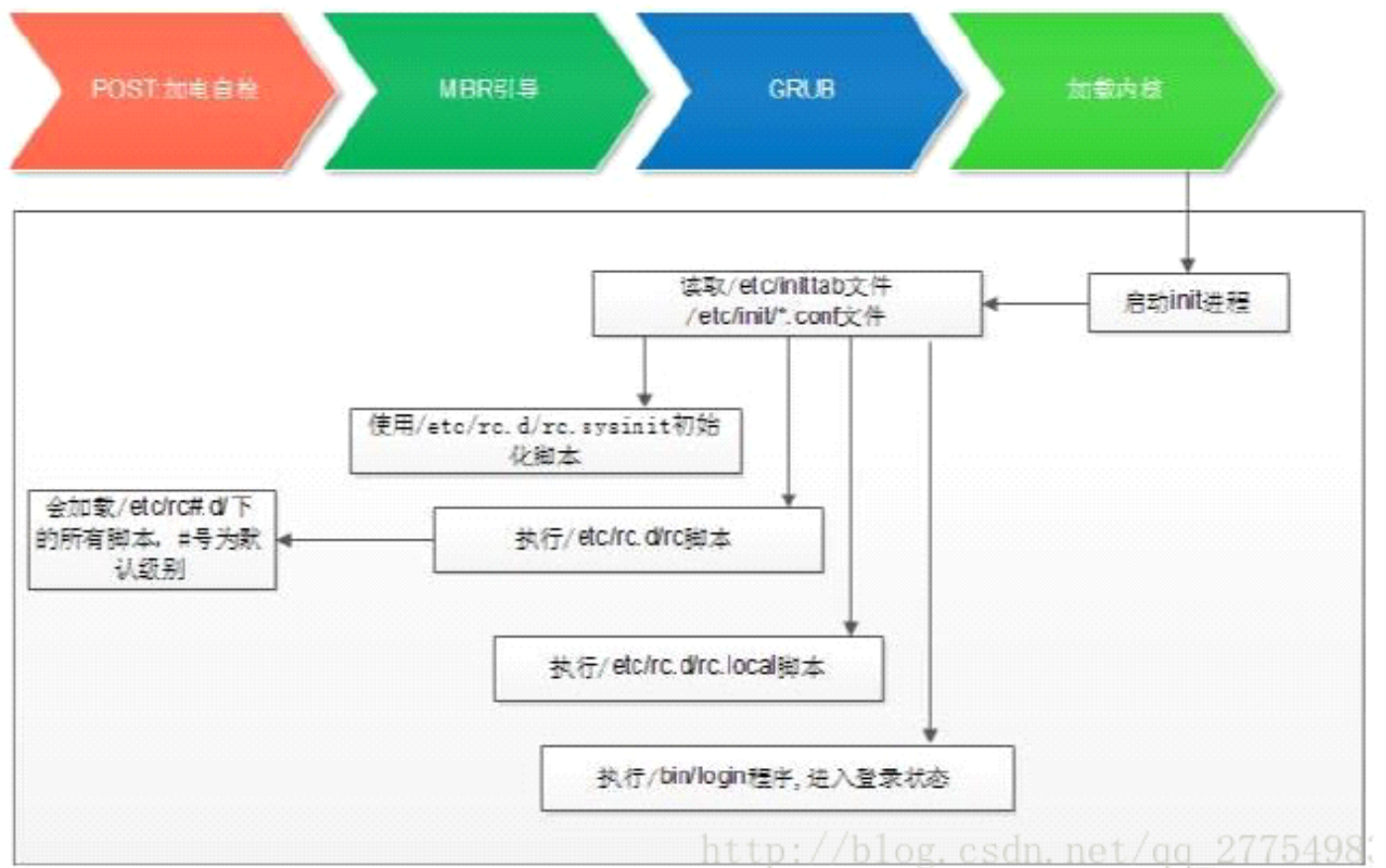


Linux开机顺序 (rhel6 upstart)



1、POST初始化硬件，安全自检完成后去查找相应需要启动的硬件设备（光盘、u盘、硬盘、网卡）

2、假设我们这里是设置的硬盘启动，通过硬盘启动的方式会把导入到内存中，然后BIOS任务完成，以后的工作移交给MBR。

主引导记录（MBR）是一种特殊类型的引导扇区，硬盘的第一个扇区，MBR保存硬盘分区的信息和主引导程序(grub)用于加载已安装的操作系统(格式化分区命令不会擦除MBR信息，因此此特殊空间不属于任何分区)。MBR保存着系统的主引导程序和分区表（grub 446字节，分区表64字节）。

