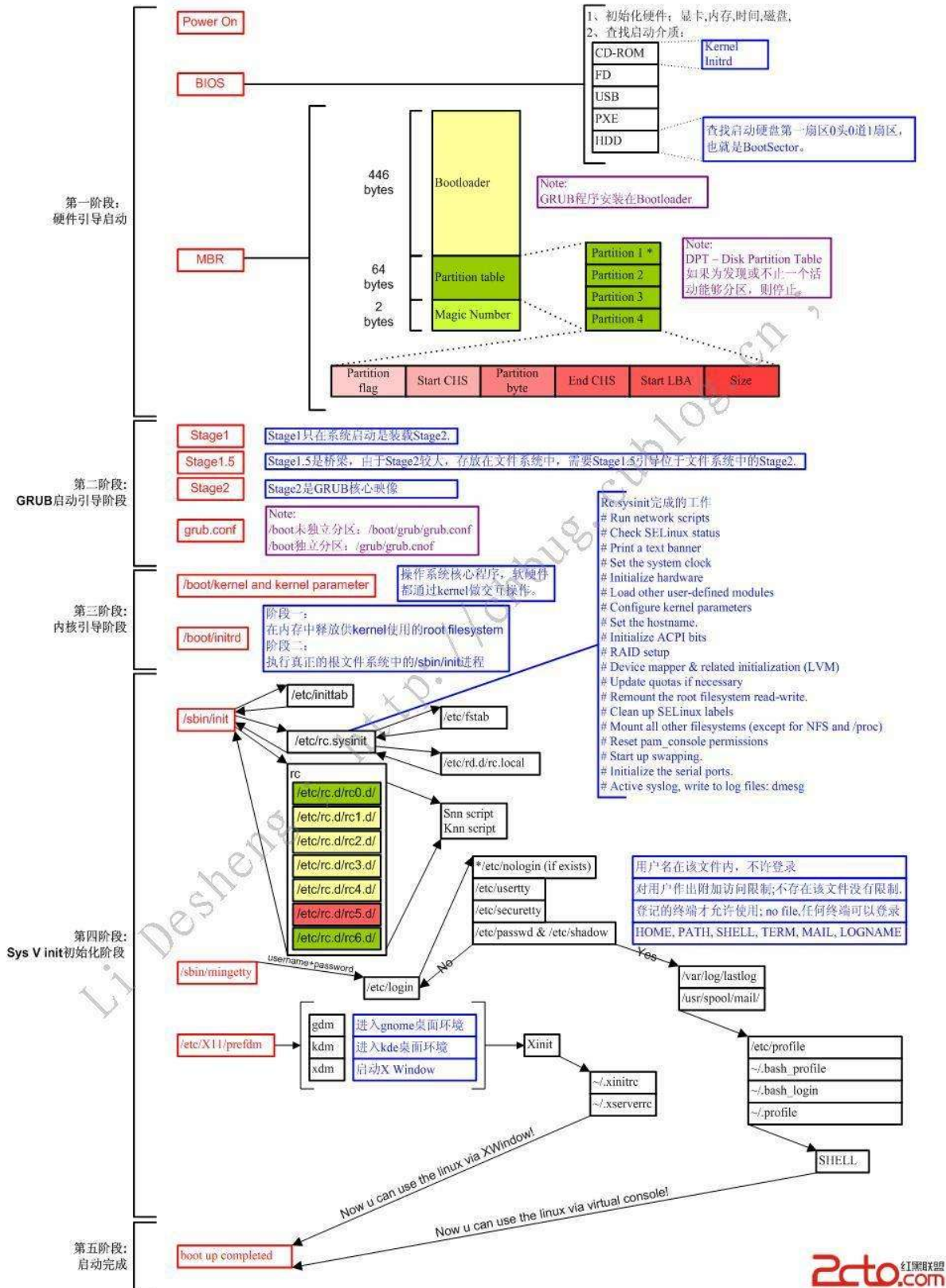


一.系统启动流程



(一) 启动流程

- 1、BIOS加电自检、检测硬件、决定启动介质、找到引导扇区
- 2、读取并且执行第一个开机设备中的MBR的bootloader(grub程序)
- 3、依据bootloader设置去加载内核, 内核开始侦测硬件并加载驱动程序

