

03 【基础篇-系统管理】

1.Linux 中的进程和服务

计算机中，一个正在执行的程序或命令，被叫做“进程”（process）。

启动之后一直存在、常驻内存的进程，一般被称作“服务”（service）。

详细操作后面说明

2.systemctl 服务管理

service 服务管理（CentOS 6 版本-了解）

systemctl （CentOS 7 版本-重点掌握）

CentOS 7使用Systemd管理守护进程。centos7采用 systemd管理，服务独立的运行在内存中，服务响应速度快，但占用更多内存。独立服务的服务启动脚本都在目录 /usr/lib/systemd/system里。Systemd的新特性：

- 系统引导时实现服务的并行启动；
- 按需激活进程；
- 系统实现快照；
- 基于依赖关系定义服务的控制逻辑；

systemctl可用于内省和控制“systemd”系统和服务管理器的状态。centos7.x系统环境下我们经常使用此命令启停服务，实际上此命令除了其他独立服务还有很多其他用途。

2.1 systemctl参数说明

基本语法：`systemctl start | stop | restart | status | reload 服务名`

`systemctl` 指令管理的服务在 `/usr/lib/systemd/system`

查看查看服务的方法：`pwd /usr/lib/systemd/system`

1、使用语法

用法：`systemctl [OPTIONS...] {COMMAND} ...`

2、参数说明

参数	参数说明
start	立刻启动后面接的unit
stop	立刻关闭后面接的unit
restart	立刻关闭后启动后面接的unit，亦即执行stop再start的意思
reload	不关闭后面接的unit的情况下，重载配置文件，让设定生效
enable	设定下次开机时，后面接的unit会被启动
disable	设定下次开机时，后面接的unit 不会被启动
status	目前后面接的这个unit 的状态，会列出是否正在执行、是否开机启动等信息。
is-active	目前有没有正在运行中
is-enable	开机时有没有预设要启用这个unit
kill	不要被kill这个名字吓着了,它其实是向运行unit的进程发送信号
show	列出unit的配置。
mask	注销unit,注销后你就无法启动这个unit了
unmask	取消对unit的注销
list-units	依据unit列出目前有启动的unit。若加上-all才会列出没启动的。（等价于无参数）
list-unit-files	列出所有以安装unit以及他们的开机启动状态（enabled、disabled、static、mask）。
-type=TYPE	就是unit type，主要有service，socket，target等
get-default	取得目前的 target
set-default	设定后面接的 target 成为默认的操作模式
isolate	切换到后面接的模式

3、unit file结构

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"及"systemctl disable"命令在实现服务启用或禁用时用到的一些选项。

4、Unit段的常用选项

- Description: 描述信息，意义性描述；
- After: 定义unit的启动次序；表示当前unit应晚于哪些unit启动；其功能与Before相反；
- Requires: 依赖到其它的units；强依赖，被依赖的units无法激活时，当前的unit即无法激活；
- Wants: 依赖到其它的units；弱依赖；

- Conflicts: 定义units 的冲突关系;

5、Service段的常用选项

- Type: 用于定义影响ExecStart及相关参数的功能的unit进程类型;
类型有: simple、forking、oneshot、dbus、notify、idle。
- EnvironmentFile: 环境配置文件;
- ExecStart: 指明启动unit要运行的命令或脚本; ExecStart, ExecStartPost
- ExecStop: 指明停止unit要运行的命令或脚本;
- Restart:

6、Install段的常用配置:

- Alias:
- RequiredBy: 被哪些unit所依赖;
- WantBy: 被哪些unit所依赖;

7、Unit文件样例

```
[root@s153 system]# cat chronyd.service
[Unit]
Description=NTP client/server
Documentation=man:chronyd(8) man:chrony.conf(5)
After=ntpdate.service sntp.service ntpd.service
Conflicts=ntpd.service systemd-timesyncd.service
ConditionCapability=CAP_SYS_TIME

[Service]
Type=forking
PIDFile=/var/run/chronyd.pid
EnvironmentFile=/etc/sysconfig/chronyd
ExecStart=/usr/sbin/chronyd $OPTIONS
ExecStartPost=/usr/libexec/chrony-helper update-daemon
PrivateTmp=yes
ProtectHome=yes
ProtectSystem=full

[Install]
WantedBy=multi-user.target
```

