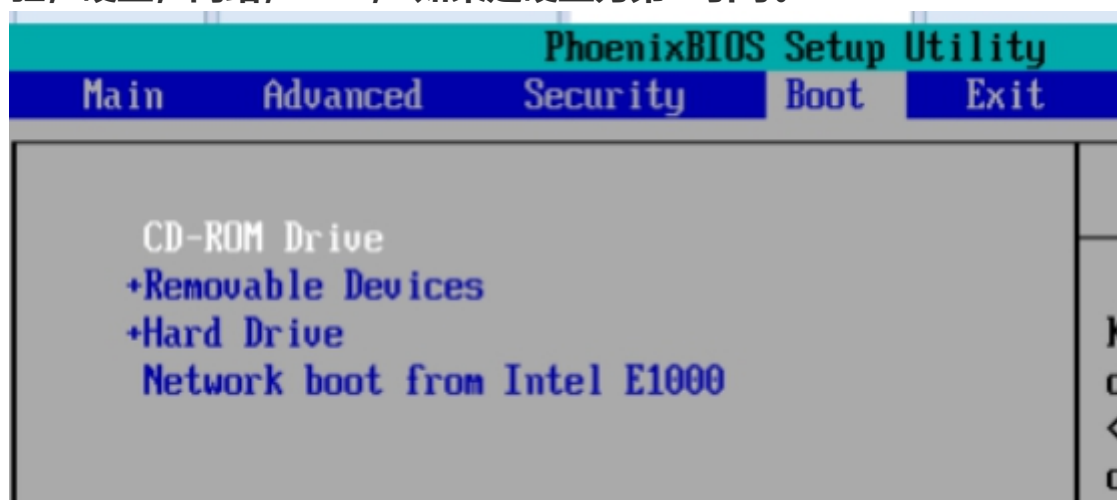


centos6 系统启动过程及相关配置文件

centos6 系统启动过程

1. 加载 BIOS 的硬件信息，根据设定取得第一个可开机引导设置，如：光驱，硬盘，网络，USB；如果是硬盘为第一引导。



2. 读取硬盘中 MBR 的 boot Loader 就是 grub 引导

GRUB (GRand Unified Bootloader 简称“GRUB”) 是一个来自GNU 项目的多操作系统启动程序。

MBR 的硬盘的 0 柱面、0 磁头、1 扇区称为主引导扇区（也叫主引导记录 MBR）。它由三个部分组成，主引导程序、硬盘分区表 DPT (Disk Partition table) 和硬盘有效标志 (55AA)。