如果是sysvinit启动程序

- 4、内核启动init进程
- 5、系统初始化
- 6、init找到/etc/inittab文件，决定系统的运行级别(X)
- 7、触发相应的运行级别的事件，运行/etc/rcX.d下的脚本
- 8、执行/etc/rc.local文件，来到登录界面
- 9、输入用户名密码，进入到系统中

如果是systemd启动程序

- 4、在硬件驱动加载成功后，内核会主动呼叫systemd程序，并以default.target

流程开机

- systemd执行sysinit.target初始化系统及basic.target准备操作系统
- systemd启动multi-user.target下的本机与服务器服务
- systemd执行multi-user.target下的/etc/rc.d/rc.local文件
- systemd执行multi-user.target下的getty.target及登录服务
- systemd执行graphical需要的服务

boot loader主要功能如下：

- 提供选单：用户可以选择不同的开机项目
- 载入核心文件：直接指向可开机的程序区域来开始操作系统
- 转交其他loader：将开机管理功能转交给其他Loader负责

内核在Linux中的存放位置为/boot分区下，主要有2个文件要被加载：

vmlinuz-3.10.0-514.el7.x86_64 内核文件

initramfs-3.10.0-514.el7.x86_64.img 虚拟文件系统

查看initramfs文件里的内容：

```
[root@cloud002 boot]# lsinitrd initramfs-3.10.0-514.el7.x86_64.img
```

如果systemd默认的是multi-user.target，下面命令可以列出它加载的服务

```
[root@cloud002 system]# systemctl list-dependencies multi-user.target
```

```
[root@cloud002 system]# systemctl list-dependencies multi-user.target
multi-user.target
● auditd.service
● brandbot.path
● crond.service
● dbus.service
● irqbalance.service
● iscsid.service
● network.service
● NetworkManager.service
● plymouth-quit-wait.service
● plymouth-quit.service
● rsyslog.service
● sendmail.service
● sm-client.service
● sshd.service
● systemd-ask-password-wall.path
● systemd-logind.service
● systemd-readahead-collect.service
```

rhel7 grub配置文件及排错

RHEL7用了grub2，而不再是grub了

MBR分区的前446个字节存放的是系统引导程序grub，中间64字节是分区表，最后2个字节表示结束。那么什么是grub是怎么工作的呢？

简单的说，开机会经历以下几步：

- 1、BIOS自检，检查硬件；
- 2、激活MBR，MBR上不存在文件系统，可以视作硬件一部分，因此可以被直接读取
- 3、grub加载到内存，生成一个微系统，微系统内置了精简版的文件系统
- 4、通过这个微系统，他会去引导分区，比如默认一般是sda1上去找内核文件如vmlinuz,然后再调用grub的配置文件。

grub的主要把他的配置文件放在了3个地方。

/boot/grub2/grub.cfg （/etc/grub2.cfg是/boot/grub2/grub.cfg文件的符号链接）

/etc/grub.d/

/etc/default/grub

如下所示。他们的关系是 grub.cfg里面通过 #####BEGIN ##### 这种格式按照顺序调用/etc/grub.d里面的脚本实现不同的功能。grub.d目录里面有很多数字开头的脚本，按照从小到大的顺序执行。以00_header为例,他又会调用 /etc/default/grub 配置文件来实现最基本的开机界面配置

```
[root@server1 ~]# ls /boot/grub2/grub.cfg
/boot/grub2/grub.cfg
[root@server1 ~]# ls /etc/grub.d/
00_header 10_linux 20_linux_xen 20_ppc_terminfo 30_os-prober 40_custom 41_custom README
[root@server1 ~]# ls /etc/default/grub
/etc/default/grub
```