2.2 systemctl使用示例

1.查看开机启动列表

```
systemctl list-unit-files [ | grep 服务名] (查看服务开机启动状态, grep 可以进行过滤)
[root@localhost ~]# systemctl list-unit-files
[root@localhost ~]# systemctl list-unit-files | grep firewalld
firewalld.service                                disabled

#查看已启动的服务列表
systemctl list-unit-files|grep enabled
#
显示所有已启动的服务
systemctl list-units --type=service
```

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl list-unit-files
UNIT FILE                                STATE
proc-sys-fs-binfmt_misc.automount      static
dev-hugepages.mount                    static
dev-mqueue.mount                       static
proc-sys-fs-binfmt_misc.mount          static
sys-fs-fuse-connections.mount          static
sys-kernel-config.mount                static
sys-kernel-debug.mount                 static
tmp.mount                              disabled
brandbot.path                          disabled
systemd-ask-password-console.path      static
systemd-ask-password-plymouth.path     static
systemd-ask-password-wall.path         static
session-17271.scope                    static
session-18774.scope                    static
session-21836.scope                    static
session-3569.scope                     static
session-4984.scope                     static
session-6301.scope                     static
```

可以写一半再查看完整的服务名，一般也可以简写：`firewalld.service = firewall`

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl list-unit-files | grep firewall
firewalld.service                                enabled
[root@izwz91bne18alc4g6ixb37z ~]#
```

说明防火墙是一个自启的状态，Linux系统启动的时候防火墙也会自启。

2. 设置开机启动

`systemctl`在enable、disable、mask子命令里面增加了`--now`选项，可以激活同时启动服务，激活同时停止服务等。

```
# 设置开机启动并现在启动
## 相当于同时执行了systemctl start 服务名
systemctl enable --now firewalld

# 查看服务启动状态
root@localhost ~]# systemctl status firewalld
```

3. 取消开机启动

```
# 取消开机启动并现在就停止服务
systemctl disable --now firewalld
## 查看服务状态是否停止
[root@localhost ~]# systemctl status firewalld

# 查看启动列表
[root@localhost ~]# systemctl list-unit-files |grep firewalld
firewalld.service                                disabled
```

使用 `systemctl disable firewalld` 时，下次重启系统时防火墙还是处于关闭的状态

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
[root@izwz91bne18alc4g6ixb37z ~]# systemctl is-enabled firewalld
disabled
[root@izwz91bne18alc4g6ixb37z ~]#
```

重新打开自启动防火墙:

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl enable firewalld
Created symlink from /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service to /usr/lib/systemd/system/firewalld.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/firewalld.service to /usr/lib/systemd/system/firewalld.service.
[root@izwz91bne18alc4g6ixb37z ~]# systemctl is-enabled firewalld
enabled
[root@izwz91bne18alc4g6ixb37z ~]#
```

1. `systemctl enable` 服务名 (设置服务开机启动), 对 3 (无界面) 和 5 (GUI) 运行级别都生效
2. `systemctl disable` 服务名 (关闭服务开机启动), 对 3 (无界面) 和 5 (GUI) 运行级别都生效

4. 开启服务

```
systemctl start firewall
```

开启防火墙:

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl start firewalld
[root@izwz91bne18alc4g6ixb37z ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-05-25 16:10:45 CST; 1s ago
     Docs: man:firewalld(1)
    Main PID: 31485 (firewalld)
      Memory: 26.7M
      CGroup: /system.slice/firewalld.service
              └─31485 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

May 25 16:10:44 izwz91bne18alc4g6ixb37z systemd[1]: Starting firewalld - dynamic firewall daemon...
May 25 16:10:45 izwz91bne18alc4g6ixb37z systemd[1]: Started firewalld - dynamic firewall daemon.
[root@izwz91bne18alc4g6ixb37z ~]#
```

5. 关闭服务(但是下次开机还是会启动)

```
systemctl stop firewall
```

关闭防火墙:

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl stop firewalld
[root@izwz91bne18alc4g6ixb37z ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Tue 2021-05-25 16:10:03 CST; 1s ago
     Docs: man:firewalld(1)
    Main PID: 492 (code=exited, status=0/SUCCESS)

May 25 16:10:02 izwz91bne18alc4g6ixb37z systemd[1]: Stopping firewalld - dynamic firewall daemon...
May 25 16:10:03 izwz91bne18alc4g6ixb37z systemd[1]: Stopped firewalld - dynamic firewall daemon.
Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
[root@izwz91bne18alc4g6ixb37z ~]#
```

