

Systemd

Servizos en GNU/Linux

Un servizo é un programa se executa en segundo plano e que realiza unha función de xeito transparente para o usuario. Non soen ser iterativos.

Tamén son chamados **daemons** (deimons), non hai que confundir coa imaxe do demo cristiano, ven do grego antigo onde os daemons eran unha especie de anxos da garda dos humanos.

Esta é a razón pola que moitos servizos rematan o seu nome en d

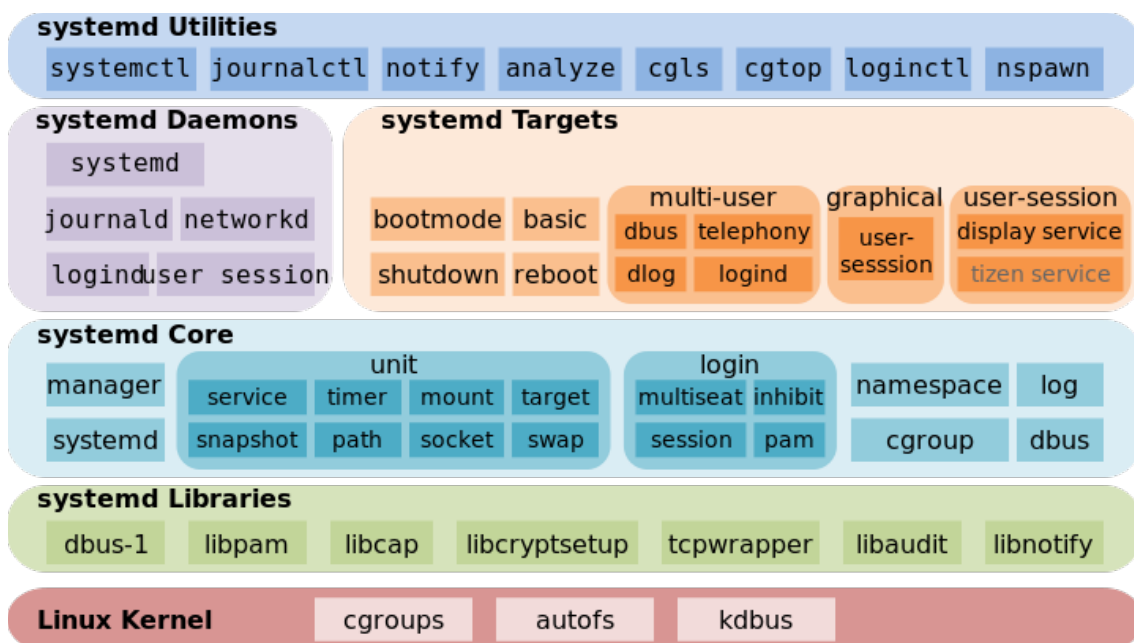
```
root@debianTesting:~# netstat -lpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           State      PID/Program name
tcp        0      0 0.0.0.0:22              LISTEN     382/sshd
tcp        0      0 127.0.0.1:631          LISTEN     336/cupsd
```

Systemd

É, entre outras cousas, o **sistema encargado de xestionar o inicio de GNU/Linux**. É o primeiro proceso, e polo tanto o encargado de iniciar os servizos.

É o sistema de inicio estándar de Debian dende Debian 8 (2015), pero sufriu un importante rechazo por parte da comunidade xa que rompe o enfoque **KISS (Keep It Simple Stupid!)** tradicional das aplicacións GNU/Linux. A idea é que os programas fagan unha soa cousa, pero que a fagan ben.

Systemd realiza moitas tarefas e é moi dependente do kernel.



Optimizando o inicio do equipo

Os sistemas intentan cada vez máis facilitarlle a vida ó usuario, polo que durante o inicio do sistema inician servizos por se os necesitamos. Pero que en caso de non necesitalos están ocupando recursos innecesarios e retrasando o inicio do sistema.