3、mbr读取grub，让用户选择操作系统（系统默认选择）

4、执行/sbin/init程序。这个init是系统的第一个进程，PID为1，以后所有的进程都是在它的基础上产生的,或者说都是它的子进程。

5、init进程读取的第一个脚本就是/etc/inittab。inittab文件里面定义了系统默认运行级别，init进程根据inittab这个配置文件启动第几个运行级别

Linux的运行级别介绍：所谓运行级别，简单点来说，运行级别就是操作系统当前正在运行的功能级别。级别是从0到6，具有不同的功能。

Linux下的7个运行级别：

0:系统停机状态，系统默认运行级别不能设置为0，否则不能正常启动，机器关闭。

1:单用户工作状态，root权限，用于系统维护，禁止远程登陆，就像Windows下的安全模式登录。

2:多用户状态，没有NFS支持。

3:完整的多用户模式，有NFS，登陆后进入控制台命令行模式。

4:系统未使用，保留一般不用，在一些特殊情况下可以用它来做一些事情。例如在笔记本电脑的电池用尽时，可以切换到这个模式来做一些设置。

5:X11控制台，登陆后进入图形GUI模式，XWindow系统。

6:系统正常关闭并重启，默认运行级别不能设为6，否则不能正常启动。运行init6机器就会重启。

6、第一个根据运行级别启动的脚本是/etc/rc.d/rc.sysinit(决定了系统的一些基本配置)，这个脚本将完成以下一些工作：设置

系统时钟、设置主机名、激活udev、激活selinux、定义网络配置参数、挂载分区、激活raid、激活lvm、激活磁盘配额、加载swap等

7、/etc/rc#.d/

[root@localhost ~]# ls /ect/rc5.d/

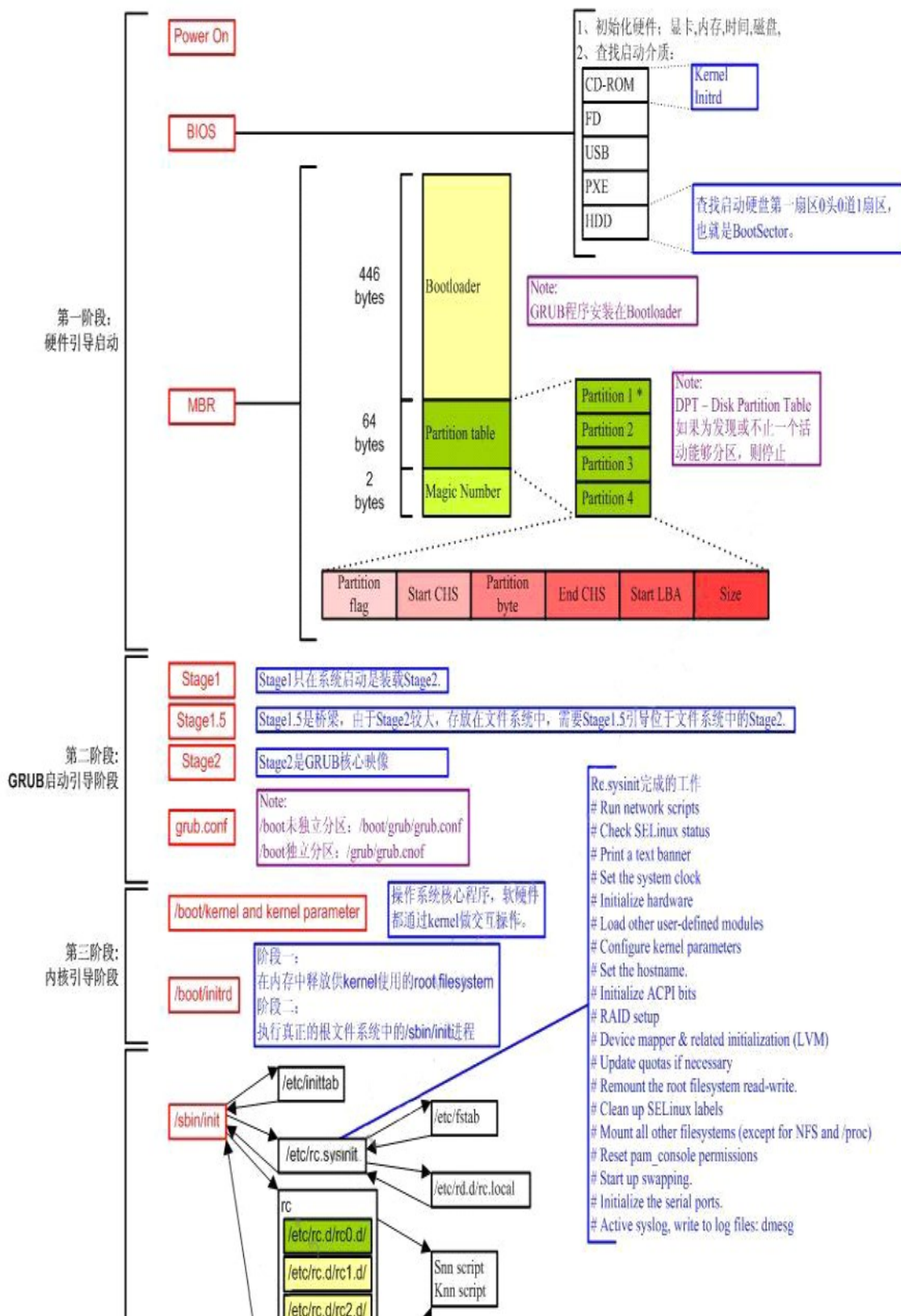
#代表数字，一个数字代表一个运行级别，共7个运行级别