6. 重启服务

```
systemctl restart 服务名
```

7. 重新加载配置

```
systemctl reload 服务名
```

8.输出服务运行的状态

```
systemctl status 服务名  
systemctl status firewalld
```

查看防火墙的状态，现在是运行中：

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl status firewalld  
● firewalld.service - firewalld - dynamic firewall daemon  
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)  
   Active: active (running) since Sun 2021-01-17 11:20:41 CST; 4 months 6 days ago  
     Docs: man:firewalld(1)  
   Main PID: 492 (firewalld)  
    Memory: 1.2M  
    CGroup: /system.slice/firewalld.service  
            └─492 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid  
  
Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.  
[root@izwz91bne18alc4g6ixb37z ~]#
```

9.检查service是否在启动状态

写脚本是判断服务器是否启动很管用

```
# systemctl is-active 服务名  
systemctl is-active NetworkManager  
# active
```

10.检测unit单元是否为自动启动

写脚本时判断服务器是否开机自启很管用

```
# systemctl is-enabled 服务名  
systemctl is-enabled firewalld  
# enabled
```

```
[root@izwz91bne18alc4g6ixb37z ~]# systemctl is-enabled firewalld  
enabled  
[root@izwz91bne18alc4g6ixb37z ~]#  
  
[root@izwz91bne18alc4g6ixb37z ~]# systemctl is-enabled sshd  
enabled  
[root@izwz91bne18alc4g6ixb37z ~]# systemctl is-enabled sshd.service  
enabled  
[root@izwz91bne18alc4g6ixb37z ~]#
```

11.注销一个服务(service)

systemctl mask 是注销服务的意思。

注销服务意味着：

该服务在系统重启的时候不会启动

该服务无法进行做systemctl start/stop操作

该服务无法进行systemctl enable/disable操作


```
systemctl mask firewalld
```

12.取消注销服务(service)

