

# CentOS7之Systemd管理

本章节内容:

CentOS7启动

Unit介绍

服务管理和查看

启动排错

破解口令

修复grub2

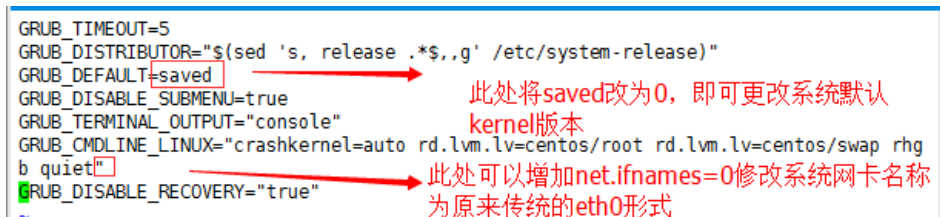
1、CentOS启动流程: POST → Boot Sequence → Bootloader → kernel + initramfs(initrd) → rootfs → /sbin/init

UEFI或BIOS初始化, 运行POST开机自检

选择启动设备

引导装载程序, centos7是grub2

加载装载程序的配置文件: /etc/grub.d/ /etc/default/grub /boot/grub2/grub.cfg (注: 一般上修改grub.cfg配置文件, 是直接修改/etc/default/grub, 然后使用命令#grub2-mkconfig /boot/grub2/grub.cfg来生成修改)



```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

blob.png

加载initramfs驱动模块

加载内核选项

内核初始化, centos7使用systemd代替init

执行initrd.target所有单元, 包括挂载/etc/fstab

从initramfs根文件系统切换到磁盘根目录

systemd执行默认target配置, 配置文件/etc/systemd/default.target /etc/systemd/system/

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

systemd执行multi-user.target下的getty.target及登入服务

systemd执行graphical需要的服务 (此为图形界面所有)

2、Systemd: 系统启动和服务器守护进程管理器, 负责在系统启动或运行时, 激活系统资源, 服务器进程和其它进程

3、Systemd新特性:

系统引导时实现服务并行启动

按需启动守护进程 (端口与服务独立分离)

自动化的服务依赖关系管理

同时采用socket式与D-Bus总线式激活服务

系统状态快照

4、Unit介绍: unit表示不同类型的systemd对象, 通过配置文件进行标识和配置; 文件中主要包含了系统服务、监听socket、保存的系统快照以及其它与init相关的信息

配置文件: /usr/lib/systemd/system每个服务最主要的启动脚本设置, 类似于之前的/etc/init.d/

/run/systemd/system: 系统执行过程中所产生的服务脚本, 比上面目录优先运行

/etc/systemd/system: 管理员建立的执行脚本, 类似于/etc/rc.d/rcN.d/Sxx类的功能, 比上面目录优先运行

5、Unit类型:

systemctl -t help 查看unit类型

```
[root@node1 ~]# systemctl -t help
Available unit types:
service
socket
busname
target
snapshot
device
mount
automount
swap
timer
path
slice
scope
```

blob.png

service unit: 文件扩展名为.service, 用于定义系统服务

target unit: 文件扩展名为.target, 用于模拟实现“运行级别”

device unit: .device, 用于定义内核识别的设备

mount unit: .mount, 定义文件系统挂载点

socket unit: .socket, 用于标识进程间通信用的socket文件, 也可在系统启动时, 延迟启动服务, 实现按需启动

snapshot unit: .snapshot, 管理系统快照

swap unit: .swap, 用于标识swap设备

automount unit: .automount, 文件系统的自动挂载点

path unit: .path, 用于定义文件系统中的文件或目录使用, 常用于当文件系统变化时, 延迟激活服务, 如: spool 目录

## 6、Systemd的特性:

关键特性:

基于socket的激活机制: socket与服务程序分离

基于d-bus的激活机制:

基于device的激活机制:

基于path的激活机制:

系统快照: 保存各unit的当前状态信息于持久存储设备中

向后兼容sysvinit脚本

不兼容:

systemctl命令固定不变, 不可扩展

非由systemd启动的服务, systemctl无法与之通信和控制

## 7、Systemd管理服务:

管理系统服务: CentOS 7: service unit

注意: 能兼容早期的服务脚本

命令: systemctl COMMAND name.service

启动: service name start ==> systemctl start name.service

停止: service name stop ==> systemctl stop name.service

重启: service name restart ==> systemctl restart name.service

状态: service name status ==> systemctl status name.service

