

systemctl

1. 简介

Systemd 是 Linux 系统中最新的初始化系统（init），它主要的设计目标是克服 sysvinit 固有的缺点，提高系统的启动速度。Systemd 的很多概念来源于苹果 Mac OS 操作系统上的 launchd。

- 作为 init 系统，systemd 进程作为系统中的 1 号进程应该是毋庸置疑的

2. 概念、原理

- systemd 是解决了 sysvinit 服务失控的缺点；systemd 使用了 cgroup 技术，可以有效地关闭关联服务，避免服务失控

当进程创建子进程时，子进程会继承父进程的 CGroup。因此无论服务如何启动新的子进程，所有的这些相关进程都会属于同一个 CGroup，systemd 只需要简单地遍历指定的 CGroup 即可正确地找到所有的相关进程

- Systemd 提供自动挂载服务，实现类似 autofs 的；

2.1 单元 - unit

系统初始化需要做的事情非常多。需要启动后台服务，比如启动 SSHD 服务；需要做配置工作，比如挂载文件系统。这个过程每一步都被 systemd 抽象为一个配置单元，即 unit。可以认为一个服务是一个配置单元；一个挂载点是一个配置单元；一个交换分区的配置是一个配置单元；等等

2.1.1 单元类型

- **service unit** : 文件扩展名为 **.service**，用于定义系统类服务
- **target unit** : 文件扩展为 **.target**，用于实现模拟“运行级别”
- **device unit** : 文件扩展为 **.device**，用于定义内核识别的设备。
- **mount unit** : 文件扩展为 **.mount**，定义文件系统挂载点，利用 logind 服务，为用户的会话进程分配 CGroup 资源
- **socket unit** : 文件扩展为 **.socket**，用于标识进程间通信的 socket 文件
- **snapshot unit** : 文件扩展为 **.snapshot**，管理系统快照

- `swap unit` : 文件扩展为 `.swap` ,用于标识swap设备
- `automount unit` : 文件扩展为 `.automount` ,文件系统自动挂载设备
- `path unit` : 文件扩展为 `.path` , 用于定义文件系统中的 一个文件或目录
- `timer unit` : 文件扩展为 `.timer` ,定时器配置单元，用来定时触发用户定义的操作，这类配置单元取代了atd、crond等传统的定时服务

2.1.2 Target和运行级别的对应关系

sysvinit	systemd target	备注
0	poweroff.target	关闭系统
1,s, single	rescue.target	单用户模式
2,4	multi-user.target	用户定义/域特定运行级别。默认等同于 3
3	multi-user.target	多用户，非图形化界面
5	graphical.target	多用户，图形化界面
6	reboot.target	重启
emergency	emergency.target	紧急shell

`systemctl list-units --type=target --all` 命令可以显示所有target， 下面介绍几种常见的target:

- **rescue.target** - 在无法使用 root 登陆的情况下，systemd 在开机时会多加一个额外的临时系统，与你原本的系统无关。这时你可以取得 root 的权限来维护你的系统。
- **shutdown.target** - 就是执行关机。
- **graphical.target** - 就是文字界面再加上图形界面，这个 target 已经包含了下面的 multi-user.target
- **multi-user.target** - 纯文本模式！
- **getty.target**- 可以设置 tty 的配置，如果要修改可提供登陆的 tty 数量，则修改 getty.target

2.2 并行启动原理

- **linux特性** - 当进程调用 fork 或者 exec 创建子进程之后，所有在父进程中被打开的文件句柄

(file descriptor) 都被子进程所继承；

3. 命令

3.1 电源管理命令

命令	操作
systemctl reboot	重启机器
systemctl poweroff	关机
systemctl suspend	挂起
systemctl hibernate	休眠
systemctl hybrid	混合休眠模式（快照并挂起）

3.2 添加配置文件

- **daemon-reload** 命令会做很多的事情，其中之一是重新生成依赖树(也就是 **unit** 之间的依赖关系)，所以当你修改了 **unit** 配置文件中的依赖关系后如果不执行 **daemon-reload** 命令是不会生效的
 - `systemctl daemon-reload`

3.3 unit启动状态

- **enable/disable** - 是否开机自启
- **static** - **unit** 不可以自己启动，不过可能会被其它的 **enabled** 的服务来唤醒
- **mask** - **unit** 无论如何都无法被启动！因为已经被强制注销。可通过 **systemctl unmask** 改回原来的状态

3.4 通用命令

- `systemctl list-dependencies firewalld.service --reverse` - 查看哪些unit依赖了firewalld服务
- `systemctl list-dependencies firewalld.service` - 查看firewalld依赖了哪些unit

4. 配置文件

4.1 systemd的相关配置文件路径

- /usr/lib/systemd/system
- /run/systemd/system
- /etc/systemd/system

参考博客

- [Linux之systemd的工作原理及使用](#)