例如：在/etc/grub2.cfg文件里面调用 /etc/grub.d/10_linux 来配置不同的内核，这里面有2个 menuentry (菜单入口)，所以我们开机的时候会看见两个默认选项，一个是普通模式，一个是救援模式

```
### BEGIN /etc/grub.d/10_linux ###
menuentry 'Red Hat Enterprise Linux Server, with Linux 3.10.0-123.el7.x86_64' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-fecc9c90-09ae-48fa-a8c9-2999d981fdeb' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x${feature_platform_search_hint} = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 8903fe24-7c0d-42bd-bac1-5cc35773de2b
    else
        search --no-floppy --fs-uuid --set=root 8903fe24-7c0d-42bd-bac1-5cc35773de2b
    fi
    linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=fecc9c90-09ae-48fa-a8c9-2999d981fdeb ro rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap vconsole.font=latarcyrheb-sun16 vconsole.keymap=us rhgb quiet vga=0x340
    initrd16 /initramfs-3.10.0-123.el7.x86_64.img
}
menuentry 'Red Hat Enterprise Linux Server, with Linux 0-rescue-60e839870acd49b68d569dcbb05142e2' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-60e839870acd49b68d569dcbb05142e2-advanced-fecc9c90-09ae-48fa-a8c9-2999d981fdeb' {
    load_video
```

下面是/etc/default/grub 文件。和其他的脚本比较起来，非常简单直观了。后面会举例如何修改

需要注意的是，最好不要直接去修改 /etc/grub2.cfg 文件。这是因为如果后期升级内核，所有的配置都会失效。如果需要自定义这个文件，我们可以修改对应的脚本或者 /etc/default/grub文件，然后通过 grub2-mkconfig 重新生成grub.cfg文件。

例1：修改启动的等待时间

rhel7默认启动等待时间为5秒，下面将启动时间修改为3秒（注：如果改为-1，那么每次启动时需手动确认才可以）

修改/etc/default/grub文件，如下图所示：

```
[root@server1 ~]# cat /etc/default/grub
GRUB_TIMEOUT=3
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap vconsole.font=latarcyrheb-sun16 vconsole.keymap=us rhgb quiet net.ifnames=0 biosdevname=0"
GRUB_DISABLE_RECOVERY="true"
```

修改之后重新编译生成grub.cfg文件

```
[root@server1 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-60e839870acd49b68d569dcbb05142e2
Found initrd image: /boot/initramfs-0-rescue-60e839870acd49b68d569dcbb05142e2.img
[ 90.010104] end_request: I/O error, dev fd0, sector 0
done
```

修改网卡的显示名字，这个前面做网络配置的时候提到过，这里不赘述了。

加密grub：

开机界面的时候如果输入e,会打开编辑窗口，我们可以根据需要进入rescue, emergency 或者 shell 模式。如何限制访问。

在/etc/grub.d/00_header 文件末尾，添加以下内容

```
cat <<EOF
set superusers='admin'
password admin pwd123
EOF
```

重新编译生成grub.cfg文件

```
[root@server1 ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-60e839870acd49b68d569dcbb05142e2
Found initrd image: /boot/initramfs-0-rescue-60e839870acd49b68d569dcbb05142e2.img
[ 1324.785564] end_request: I/O error, dev fd0, sector 0
done
```

重启之后输入e，就需要用户和密码才能进入编辑窗口了

以上设置的是明文密码，那如何设置密文口令呢？

加密密码由命令grub2-mkpasswd-pbkdf2生成

```
[root@server1 ~]# grub2-mkpasswd-pbkdf2
```

在/etc/grub.d/00_header 文件末尾，添加以下内容

```
cat <<EOF
set superusers=' 用户名'
password_pbkdf2 用户名加密密码
EOF
```

如下图所示：

```
cat <<EOF
set superusers='admin'
password_pbkdf2 admin grub.pbkdf2.sha512.10000.DAD17D288EEB7797B7B3323F9D9629DA8722D226A8FA713722ADE72E3530CBBC6E0ED5FC770F1
1696EADA72C885E.61575A3992C247B9345D539AF3609BF6981AFB1A9E2803107400875535A5C15CD65A0557AC97CCA12C35562CE7DEF755E85004BFF880
EOF
```

重新编译生成grub.cfg文件

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

重启验证

注：在文本终端模式下，安装并启用gpm后就可复制粘贴了。

```
[root@server1 Packages]# rpm -vih gpm-1.20.7-5.el7.x86_64.rpm
warning: gpm-1.20.7-5.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID fd431d51: NOKEY
Preparing...                                [100%]
Updating / installing...
 1:gpm-1.20.7-5.el7                          [100%]
[root@server1 Packages]# systemctl start gpm.service
```