```
systemctl unmask firewalld
```

13.显示单元的手册页（前提是由unit提供）

```
systemctl help
```

14.当新增或修改service单元文件时，需要系统重新加载所有修改过的配置文件

```
systemctl daemon-reload
```

15.查看systemd资源使用率

```
systemd-cgtop
```

16.杀死服务

```
[root@sl53 system]# systemctl kill xinetd  
[root@sl53 system]# systemctl is-failed xinetd  
inactive
```

3.系统运行级别

1) Linux 运行级别 CentOS 6

➤ 运行级别(runlevel)

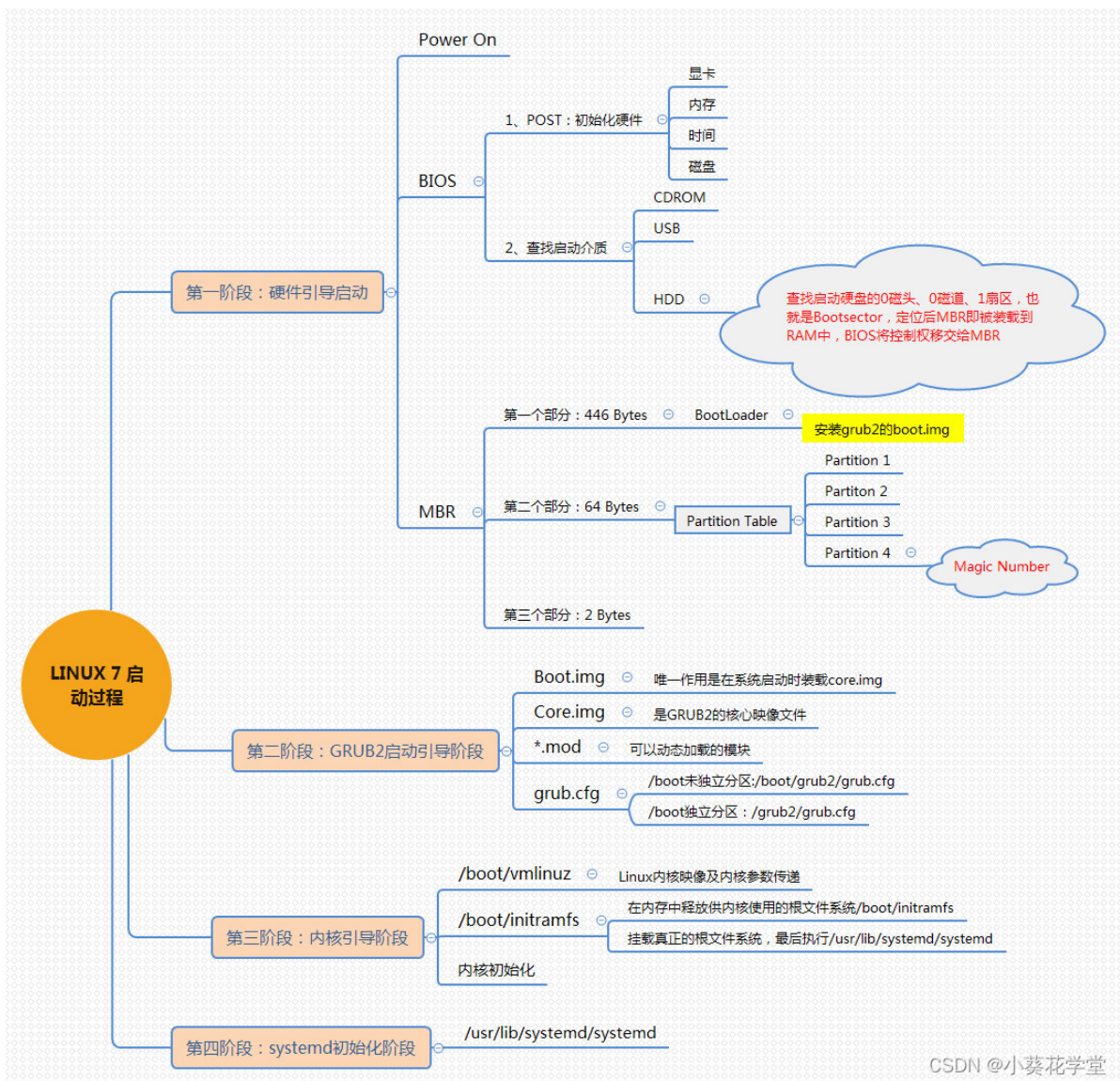


查看默认级别： `vi /etc/inittab`

Linux系统有7种运行级别(runlevel)：**常用的是级别3和5**

- **运行级别0：**系统停机状态，系统默认运行级别不能设为0，否则不能正常启动
- **运行级别1：**单用户工作状态，root权限，用于系统维护，禁止远程登陆
- **运行级别2：**多用户状态(没有NFS)，不支持网络
- **运行级别3：**完全的多用户状态(有NFS)，登陆后进入控制台命令行模式
- **运行级别4：**系统未使用，保留
- **运行级别5：**X11控制台，登陆后进入图形GUI模式
- **运行级别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

Centos7的启动流程图



CentOS7中我们的初始化进程变为了systemd。执行默认target配置文件/etc/systemd/system/default.target（这是一个软链接，与默认运行级别有关）。然后执行sysinit.target来初始化系统和basic.target来准备操作系统。接着启动multi-user.target下的本机与服务器服务，并检查/etc/rc.d/rc.local文件是否有用户自定义脚本需要启动。最后执行multi-user下的getty.target及登录服务，检查default.target是否有其他的服务需要启动。

注意：/etc/systemd/system/default.target指向了/lib/systemd/system/目录下的graphical.target或multiuser.target。而graphical.target依赖multiuser.target，multiuser.target依赖basic.target，basic.target依赖sysinit.target，所以倒过来执行。

2) CentOS7 的运行级别简化为:

- multi-user.target 等价于原运行级别 3（多用户有网，无图形界面）
- graphical.target 等价于原运行级别 5（多用户有网，有图形界面）

3) 查看当前运行级别:

```
[root@localhost etc]# systemctl get-default
multi-user.target
```

4) 修改当前运行级别