目录里面定义的是各种服务的启动脚本，可以ls查看，S开头代表开机启动的服务，K开头的是关机要执行的任务。

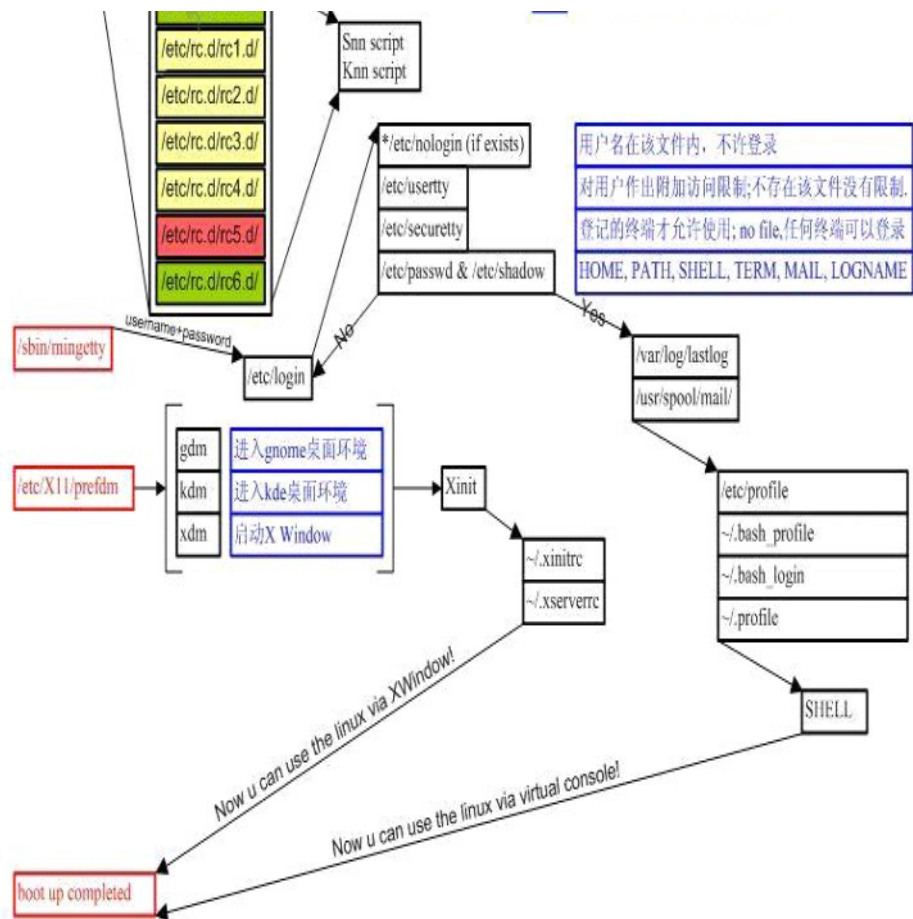
8、/etc/rc.d/rc.local文件

这里面可以自定义开机启动的命令。

9、执行/bin/login。等待用户登录



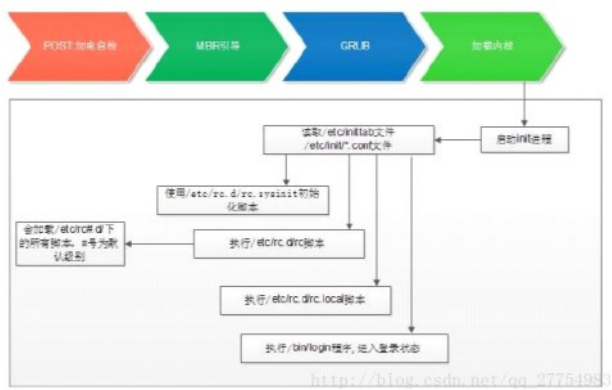
第四阶段:
Sys V init初始化阶段



第五阶段:
启动完成

CentOS6启动流程

2019年6月15日 莫宇剑 著 9:41



看到图片应该了解的差不多了，我再详细讲一下吧

1. post加电自检

这个过程是开机后，BIOS或UEFI进行硬件检查的阶段

2. MBR引导

自检硬件没有问题时候，这里以BIOS为例，BIOS将会直接去找硬盘的第一个扇区，找到前446字节，将MBR加载到内存中，MBR将告诉程序下一阶段去哪里找系统grub引导。此阶段属于grub的第一阶段。grub还有1.5阶段和2阶段。

3. GRUB引导

grub第1.5和2阶段，信息默认存放在扇区中，如果使用grub-install生成的2阶段的文件是存放在/boot分区中的。为了加载内核系统，不得不加载/boot分区，而加载/boot分区，需要有/boot分区的驱动，/boot分区驱动是放在/boot分区中的，啊，我们好像进入了死循环了，Linux是怎么解决的呢？就是靠放在1.5阶段中的数据，是放在第一个扇区后的后续扇区中，具体占用多少字节，不太清楚，只知道1.5阶段和2阶段总共27个扇区。

stage1.5:

mbr之后的扇区，识别stage2所在的分区上的文件系统

stage2:

开机启动的时候看到的Grub选项、信息，还有修改GRUB背景等功能都是stage2提供的，stage2会去读入/boot/grub/grub.conf或者menu.lst等配置文件