在这个编辑窗口，我们可以根据需要进入 rescue, emergency和 shell引导的3种模式。这3种模式对于系统启动排错很有帮助，比如某个服务卡住了无法加载我们可以通过这3种模式来排错。

rescue模式: 在commandline的配置末尾添加s，类似rhel6之前的单用户模式


```
linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=fecc9c90-09ae-48fa-a8\
c9-2999d981fdeb ro rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap v\
console.font=latarcyrheb-sun16 vconsole.keymap=us rhgb quiet net.ifnames=0 bio\
sdevname=0 s
initrd16 /initramfs-3.10.0-123.el7.x86_64.img
```

按ctrl+x启动

普通模式需要加载的服务很多，但是这个rescue模式加载的就少很多了，输入管理员密码就可以进入了。

```
[ 4.502194] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 4.503231] sd 0:0:1:0: [sdb] Assuming drive cache: write through
[ 4.507168] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 4.514837] sd 0:0:1:0: [sdb] Assuming drive cache: write through
[ 4.534218] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 4.542697] sd 0:0:1:0: [sdb] Assuming drive cache: write through
[ 11.551221] end_request: I/O error, dev fd0, sector 0
[ 11.574228] end_request: I/O error, dev fd0, sector 0
[ 12.955836] piix4_smbus 0000:00:07.3: Host SMBus controller not enabled!
Welcome to rescue mode! Type "systemctl default" or ^D to enter default mode.
Type "journalctl -xb" to view system logs. Type "systemctl reboot" to reboot.
Give root password for maintenance
(or type Control-D to continue): _
```

emergency 模式和rescue模式类似，不过加载的服务更少，把s改成 emergency就行了
那么如果忘记root密码了怎么办？

方法1：

- 1、重启系统，按e
- 2、在linux16那行的最尾部加上一下信息

rd.break console=tty0

```
linux16 /boot/vmlinuz-3.10.0-123.el7.x86_64 root=UUID=9bf6b9f7-92ad-44\
1b-848e-0257cbb883d1 ro vconsole.keymap=us console=tty0 console=ttyS0,115200n8\
crashkernel=auto vconsole.font=latarcyrheb-sun16 LANG=en_US.UTF-8 rd.break c\
onsole=tty0
```

3、按ctrl+x启动

4、重新挂在文件系统

```
switch_root:/# mount -o remount,rw /sysroot/
```

5、改变根目录

```
switch_root:/# chroot /sysroot/
sh-4.2#
```

6、修改root密码

```
sh-4.2# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
sh-4.2#
```

7、在根目录下创建相关文件（用于重新标记selinux环境值）

```
sh-4.2# touch /.autorelabel
```

8、退出，系统将会重启

```
sh-4.2# exit
exit
switch_root:/# exit
logout
```

方法2:

在linux16 的最后一行删除 rhgb quiet, 然后添加 init=/bin/sh, 他会用shell替代默认的daemon进程

```
linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=UUID=fecc9c90-09ae-48fa-a8\
c9-2999d981fdeb ro rd.lvm.lv=rhel/root crashkernel=auto rd.lvm.lv=rhel/swap v\
console.font=latarcyrheb-sun16 vconsole.keymap=us init=/bin/sh net.ifnames=0 b\
iosdevname=0
initrd16 /initramfs-3.10.0-123.el7.x86_64.img
```

按ctrl+x启动

挂载文件系统为可写模式，原因很简单，根目录加载的权限是ro，只读，重新加载成 rw的权限。

```
sh-4.2# mount -o remount,rw /
```

这个时候就可以修改密码了，执行passwd，按提示修改root密码

```
sh-4.2# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

如果之前系统启用了selinux，必须运行以下命令，否则将无法启动系统：

```
sh-4.2# touch /.autorelabel  
sh-4.2# exec /sbin/init
```

创建这个文件会自动在开启的时候重新做标签即用于重新标记 SELINUX 环境值
运行命令exec /sbin/init来正常启动，或者用命令exec /sbin/reboot重启

开机排错

例1. grub故障：比如把MBR的前446个字节都覆盖了。

```
[root@srv1 ~]# dd if=/dev/zero of=/dev/sda bs=1 count=446  
446+0 records in  
446+0 records out  
446 bytes (446 B) copied, 0.000937765 s, 476 kB/s
```

系统重启后则出现如下故障现象

开机的时候会自动尝试从本地，光盘和网络加载引导程序（如果光盘、网络等引导失败则会operatiing system not found）；这里是从我的光盘加载的，因为本地的引导程序已经没有了。

