Lec17 认识系统服务（daemons）

in Unix-Like, "daemon"! where? what? how? ports?

after CentOS 7.x, traditional init ✗ systemd ✓

什么是 daemon 与 service？

service："常驻在内存中的程序，且可以提供一些系统或网络功能，称为服务"

"系统为了某些功能必须要提供一些服务（不论是系统本身还是网络方面）"

daemon："service 的提供总需要程序的运作"，"达成这个 service 的程序就称为 daemon"

"启动 XX daemon 来提供 XX service"

例：达成例行性工作排程服务的程序为 crond 这个 daemon。

一般来说，daemon ≈ service

早期的 init（optional）

系统核心第一支呼叫的程序是 init，然后 init 去唤起所有的系统所需要的服务（本地服务＋网络服务）。

现在的 systemd

advantages：

- 平行地处理所有服务，加速开机流程：旧的 init 启动脚本是一项一项地依序启动的。（硬件升级）
- 一经要求就响应的 on-demand 启动方式：只需一支 systemd 服务搭配 systemctl 指令（常驻内存），无需其他。
- 服务相依性的自我检查：自动。
- 依 daemon 的功能分类：systemd 将 unit 分为 service, socket, target, timer 等不同 type。
- 将多个 daemons 集合为一个群组：target，执行某个 target 就是执行好多个 daemon 的意思。
- 向下兼容旧有的 init 服务脚本

不过 systemd 也是有些地方无法完全取代 init 的。

基本上，systemd 将过去所谓的 daemon 执行脚本通通称为一个 unit，而每个 unit 依据其功能区分为不同的 type，配置文件被放在：/etc/systemd/system/ > /run/systemd/system/ > /usr/lib/systemd/system

表：types of systemd

扩展名	主要服务功能
.service	主要是系统服务，本地服务＋网络服务
.socket	内部程序数据交换的插槽服务，主要是 Inter-Process Communication 的 socket file 功能
.target	一群 unit 的集合
.mount/.automount	文件系统挂载相关的服务（automount unit / mount unit）
.path	侦测特定的文件或目录
.timer	循环执行的服务，类似于 crontab

透过 systemctl 管理服务

systemd 这个启动服务的机制，主要是透过 systemctl 的指令来处理的。

通过 systemctl 管理单一服务（service unit）的启动/开机自启与观察状态

状态	开机时预设的状态
active(running)	enabled
active(exited)	disabled
active(waiting)	static 不可配启动，但可被其他的 enabled 的服务唤起
inactive	masked 被注销

通过 systemctl 观察系统上所有的服务

systemctl list-units
列出目前已启动的 unit，若加上 --all 才会列出没启动的。
systemctl list-unit-files
依据 /usr/lib/systemd/system/ 内的文件，将所有文件进行列表说明。--type可指定type。

通过 systemctl 分析各服务之间的相依性

systemctl list-dependencies <unit> [--reverse]
[--reverse] 反向追踪，"谁使用了这个 unit"

与 systemd 的 daemon 运作过程相关的目录

- /usr/lib/systemd/system/
- /run/systemd/system/
- /etc/systemd/system/
- /etc/sysconfig/* 设定
- /var/lib/ 服务产生的数据
- /run/ daemon 的暂存档，包括 lock file 以及 PID file 等
- `systemctl list-sockets` 列出 socket file 的位置

关闭网络服务

"基本上，会产生一个 port 的程序，就可被称为网络服务"

```
systemctl stop <….service>
systemctl stop <….socket>
systemctl disable <….service> <….socket>
netstat -tlunp
```

systemctl 针对 service 类型的配置文件

systemctl 配置文件的相关目录

以 vsftpd.service 为例，

- /usr/lib/systemd/system/vsftpd.service (*)
- /etc/systemd/system/vsftpd.service.d/custome.conf 累加设定到 (*)
- /etc/systemd/system/vsftpd.service.wants/* 启动了 vsftpd.service 之后，最好再启动…
- /etc/systemd/system/vsftpd.service.requires/* 启动 vsftpd.service 之前，需要事先启动…

systemctl 的配置文件

[Unit]：unit 本身的说明，以及与其他相依 daemon 的设定，包括在什么服务之后才启动此 unit 之类的设定

[Service][Socket][Timer][Mount][Path]…
：不同的 unit type 会使用相对应的设定项目，主要用于规范服务启动的脚本、环境配置文件等

[Install]：将此 unit 安装到哪个 target 里面。

Description	说明
Documentation	文档
After	非强制
Before	非强制
Requires	强制要求此 unit 必须在哪个 daemon 启动后才能启动，即设定相依服务
Wants	非强制，"建议"在启动此 unit 之后还启动…
Conflicts	冲突性检查

Type	simple/forking/oneshot/dbus/idle 详见官网
EnvironmentFile	指定启动脚本的环境配置文件
ExecStart	详见原文
ExecStop	systemctl stop
ExecReload	systemctl reload
Restart	是否在此 daemon 服务终止后，再次启动此服务？除非使用 systemctl 将此服务关闭
RemainAfterExit	是否当此 daemon 所属的所有程序都终止后，是否再尝试启动此服务
TimeoutSec	强制结束启动或关闭
KillMode	process/control-group/none
RestartSec	sleep 多少时间再重新启动

WantedBy	这个 unit 本身附挂在哪一个 target unit 下
Also	当此 unit 本身被 enable 时，Also 后面跟的 unit 也"请" enable
Alias	详见官网

systemctl 针对 timer 的配置文件

"除了 crond 之外，如何使用 systemd 内建的 timer 来处理各种任务"

advantages：
- 所有的 systemd 的服务产生的信息都会被记录，因此比 crond 在 debug 上更清晰
- 各项 timer 的工作可以跟 systemd 的服务相结合
- 各项 timer 的工作可以跟 control group 结合，来限制该工作的资源利用
- crond 最小的单位到分，systemd 可以到秒甚至毫秒

具体的使用说明及实例见原书。