4. 读取grub.conf文件

读取grub.conf文件以确定内核启动的参数，准备启动内核

5. 启动内核

加载内核，核心开始解压缩，启动一些最核心的程序。

因为为了让内核足够轻小，硬件驱动并没有放在内核文件里面，我们可以看到内核很小，才4M左右，我们可以想象Windows中的驱动，安装系统时候还需要使用驱动软件下载好长时间呢

因此需要使用/initramfs-2.6.32-696.el6.x86_64.img来驱动硬件

```
[root@CentOS6 ~]# ll -h /boot/vmlinuz-2.6.32-696.el6.x86_64
-r-xr-xr-x. 1 root root 4.1M Jul 8 21:06 /boot/vmlinuz-2.6.32-696.el6.x86_64
```

1
2

6. 加载伪文件系统（ramdisk），

内核已将启动起来了，再调用ramdisk文件，尝试驱动所有的硬件设备，到这一步，内核起来了，所有驱动也装上了，因此后面的启动就可以交给程序了

7. 启动init进程

grub中默认指定init=/sbin/init程序，可以在grub.conf中kernel行自定义执行程序init=/bin/bash,此时可以绕过下面步骤直接进入bash界面。内核源代码文件中显示996行左右，规定了init启动的顺序，/sbin/init->/etc/init->/bin/init->/bin/sh,/bin/bash没有写，应该是和/bin/sh一样吧

(1) 读取/etc/inittab文件

inittab文件里面定义了系统默认运行级别，这一步做了一些工作如下：

- a) 初始运行级别 (RUN LEVEL)
- b) 系统初始化脚本
- c) 对应运行级别的脚本目录
- d) 定义UPS电源终端/恢复脚本
- e) 在虚拟控制台生成getty, 以生成终端
- f) 在运行级别5初始化X