互动：为什么 MBR 分区表，只能分 4 个主分区？

注：磁盘默认一个扇区大小为：512 字节。MBR 由以下 3 部分组成：

第一部分是：主引导程序 (boot loader) 占 446 个字节。主引导程序，它负责从活动分区中装载，并运行系统引导程序。

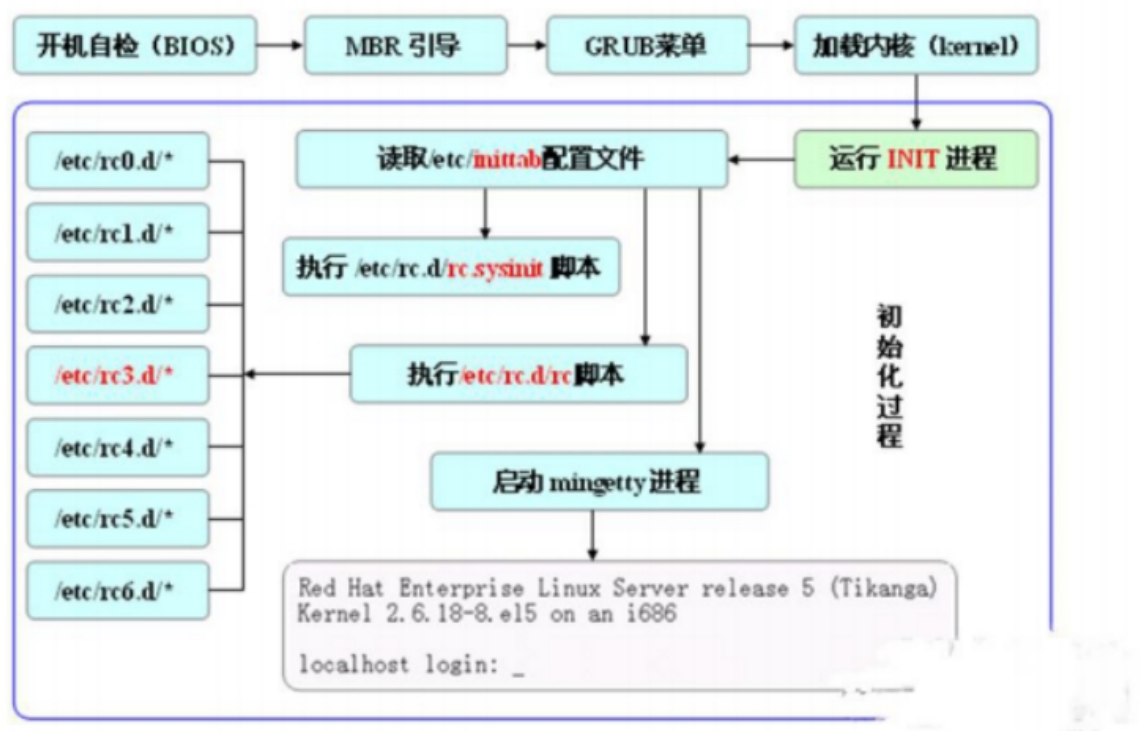
第二部分是 Partition table 区 (分区表)，即 DPT，占 64 个字节，硬盘中分区有多少以及每一分区的大小都记在其中。每个分区表项长 16 个字节， $16 \times 4 = 64$ 字节。为分区项 1、分区项 2、分区项 3、分区项 4。64 字节只存 4 个分区表。

第三部分是 MBR 有效标识位，占 2 个字节，固定为 55AA。如果这个标志位 0xAA55，就认为这个是 MBR。所以： $16 \times 4 + 446 + 2 = 512$

**3. 依据/boot/grub里 boot loader 的设置，到引导分区加载 Kernel ，Kernel 会开始侦测硬件并加载驱动程序；

4. 在硬件驱动成功后，Kernel 会主动执行 init 程序，而 init 会取得 run-level 信息；
5. init 执行 /etc/rc.d/rc.sysinit 文件来准备软件执行的作业环境 (如网络、时区等)；
6. init 执行 run-level 下各个服务并启动 (script 方式)；
7. init 执行开机后自动运行脚本 /etc/rc.d/rc.local 文件；
8. init 执行虚拟终端机控制程序 mingetty 来启动 login 程序，最后就等待用户登入啦； **

如图：



centos6 启动相关的配置文件

[root@xuegod64 Desktop]# vim /boot/grub/grub.conf

default=0 设定默认启动菜单项，当系统中有多个内核时，0 表示默认加载第 1 个，1 表示第 2 个内核

timeout=5 菜单项等待选项时间为 5s

splashimage=(hd0,0)/grub/splash.xpm.gz 指明菜单背景图片路径为

hiddenmenu 隐藏菜单

title CentOS (2.6.32-358.6.1.el6.x86_64) 定义菜单项

root (hd0,0) grub 查找 stage2 及 kernel 文件所在设备分区, grub 的根

kernel /vmlinuz-2.6.32-358.6.1.el6.x86_64 ro

root=/dev/vg_have/lv_root rd_NO_LUKS

LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 crashkernel=auto

rhgb quiet 启动的内核

initrd /initramfs-2.6.32-358.6.1.el6.x86_64.img 内核匹配的 ramfs 文件

修改系统启动级别:

```
[root@xuegod64 Desktop]# vim /etc/inittab
```

```
# Default runlevel. The runlevels used are:
```

```
# 0 - halt (Do NOT set initdefault to this)
```

```
# 1 - Single user mode
```

```
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
```

```
# 3 - Full multiuser mode
```

```
# 4 - unused
```

```
# 5 - X11
```

```
# 6 - reboot (Do NOT set initdefault to this)
```

```
#
```

```
id:3:initdefault: #这里决定系统启动的级别
```