```
[root@localhost etc]# systemctl set-default graphical.target
```



```
[root@localhost etc]# cat /etc/inittab
# inittab is no longer used when using systemd.
#
# ADDING CONFIGURATION HERE WILL HAVE NO EFFECT ON YOUR SYSTEM.
#
# Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target
#
# systemd uses 'targets' instead of runlevels. By default, there are two
main targets:
#
# multi-user.target: analogous to runlevel 3      #类似运行级别3
# graphical.target: analogous to runlevel 5      #类似运行级别5
#
# To view current default target, run:
# systemctl get-default                          #查看系统运行级别
#
# To set a default target, run:
# systemctl set-default TARGET.target            #修改系统默认运行级别
```

4.关机重启命令

4.1 关机重启命令汇总

halt	关机	root 用户	halt: 只关闭系统, 电源还在运行 halt -p: 关闭系统, 关闭电源 (先执行halt, 再执行poweroff)
poweroff	关机	root 用户	poweroff会发送一个关闭电源的信号给acpi
reboot	重启	root 用户	
shutdown	-h: 关机 -r: 重启 -c: 取消shutdown操作	root 用户	shutdown实际上是调用init 0, init 0会cleanup一些工作然后调用halt或者poweroff shutdown -r now: 一分钟后重启 shutdown -r 05:30: 最近的5:30 重启 shutdown -r +10: 十分钟后重启
init	init 0: 关机 init 6: 重启	root 用户	init: 切换系统的运行级别

halt	关机	root 用户	halt: 只关闭系统, 电源还在运行 halt -p: 关闭系统, 关闭电源 (先执行halt, 再执行poweroff)
systemctl	systemctl halt [-i]: 关机 systemctl poweroff [-i]: 关机 systemctl reboot [-i]: 重启	普通 用户 超级 用户	普通用户需要加-i root用户不需要加-i

- (1) sync (功能描述: 将数据由内存同步到硬盘中)
- (2) halt (功能描述: 停机, 关闭系统, 但不断电)
- (3) poweroff (功能描述: 关机, 断电) (
- (4) reboot (功能描述: 就是重启, 等同于 shutdown -r now)

在关机或者重启之前, 执行3至4次sync, 将在内存中还未保存到硬盘的数据更新到硬盘中, 否则会造成数据的丢失。执行sync时要以管理员的身份运行, 因为管理员具有所有文件的权限, 而普通用户只具有自己的部分文件的权限。

最经常使用的关机重启的命令是shutdown, 因此下面详细学习的使用。

4.2 shutdown命令

基本格式: shutdown [选项] [时间] [警告信息]

选项:

- 1. -h: 关机
- 2. -r: 重启
- 3. -c: 取消shutdown执行的关机或者重启命令
- 4. -k: 不关机, 发出警告

时间:

- 1. shutdown: 一分钟后关机 (默认)
- 2. shutdown now: 立刻关机
- 3. shutdown 10: 10分钟后关机
- 4. shutdown 05:00: 5点关机

示例:

shutdown -r now: 系统立马重启 (等同于 reboot

shutdown -r 05:30: 最近的5:30重启

shutdown -r 10: 十分钟后重启

shutdown -h now: 立马关机 (等同于 poweroff

shutdown -h 05:30: 最近的5:30关机

shutdown -h +10: 十分钟后关机

shutdown -c: 取消上面的关机重启操作

shutdown -k +10 "I will shutdown in 10 minutes": 10分钟后并不会真的关机, 但是会把警告信息发给所有的用户。

4.3 sync命令

sync：linux同步数据命令，**将数据由内存同步到硬盘中**，包含已修改的 i-node、已延迟的块 I/O 和读写映射文件。如果不去手动的输入sync命令来真正的去写磁盘，linux系统也会周期性的去sync数据。

```
[root@hadoop100 桌面]#sync
```

使用场景：

- 1.在 关机或者开机之前最好多执行这个几次，以确保数据写入硬盘。
- 2.挂载时，需要很长时间的的操作动作（比如，cp 大文件，检测文件），在这个动作之后接sync。
- 3.卸载U盘或其他存储设备，需要很长时间，使用sync。

经验技巧

Linux 系统中为了提高磁盘的读写效率，对磁盘采取了“预读迟写”操作方式。当用户 保存文件时，Linux 核心并不一定立即将保存数据写入物理磁盘中，而是将数据保存在缓冲区中，等缓冲区满时再写入磁盘，这种方式可以极大的提高磁盘写入数据的效率。但是，也带来了安全隐患，如果数据还未写入磁盘时，系统掉电或者其他严重问题出现，则将导致数据丢失。使用 sync 指令可以立即将缓冲区的数据写入磁盘。