- 1
- 2
- 3
- 4
- 5
- 6

(2) 执行/etc/rc.d/rc.sysinit程序

系统初始化一些脚本，主要完成以下工作

- a) 设置主机名
- b) 设置欢迎信息
- c) 激活udev和selinux可以在grub.conf中, kernel行添加selinux=0以关闭selinux
- d) 挂载/etc/fstab文件中定义的文件系统
- e) 检测根文件系统，并以读写方式重新挂载根文件系统
- f) 设置系统时钟
- g) 激活swap设备
- h) 根据/etc/sysctl.conf文件设置内核参数
- i) 激活lvm及software raid设备
- j) 加载额外设备的驱动程序
- k) 清理操作

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11

(3)/etc/rc#.d/文件（各种服务）

里面定义的是各种服务的启动脚本，可以ls查看，S开头代表开机启动的服务，K开头的是关机要执行的任务。#代表数字，一个数字代表一个运行级别，共7个运行级别，这里就不多说了

(4)/etc/rc.d/rc.local文件

这里面可以自定义开机启动的命令。

8. 执行/bin/login

执行/bin/login程序，等待用户登录

CentOS7启动流程

CentOS7和CentOS6启动流程差不多，只不过到init程序时候，改为了systemd，因此详细解释一下systemd后的启动流程

1. uefi或BIOS初始化，开始post开机自检
2. 加载mbr到内存
3. GRUB阶段
4. 加载内核和initramfs模块
5. 内核开始初始化，使用systemd来代替centos6以前的init程序
- (1) 执行initrd.target

包括挂载/etc/fstab文件中的系统, 此时挂载后，就可以切换到根目录了

- (2) 从initramfs根文件系统切换到磁盘根目录
- (3) systemd执行默认target配置

centos7表面是有“运行级别”这个概念，实际上是为了兼容以前的系统，每个所谓的“运行级别”都有对应的软连接指向，默认的启动级别时/etc/systemd/system/default.target, 根据它的指向可以找到系统要进入哪个模式模式：

- 0 ==> runlevel0.target, poweroff.target
- 1 ==> runlevel1.target, rescue.target
- 2 ==> runlevel2.target, multi-user.target
- 3 ==> runlevel3.target, multi-user.target
- 4 ==> runlevel4.target, multi-user.target
- 5 ==> runlevel5.target, graphical.target
- 6 ==> runlevel6.target, reboot.target

(4) systemd执行sysinit.target

有没有很眼熟？是的，在CentOS6上是被叫做rc.sysinit程序，初始化系统及basic.target准备操作系统

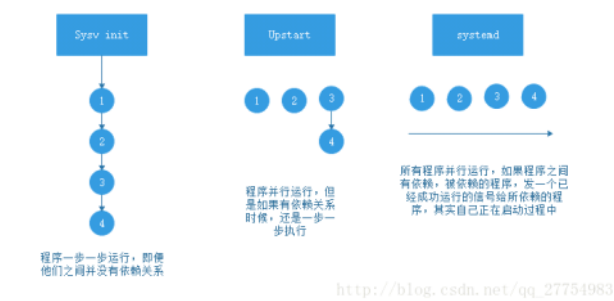
- (5) systemd启动multi-user.target下的本机与服务器服务
- (6) systemd执行multi-user.target下的/etc/rc.d/rc.local

6. Systemd执行multi-user.target下的getty.target及登录服务

getty.target我们也眼熟，它是启动终端的systemd对象。如果到此步骤，系统没有被指定启动图形桌面，到此就可以结束了，如果要启动图形界面，需要在此基础上启动桌面程序

7. systemd执行graphical需要的服务
CentOS6，7启动区别

系统启动和服务守护进程管理器，它不同于centos5的Sysv init，centos6的Upstart（Ubuntu制作出来），systemd是由Redhat的一个员工首先提出来的，它在内核启动后，服务什么的全都被systemd接管，kernel只是用来管理硬件资源，相当于内核被架空了，因此linus很不满意Redhat这种做法。
Sysv init运行程序顺序：



主要的区别就在这了。

如果有发现错，请提出来，共同进步^_^

作者：烟雨醉梦里
来源：CSDN
原文：https://blog.csdn.net/qq_27754983/article/details/75212666
版权声明：本文为博主原创文章，转载请附上博文链接！