/etc/rc.d/rc.sysinit shell 脚本 作用: 系统初始化: 像: 主机名和/etc/fstab 都在这里指定了, 完成了包括 mount 分区 激活swap 加载 modules 等重要的工作

启动对应级别下的服务如: init 3 级别

/etc/rc.d/rc3.d/ (这里的程序/服务 S 开头的全部开机执行; K 开头的表示开机不执行, 表明了关机时顺序)

rcn.d (n 为 1 到6) 是对应于不同的 runlevel 下起不同的服务. 这些目录下都是一些符号连接, 连接到

/etc/rc.d/init.d 下的一些文件. 以 S 开头的表示要启动, 以 K 开头的不启动 第一个字母后面的数值是一个优先级

```
[root@base ~]# ll /etc/rc.d/rc3.d/ | grep network
lrwxrwxrwx. 1 root root 17 Dec 18 2012 S10network ->
../init.d/network #表示 network 是第 10个启动的服务。所以init 是
顺序启动系统，需要一个一个服务启动成功，再执行下一步操作，启动系
统比较慢。而 centos7 中的 systemd 可以并行启动多个服务，启动比较
快。
```

例：

```
[root@base rc3.d]# vim /etc/init.d/network
#!/bin/bash
#
# network      Bring up/down networking
#
# chkconfig: 2345 10 90
看有chkconfig的那一行，2345 表示在 runlevel 2 3 4 5 下被启动，10 是为
此服务的启动顺序，90 为关机时，关闭此服务的顺序。
```

```
[root@base ~]# chkconfig --list | grep network
network      0:off    1:off    2:on 3:on 4:on 5:on 6:off
[root@base ~]# ll /etc/rc.d/rc3.d/ | grep network
lrwxrwxrwx. 1 root root 17 Dec 18 2012 S10network ->
../init.d/network #开机顺序
[root@base ~]# chkconfig network off
[root@base ~]# ll /etc/rc.d/rc3.d/ | grep network
lrwxrwxrwx 1 root root 17 May 23 21:17 K90network ->
../init.d/network #只显示 k90 关机顺序了
[root@xuegod64 rc3.d]# chkconfig --list network
network      0:off    1:off    2:off    3:off    4:off    5:off
6:off
```

所有服务都运行成功后，设置开机自动执行某个命令： /etc/rc.local

```
[root@xuegod64 rc3.d]# vim /etc/rc.local
[root@xuegod64 rc3.d]# ll !$
ll /etc/rc.local
```

```
lrwxrwxrwx. 1 root root 13 Dec 18 2012 /etc/rc.local ->
rc.d/rc.local
```

```
[root@xuegod64 rc3.d]# ll /etc/rc.d/rc.local
```

```
-rwxr-xr-x. 1 root root 240 Feb 5 21:17 /etc/rc.d/rc.local
```

运行 mingetty 命令, 打开 tty1-6

```
[root@xuegod64 rc3.d]# ps -axu | grep ming
```

Warning: bad syntax, perhaps a bogus '-'? See

/usr/share/doc/procps-3.2.8/FAQ

```
root 2346 0.0 0.0 4116 548 tty2 Ss+ 20:55 0:00 /sbin/mingetty
/dev/tty2
```

```
root 2348 0.0 0.0 4116 548 tty3 Ss+ 20:55 0:00 /sbin/mingetty
/dev/tty3
```

```
root 2350 0.0 0.0 4116 544 tty4 Ss+ 20:55 0:00 /sbin/mingetty
/dev/tty4
```

```
root 2352 0.0 0.0 4116 544 tty5 Ss+ 20:55 0:00 /sbin/mingetty
/dev/tty5
```

```
root 2354 0.0 0.0 4116 544 tty6 Ss+ 20:55 0:00 /sbin/mingetty
/dev/tty6
```

```
[root@base ~]# runlevel #查看系统启动级别
```

```
N 5
```

```
[root@base ~]# init 3
```

```
[root@base ~]# runlevel
```

```
5 3 #由 5 启动级别进入 3 级别
```

```
[root@base ~]# init 5
```

```
[root@base ~]# runlevel
```

```
3 5 #由 3 启动级别进入 5 级别
```