查看systemd管理的所有单元

systemctl list-unit-files

启动服务

systemctl start httpd.service

关闭服务

systemctl stop httpd.service

重启服务

systemctl restart httpd.service

重新加载

systemctl reload httpd.service

查看状态

systemctl status httpd.service

包括启动状态、启动时间、主进程及相关进程、相关日志

centos7之前的服务启动：

先会启动init进程，然后

1，/etc/rc.d/rc.sysinit

激活udev和selinux；

设置定义在/etc/sysctl.conf中的内核参数；

设置系统时钟；

加载keymaps；

激活交换分区；

设置主机名(hostname)；

根分区检查和remount；

激活RAID和LVM设备；

开启磁盘配额；

检查并挂载所有文件系统；

清除过期的locks和PID文件；

2，/etc/rc.d/rc和/etc/rc.d/rcX.d/(X代表运行级别0-6)

完成了以上这些工作之后，SysV init开始运行/etc/rc.d/rc脚本。根据不同的runlevel，rc脚本将打开对应runlevel的rcX.d目录(X就是runlevel)，

找到并运行存放在该目录下的所有启动脚本。每个runlevelX都有一个这样的目录，目录名为/etc/rc.d/rcX.d。

在这些目录下存放着很多不同的脚本。文件名以S开头的脚本就是启动时应该运行的脚本，S后面跟的数字定义了这些脚本的执行顺序。

在/etc/rc.d/rcX.d目录下的脚本其实都是一些软链接文件，真实的脚本文件存放在/etc/init.d目录下

/etc/rc.d/rc.local

大多连接到/etc/init.d中

3 SysV init和系统关闭

SysV init不仅需要负责初始化系统，还需要负责关闭系统。在系统关闭时，为了保证数据的一致性，需要小心地按顺序进行结束和清理工作。

比如应该先停止对文件系统有读写操作的服务，然后再umount文件系统。否则数据就会丢失。

这种顺序的控制这也是依靠/etc/rc.d/rcX.d/目录下所有脚本的命名规则来控制的，在该目录下所有以K开头的脚本都将在关闭系统时调用，字母K之后的数字定义了它们的执行顺序。

这些脚本负责安全地停止服务或者其他的关闭工作。

unbentu的的还有有upstart,把启动的服务分组，加快启动速度；

更多关于init的管理参照：http://www.linuxidc.com/Linux/2015-04/115937.htm

centos7的启动：

一般会用iprinit,而我们的存储中直接回调用systemd作为第一个启动的进程，应该是同一个东西的不同名字：

然后systemctl get-default 看默认是什么运行级别，如果为multi-user.target，就

会按照/etc/systemd/system/multi-user.target.wants中的脚本（是软链接）进行开机启动，

然后另外一些不需要开机启动的服务都放在/usr/systemd/system/ 及/usr/systemd/user中/

至于runlevel：

修改默认运行级别

1.首先删除已经存在的符号链接

rm /etc/systemd/system/default.target

2.默认级别转换为3(文本模式)

systemctl enable multi-user.target

相当于ln -s /lib/systemd/system/multi-user.target /etc/systemd/system/default.target

3.重启

reboot

运行级别如下：

runlevel0.target -> poweroff.target

runlevel1.target -> rescue.target

runlevel2.target -> multi-user.target

runlevel3.target -> multi-user.target

runlevel4.target -> multi-user.target

runlevel5.target -> graphical.target

runlevel6.target -> reboot.target

不知道一个工具属于哪个包，可以用下面这个查询

yum whatprovides netstat

一、加电自检

二、选择启动设备

1.读取启动设备第一个扇区，读取引导程序

2.引导程序读取配置文件/boot/grub2/grub.cfg #不要编辑此文件

上面文件是grub2-mkconfig命令（用这两个文件/etc/default/grub /etc/grub.d）生成的

3.加载内核，并且以只读方式加载根分区

4.加载init ram disk

三、加载systemd进程

1.读取/etc/fstab

2.读取所选择的target,如multi-user.target

3.启动该级别的服务

4./etc/rc.d/rc.local开机脚本 (现在应该是在/usr/lib/systemd/下)

四、login

pvscan;vgscan;lvscan

lvchange -a y /dev/rhel/home

mknod /dev/rhel/home b 253 1

xfs\_repare /dev/rhel/home

systemd.unit=emergency.target

运行级别：

graphical.target 相当于原来的level 5

multi-user.target 相当于原来的level 3

rescue.target 相当于原来的level 1 单用户模式，不启动服务。但不同的是，现在需要密码才能进入。

powerof.target 相当于原来的level 0

reboot.target 相当于原来的level 6

emergency.target 救援，文件系统故障

rd.break

以前可以定义某些服务只在3级别自动启动，而5级别不启动

现在graphical.target 调用multi-user.target，所以没法这么做了。

systemctl get-default显示当前target

systemctl isolate multi-user.targe相当于以前的init 3命令

systemctl list-dependencies graphical.target |　grep target

systemctl list-units --type=targe --all

systemctl list-unit-files --type=targe --all

systemctl isolate multi-user.targe

ddif=/dev/zero of=/dev/sda bs=446 count=1

grub2-install /dev/sda

grub2-mkpasswd-pbkdf2 加密

samba客户端多用户

echo 'username=brain' >/root/smb-multiuser.txt

echo 'password=redhat' >/root/smb-multiuser.txt

vim /etc/fstab

//serverX/smbshare /mnt/multiuser cifs credentials=root/smb-multiuser.txt,multiuser,sec=ntlmssp 0

mount /mnt/multiuser

su -brain