```
Red Hat Enterprise Linux 7.0  
  
Install Red Hat Enterprise Linux 7.0  
Test this media & install Red Hat Enterprise Linux 7.0  
  
Troubleshooting >  
  
Press Tab for full configuration options on menu items.
```

选择troubleshooting（排错）

```
Install Red Hat Enterprise Linux 7.0  
Test this media & install Red Hat Enterprise Linux 7.0  
  
Troubleshooting >  
  
Press Tab for full configuration options on menu items.
```

选择resure a red hat enterprise linux system

Troubleshooting

Install Red Hat Enterprise Linux 7.0 in basic graphics mod
Rescue a Red Hat Enterprise Linux system
Run a memory test

Boot from local drive

Return to main menu

<

Press Tab for full configuration options on menu items.

If the system will not boot, this lets you access files
and edit config files to try to get it booting again.

选择continue, 注意有耐心多等一下, 有的时候会卡个10秒才有反应
按照提示, 切换根到本机的系统

```
sh-4.2# df
Filesystem                1k-blocks    Used Available Use% Mounted on
/dev/mapper/live-rw        2030899   982932   1043871   49% /
devtmpfs                   999676      0    999676    0% /dev
tmpfs                      1017824      0   1017824    0% /dev/shm
tmpfs                      1017824   16812   1001012    2% /run
tmpfs                      1017824      0   1017824    0% /sys/fs/cgroup
/dev/sr0                    3654720 3654720      0 100% /run/install/repo
tmpfs                      1017824     324   1017500    1% /tmp
/dev/mapper/rhel-root     52403200 3398420 49004780    7% /mnt/sysimage
/dev/sda1                   508588   121312   387276   24% /mnt/sysimage/boot
devtmpfs                   999676      0    999676    0% /mnt/sysimage/dev
tmpfs                      1017824      0   1017824    0% /mnt/sysimage/dev/shm
tmpfs                      1017824      0   1017824    0% /dev/shm
/dev/mapper/rhel-home     49774852 32928 49741924    1% /mnt/sysimage/home
tmpfs                      1017824   16812   1001012    2% /mnt/sysimage/run
sh-4.2# chroot /mnt/sysimage/
```

fdisk -l 看看启动分区是哪个, 有星号的就是

Disk identifier: 0x00000000

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|---------|-----------|-----------|----|-----------|
| /dev/sda1 | * | 2048 | 1026047 | 512000 | 83 | Linux |
| /dev/sda2 | | 1026048 | 209715199 | 104344576 | 8e | Linux LVM |

重装一下 grub2 到启动分区就可以了

```
bash-4.2# grub2-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
bash-4.2#
```

重启就OK了

例2：如果grub引导程序没问题，但是我们把内核文件或者grub.cfg配置文件给删除了怎么办

```
[root@srv1 ~]# ls /boot/
config-3.10.0-123.el7.x86_64
grub2
initramfs-0-rescue-400d570488314998976fa2ddb122170d.img
initramfs-3.10.0-123.el7.x86_64.img
initramfs-3.10.0-123.el7.x86_64kdump.img
initrd-plymouth.img
symvers-3.10.0-123.el7.x86_64.gz
System.map-3.10.0-123.el7.x86_64
vmlinuz-0-rescue-400d570488314998976fa2ddb122170d
vmlinuz-3.10.0-123.el7.x86_64
[root@srv1 ~]# rm -fr /boot/*
```

重启之后，直接进入grub的救援模式，这表明引导程序木有问题，但是引导文件找不到了

```
error: file '/grub2/i386-pc/normal.mod' not found.
Entering rescue mode...
grub rescue>
```

和前面一样，光盘启动，切换根目录，挂载一下光盘到光盘挂载点，顺便看看/boot目录，里面空荡荡的，内核文件和配置文件都没了

```
sh-4.2# chroot /mnt/sysimage/
bash-4.2# ls /boot/
bash-4.2# mount /dev/cdrom /media/
mount: /dev/sr0 is write-protected, mounting read-only
```

解决方法就是重新安装内核，可以用rpm 或者yum，用yum需要先卸载当前的，再安装；rpm可以强行--force覆盖安装。