```
[root@node1 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2016-09-20 13:53:14 CST; 35min ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 1515 (httpd)
    Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"
    CGroup: /system.slice/httpd.service
            └─1515 /usr/sbin/httpd -DFOREGROUND
              └─2657 /usr/sbin/httpd -DFOREGROUND
                └─2658 /usr/sbin/httpd -DFOREGROUND
                  └─2659 /usr/sbin/httpd -DFOREGROUND
                    └─2660 /usr/sbin/httpd -DFOREGROUND
                      └─2662 /usr/sbin/httpd -DFOREGROUND

Sep 20 13:53:11 node1.chesfer.org systemd[1]: Starting The Apache HTTP Server...
Sep 20 13:53:14 node1.chesfer.org httpd[1515]: AH00557: httpd: apr_sockaddr_info...g
Sep 20 13:53:14 node1.chesfer.org httpd[1515]: AH00558: httpd: Could not reliabl...e
Sep 20 13:53:14 node1.chesfer.org systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

blob.png

条件式重启：已启动才重启，否则不做操作

service name condrestart=> systemctl try-restart name.service

重载或重启服务：先加载，再启动

systemctl reload-or-restart name.service

重载或条件式重启服务：systemctl reload-or-try-restart name.service

禁止自动和手动启动：systemctl mask name.service

```
[root@node1 ~]# systemctl mask httpd.service
Created symlink from /etc/systemd/system/httpd.service to /dev/null.
```

blob.png

取消禁止：systemctl unmask name.service

```
[root@node1 ~]# systemctl unmask httpd.service
Removed symlink /etc/systemd/system/httpd.service.
[root@node1 ~]#
```

blob.png

查看某服务当前激活与否的状态：systemctl is-active name.service

```
[root@node1 ~]# systemctl is-active httpd.service
active
[root@node1 ~]#
```

blob.png

查看所有的服务：systemctl list-unit-files -type=t service

```
[root@node1 ~]# systemctl list-unit-files -t service
UNIT FILE                                STATE
arp-ethers.service                      disabled
auditd.service                          enabled
autovt@.service                         disabled
blk-availability.service                disabled
brandbot.service                        static
console-getty.service                  disabled
console-shell.service                  disabled
container-getty@.service                static
```

blob.png

查看所有激活的服务：systemctl list-units -type service -all -a

```
[root@node1 ~]# systemctl list-units -t service -a
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
auditd.service                     loaded active running Security Auditing Service
brandbot.service                   loaded inactive dead Flexible Branding Service
cpupower.service                   loaded inactive dead Configure CPU power relate
crond.service                       loaded active running Command Scheduler
dbus.service                       loaded active running D-Bus System Message Bus
● display-manager.service          not-found inactive dead display-manager.service
dm-event.service                   loaded inactive dead Device-mapper event daemon
dracut-shutdown.service             loaded inactive dead Restore /run/initramfs
ebtables.service                   loaded inactive dead Ethernet Bridge Filtering
emergency.service                  loaded inactive dead Emergency Shell
● exim.service                     not-found inactive dead exim.service
firewalld.service                  loaded active running firewalld - dynamic firewa
getty@tty1.service                 loaded active running Getty on tty1
httpd.service                       loaded active running The Apache HTTP Server
● ip6tables.service                not-found inactive dead ip6tables.service
● iptables.service                 not-found inactive dead iptables.service
irqbalance.service                 loaded active running irqbalance daemon
● kdump.service                    loaded failed failed Crash recovery kernel arm
```

blob.png

设定某服务开机自启(chkconfig命令的对应关系): `chkconfig name on ==> systemctl enable name.service`

设定某服务开机禁止启动: `chkconfig name off ==> systemctl disable name.service`

```
[root@node1 ~]# systemctl disable httpd.service
Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.
[root@node1 ~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@node1 ~]#
```

服务设置开机启动或者禁用，实际上就是创建一个软链接

blob.png

查看所有服务的开机自启状态:

`chkconfig -list ==> systemctl list-unit-files --type=service`

用来列出该服务在哪些运行级别下启用和禁用

`chkconfig sshd -list ==> ls /etc/systemd/system/*.wants/sshd.service`

查看服务是否开机自启: `systemctl is-enabled name.service`

其它命令: 查看服务的依赖关系: `systemctl list-dependencies name.service`

```
[root@node1 ~]# systemctl list-dependencies httpd.service
httpd.service
● └─ .mount
● └─ system.slice
● └─ tmp.mount
● └─ basic.target
● └─ firewalld.service
```

blob.png

杀掉进程: `systemctl kill 进程名`

Systemctl命令事例:

显示所有单元状态: `systemctl` 或 `systemctl list-units`

只显示服务单元的状态: `systemctl --type=service`

显示sshd服务单元: `systemctl status sshd.service -l`

验证sshd服务当前是否活动: `systemctl is-active sshd`

启动, 停止和重启sshd服务

`systemctl start sshd.service`

`systemctl stop sshd.service`

`systemctl restart sshd.service`

重新加载配置: `systemctl reload sshd.service`

列出活动状态的所有服务单元: `systemctl list-units --type=service`

列出所有服务单元: `systemctl list-units --type=service --all`

查看服务单元的启用和禁用状态: `systemctl list-unit-files --type=service`

列出失败的服务: `systemctl --failed --type=service`

列出依赖的单元: `systemctl list-dependencies sshd`

验证sshd服务是否开机启动: `systemctl is-enabled sshd`

禁用network, 使之不能自动启动, 但手动可以: `systemctl disable network`

启用network: `systemctl enable network`

禁用network, 使之不能手动或自动启动: systemctl mask network

启用network: systemctl unmask network

## 7、服务状态:

systemctl list-units --type service --all显示状态

loaded:Unit配置文件已处理

active(running):一次或多次持续处理的运行

active(exited):成功完成一次性的配置

active(waiting):运行中, 等待一个事件

inactive:不运行

enabled:开机启动

disabled:开机不启动

static:开机不启动, 但可被另一个启用的服务激活

## 8、运行级别: target units

unit配置文件: .target

ls /usr/lib/systemd/system/\*.target

systemctl list-unit-files --type target --all

### 运行级别:

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

blob.png

级别切换: init N ==> systemctl isolate name.target

eg: systemctl isolate multi-user.target

注: 只有/lib/systemd/system/multi-user.target文件中AllowIsolate=yes才能切换(修改文件需执行systemctl daemon-reload才能生效)

```
[root@node1 ~]# cat /usr/lib/systemd/system/multi-user.target
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.

[Unit]
Description=Multi-User System
Documentation=man:systemd.special(7)
Requires=basic.target
Conflicts=rescue.service rescue.target
After=basic.target rescue.service rescue.target
AllowIsolate=yes
[root@node1 ~]#
```

blob.png

查看target: runlevel 或 who -r

systemctl list-units --type target

```
[root@node1 ~]# runlevel
N 3
[root@node1 ~]# who -r
run-level 3 2016-09-20 13:53
[root@node1 ~]#
```

blob.png

获取默认运行级别: /etc/inittab ==> systemctl get-default

```
[root@node1 ~]# systemctl get-default
multi-user.target
[root@node1 ~]#
```

blob.png

修改默认级别: /etc/inittab==> systemctl set-default name.target

systemctl set-default multi-user.target

ls -l /etc/systemd/system/default.target

```
[root@node1 ~]# ls -l /etc/systemd/system/default.target
lrwxrwxrwx. 1 root root 37 Jul 31 11:08 /etc/systemd/system/default.target -> /lib/s
ystemd/system/multi-user.target
[root@node1 ~]#
```

blob.png

9、其他命令:

切换至紧急救援模式: systemctl rescue

切换至emergency模式: systemctl emergency (注: 当切换到紧急救援模式失败, 可以尝试切换到该模式救援)

其它常用命令:

传统命令init, poweroff, halt, reboot都成为systemctl的软链接:

关机: systemctl halt、systemctl power off

重启: systemctl reboot

挂起: systemctl suspend

休眠: systemctl hibernate

休眠并挂起: systemctl hybrid-sleep

10、service units 文件格式:

/etc/systemd/system: 系统管理员和用户使用

/usr/lib/systemd/system: 发行版打包者使用

以“#”开头的行后面的内容会被认为是注释

相关布尔值, 1、yes、on、true 都是开启, 0、no、off、false 都是关闭。

时间单位默认是秒, 所以要用毫秒(ms) 分钟(m) 等请显式说明

service unit file文件通常由三部分组成

[Unit]: 定义与Unit类型无关的通用选项; 用于提供unit的描述信息、unit行为及依赖关系等

[Service]: 与特定类型相关的专用选项; 此处为Service类型

[Install]: 定义由“systemctl enable”以及“systemctl disable”命令在实现服务启用或禁用时用到的一些选项

```
[root@node1 ~]# cat /usr/lib/systemd/system/httpd.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=man:httpd(8)
Documentation=man:apachectl(8)

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/httpd
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
ExecStop=/bin/kill -WINCH ${MAINPID}
# We want systemd to give httpd some time to finish gracefully, but still want
# it to kill httpd after TimeoutStopSec if something went wrong during the
# graceful stop. Normally, Systemd sends SIGTERM signal right after the
# ExecStop, which would kill httpd. We are sending useless SIGCONT here to give
# httpd time to finish.
KillSignal=SIGCONT
PrivateTmp=true

[Install]
WantedBy=multi-user.target
[root@node1 ~]#
```

blob.png

(1) Unit段的常用选项:

Description: 描述信息

After: 定义unit的启动次序, 表示当前unit应该晚于哪些unit启动, 其功能与Before相反

Requires: 依赖到的其它units, 强依赖, 被依赖的units无法激活时, 当前unit即无法激活

Wants: 依赖到的其它units, 弱依赖

Conflicts: 定义units间的冲突关系

(2) Service段的常用选项: Type (定义影响ExecStart及相关参数的功能的unit进程启动类型)

simple: 默认值, 这个daemon主要由ExecStart接的指令串来启动, 启动后常驻于内存中

forking: 由ExecStart启动的程序透过spawns延伸出其他子程序来作为此daemon的主要服务。原生父程序在启动结束后就会终止

oneshot: 与simple类似, 不过这个程序在工作完毕后就结束了, 不会常驻在内存中

dbus: 与simple类似, 但这个daemon必须要在取得一个D-Bus的名称后, 才会继续运作。因此通常也要同时设定BusName=才行

notify: 在启动完成后会发送一个通知消息。还需要配合NotifyAccess 来让Systemd 接收消息

idle: 与simple类似, 要执行这个daemon必须要所有的工作都顺利执行完毕后才执行。这类的daemon通常是开机到最后才执行即可的服务

EnvironmentFile: 环境配置文件

ExecStart: 指明启动unit要运行命令或脚本的绝对路径

ExecStartPre: ExecStart前运行

ExecStartPost: ExecStart后运行

ExecStop: 指明停止unit要运行的命令或脚本

Restart: 当设定Restart=1 时, 则当次daemon服务意外终止后, 会再次自动启动此服务

(3) Install段的常用选项:

Alias: 别名, 可使用systemctl command Alias.service

RequiredBy: 被哪些units所依赖, 强依赖

WantedBy: 被哪些units所依赖, 弱依赖

Also: 安装本服务的时候还要安装别的相关服务

注意: 对于新创建的unit文件, 或者修改了的unit文件, 要通知systemd重载此配置文件, 而后可以选择重启

# systemctl daemon-reload

事例: vim /etc/systemd/system/bak.service

[Unit]

Description=backup my etc

Requires=atd.service

[Service]

Type=simple

ExecStart=/bin/bash -c "echo /testdir/bak.sh|at now"

[Install]

WantedBy=multi-user.target

#systemctl daemon-reload

#systemctl start bak

11、设置内核参数, 只影响当次启动

启动时, 在linux16行后添加systemd.unit=desired.target

systemd.unit=emergency.target

systemd.unit=recure.target

recure.target 比emergency 支持更多的功能, 例如日志等

12、启动排错:

文件系统损坏: 先尝试自动修复, 失败则进入emergency shell, 提示用户修复

在/etc/fstab不存在对应的设备和UUID, 一段时间, 如不可用, 进入emergency shell

在/etc/fstab不存在对应挂载点, systemd尝试创建挂载点, 否则提示进入emergency shell

在/etc/fstab不正确的挂载选项, 提示进入emergency shell

13、破解root密码:

启动时任意键暂停启动

按e键进入编辑模式



将光标移动到linux16开始的行，添加内核参数rd.break

按ctrl-x启动

```
#mount -o remount,rw /sysroot
#chroot /sysroot
#passwd root
#touch /.autorelabel
```

#exit

#exit

14、修改grub2:

GRUB“the Grand Unified Bootloader”

引导提示时可以使用命令行界面

可从文件系统引导

主要配置文件/boot/grub2/grub.cfg

修复配置文件: grub2-mkconfig > /boot/grub2/grub.cfg (grub2-mkconfig -o /boot/grub2/grub.cfg)

修复grub: grub2-install /dev/sda BIOS环境

grub2-install UEFI环境

习题:

1、为编译安装的httpd服务，实现service unit文件

编译安装httpd: #./configure --prefix=/usr/local/httpd --sysconfdir=/etc/httpd

```
[root@node1 httpd-2.4.9]# ./configure --prefix=/usr/local/httpd --sysconfdir=/etc/httpd
```

blob.png

编写httpd的服务Unit:

```
[root@node1 bin]# cd /etc/systemd/system
[root@node1 system]# ls
basic.target.wants      getty.target.wants
dbus-org.fedoraproject.FirewallD1.service  multi-user.target.wants
dbus-org.freedesktop.NetworkManager.service sockets.target.wants
dbus-org.freedesktop.nm-dispatcher.service  sysinit.target.wants
default.target          system-update.target.wants
default.target.wants
[root@node1 system]# vim httpd.service
[root@node1 system]# cat httpd.service
[Unit]
Description= HTTP server
[Service]
Type=simple
ExecStart=/usr/local/httpd/bin/httpd
[Install]
WantedBy=multi-user.target
[root@node1 system]#
```

blob.png

```
[root@node1 ~]# systemctl daemon-reload
[root@node1 ~]# systemctl restart httpd
[root@node1 ~]# systemctl status httpd
● httpd.service - HTTP server
   Loaded: loaded (/etc/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2016-09-20 21:59:07 CST; 8s ago
     Main PID: 19664 (httpd)
        CGroup: /system.slice/httpd.service
                └─19664 /usr/local/httpd/bin/httpd

Sep 20 21:59:07 node1.chesfer.org systemd[1]: Started HTTP server.
Sep 20 21:59:07 node1.chesfer.org systemd[1]: Starting HTTP server...
[root@node1 ~]#
```

blob.png

2、破解centos7 口令



```
CentOS Linux (3.10.0-327.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-93a7ba72498a4a9485fa326b12bd3b9d) 7 (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

blob.png

在linux16该行的后面输入rd.break完成后，使用Ctrl+x启动系统

```
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 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 8c17c644-1\
95a-469c-948f-d740cf4f9bc5
else
  search --no-floppy --fs-uuid --set=root 8c17c644-195a-469c-948f-d740\
cf4f9bc5
fi
linux16 /vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/centos-root ro\
crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet rd.br\
eak_
initrd16 /initramfs-3.10.0-327.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

blob.png

下图为进入破解的命令行操作界面：

```
[ 3.017806] sd 2:0:0:0: [sda] Assuming drive cache: write through

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/# _
```

blob.png

执行破解的步骤：

```
[ 3.017806] sd 2:0:0:0: [sd] Assuming drive cache: write through

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/# mount -o remount,rw /sysroot/
switch_root:/# chroot /sysroot/
sh-4.2# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
sh-4.2# touch /.autorelabel
sh-4.2# exit
exit
switch_root:/# exit
```

blob.png

### 3、修改默认的启动内核

[illegible]

blob.png

```
CentOS Linux (3.10.0-327.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-93a7ba72498a4a9485fa326b12bd3b9d) 7 (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

blob.png

```
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 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 8c17c644-1\
95a-469c-948f-d740cf4f9bc5
else
    search --no-floppy --fs-uuid --set=root 8c17c644-195a-469c-948f-d740\
cf4f9bc5
fi
linux16 /vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/centos-root ro\
crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet selin\
ux=0_
initrd16 /initramfs-3.10.0-327.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

blob.png

## 5、启动时进入emergency模式

```
CentOS Linux (3.10.0-327.el7.x86_64) 7 (Core)
CentOS Linux (0-rescue-93a7ba72498a4a9485fa326b12bd3b9d) 7 (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

blob.png

```
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 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 8c17c644-1\
95a-469c-948f-d740cf4f9bc5
else
    search --no-floppy --fs-uuid --set=root 8c17c644-195a-469c-948f-d740\
cf4f9bc5
fi
linux16 /vmlinuz-3.10.0-327.el7.x86_64 root=/dev/mapper/centos-root ro\
crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet syste\
md.unit=emergency.target
initrd16 /initramfs-3.10.0-327.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

blob.png

输入root的密码后得到下图的界面:

```
[ 3.007746] sd 2:0:0:0: [sda] Assuming drive cache: write through
Welcome to emergency mode! After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or ^D to
try again to boot into default mode.
Give root password for maintenance
(or type Control-D to continue):
[root@node1 ~]#
```

blob.png

## 6、删除编译安装的新内核

```
#rm -rf /boot/*.3.18.41*
```

```
#vim /boot/grub2/grub.cfg 删除相应的kernel信息
```

```
#rm -rf /lib/modules/3.18.41*
```

```
#rm -rf /usr/src/linux-3.18.41 （一般建议在编译安装完成后，将相应的源码文件删除掉）
```