```
bash-4.2# rpm -vih /media/Packages/kernel-3.10.0-123.el7.x86_64.rpm --force
warning: /media/Packages/kernel-3.10.0-123.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID fd
431d51: NOKEY
Preparing...                               [100%]
Updating / installing...
 1:kernel-3.10.0-123.el7                     [100%]
```

装完内核之后，重新安装一次 grub2，输入 grub2-install /dev/sda，然后重新编译一下 grub2就行了

```
bash-4.2# grub2-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
bash-4.2# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-123.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-123.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-400d570488314998976fa2ddb122170d
Found initrd image: /boot/initramfs-0-rescue-400d570488314998976fa2ddb122170d.img
done
```

重启系统就可以了。

重启之后，如果遇见以下问题：

问题是开机直接进入命令行模式，即使runlevel显示的是5，手动输入startx一样报错。

解决方式是重新安装图形化界面

yum grouplist

yum groupinstall 'Server with GUI'

(二) .修复MBR:

1.备份文件的方式: dd if=/dev/sda of=/loring/sda.mbr.bak bs=512
count=1

2.模拟扇区故障:

dd if=/dev/zero of=/dev/sda bs=512 count=1

3.恢复

dd if=tempdir/sda.mbr.bak of=/dev/sda bs=512 count=1

实验:

mkdir loring

mount /dev/sdb1 /loring

dd if=/dev/sda of=/loring/mbr.bak bs=512 count=1

dd if=/dev/zero of=/dev/sda bs=512 count=1

打开电源进入到bios里面, boot里面设定光盘启动

进入到修复模式

/mnt/sysimage(挂载点)

mkdir /backup

mount /dev/sdb1 /backup

dd if=/backup/mbr.bak of=/dev/sda bs=512 count=1

(三) .grup故障修复

MBR中的grub坏掉了

grub.conf文件损坏或者丢失

(系统引导不了, 停住, 显示grup>提示符)

1.手动引导

2.进入救援模式, 修复grup

配置文件: /boot/grub/grub.conf

cd /boot/grub mv grub.conf文件

1.root (hd0,0)

kernel (/vmlinuz-2.6.32... ro root=UUID#)

initrd /initra...

boot

进入到救援模式, 默认会找到根分区, 并挂载到/mnt/sysimage

chroot /mnt/sysimage

修复grub

1、root密码

安装系统时候给root设置密码

破解root密码：

前提：能重启系统

进入单用户模式

2、grub密码

为了防止用户进入单用户模式修改root密码，怎么办呢？

给grub加上密码

1) 明文密码 —— 修改grub配置文件

```
# vim /boot/grub/grub.conf
```

在timeout下面添加一行

```
-----  
password 123456  
-----
```

```
# reboot
```

重启之后，想要编辑grub进入单用户的话，需要提供grub的密码

按p：输入密码

必须正确输入grub的密码，然后才能编辑grub

2) 密文密码

(1) 生成加密后的字符串

```
# grub-md5-crypt
```

Password:

Retype password:

```
$1$r5htQ$IRUJY./85zdZT3a6JDmfv0 //加密后的字符串
```

串

(2) 修改grub配置文件

```
# vim /boot/grub/grub.conf
```

```
password --md5 $1$r5htQ$IRUJY./85zdZT3a6JDmfv0 //
```

修改password那行

3、破解grub密码

前提条件：手头要有光盘

需要进入救援模式

想要进入到救援模式，需要进入bios修改引导介质的优先级，让

cdrom优先

虚拟机进入BIOS：按F2 真机进入BIOS：一般按(. del键)

将光盘移动到硬盘上面，按F10保存

选择救援已安装系统 Rescue

可以执行chroot /mnt/sysimage可以切换到真正的系统的根目录

修改grub配置文件，将password那行删除掉

bash-4.1# reboot

重启之后又可以进到单用户模式了。

4、BIOS密码

为了防止用户进入救援模式，给BIOS加密码

重启系统，按F2想要进入BIOS时候，

小总结：

- 1、如果破解root密码，需要进入单用户模式
- 2、防止进入单用户模式，给grub加密码
- 3、破解grub密码，需要进入救援模式
- 4、防止进入救援模式，给BIOS加密码

要想进入单用户模式的前提，需要能够重启系统

要想进入救援模式，要有光盘，而且能够进入BIOS修改引导介质的顺序