Podemos examinar o arranque con `systemd-analyze`

- **systemd-analyze** : Amonsa o tempo empregado no inicio do sistema
- **systemd-analyze blame**: Amonsa o tempo empregado por cada servizo durante o inicio

```
systemd-analy blame
17.492s apt-daily.service
15.073s apt-daily-upgrade.service
7.926s libvirtd.service
6.827s networking.service
4.620s nmbd.service
4.242s accounts-daemon.service
3.874s winbind.service
3.874s vboxdrv.service
3.864s smbd.service
3.625s ModemManager.service
```

Podemos demorar o inicio de algúns servizos ou desactivar servizos innecesarios

Xestionando os servizos

Temos dúas alternativas: **Service** OU **systemctl**.

Realmente **Service** é un front-end que chama a **systemctl**.

	Service	Systemctl
Listar servizos activos	<code>service --status-all</code>	<code>systemctl list-units --type=service --state=running</code>
Información dun servizo	<code>service servicio status</code>	<code>systemctl status servicio</code>
Iniciar un servizo	<code>service servicio start</code>	<code>systemctl start servicio</code>
Deter un servizo	<code>service servicio stop</code>	<code>systemctl stop servicio</code>
Deshabilitar servizo		<code>systemctl disable servicio</code>
Habilitar servizo		<code>systemctl enable servicio</code>

Exemplos:

```
root@debian:~# service --status-all
[ - ] anacron
[ + ] avahi-daemon
[ - ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ + ] cups
[ - ] cups-browsed
root@debian:~# service cups stop
root@debian:~# service --status-all
[ - ] anacron
[ + ] avahi-daemon
[ - ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ - ] cups
[ - ] cups-browsed
```

Identificando os servizos

Os seguintes son algúns servizos que poderíamos desactivar

Nome	Función
apt-daily	Actualizacións automáticas
avahi	É un sistema que facilita o descubrimento de servizos nunha rede local
Bluetooth	Dispositivos Bluetooth
Cups	Servidor de impresión

Podemos atopar un listado máis completo [aquí](#)

Unidades (Systemd Units)

Systemd pode configurar moitos aspectos do sistema: servizos, puntos de montaxe, logs. A súa xestión esta baseada no concepto de Unidade.

Unha unidade é un obxecto que representa un obxecto do sistema manexado por Systemd.

A súa configuración faixe mediante os **unit files**. Teremos distintos tipos de units dependendo do recurso a xestionar, por exemplo as units relacionadas con servizos teñen o sufixo **.service**.

Para amosar as units relacionadas con servizos que se inician no sistema

```
usuario@debian:~$ systemctl | grep service
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
ssh.service                        loaded active running OpenBSD Secure Shell..
```

Para obter información dun servizo concreto

```
usuario@debian:~$ systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Mon 2019-06-24 11:34:35 CEST; 37min ago
   Process: 516 ExecReload=/bin/kill -HUP $MAINPID (code=exited,
status=0/SUCCESS)
   Process: 511 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Process: 388 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 458 (sshd)
     Tasks: 1 (limit: 4915)
    CGroup: /system.slice/ssh.service
            └─458 /usr/sbin/sshd -D
```

Agora xa sabemos a ruta do seu UNIT File, o seu arquivo de configuración

```
usuario@debian:~$ more /lib/systemd/system/ssh.service
```

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run
```

```
[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
```

```
[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Creando o noso propio servizo que inicie co inicio do equipo.

- Definir o noso unit file

```
root@debian:~# vi /lib/systemd/system/myservice.service
```

```
[Unit]
Description=Example systemd service.
```

```
[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/test_service.sh
```

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

Targets

Os [targets](#) son outro tipo de unidades que podemos configurar en Systemd. Os targets agrupan varias unidades e empréganse para definir o estado que debe alcanzar un sistema cando se inicie (servizos que debe executar...)

Podemos listar os targets dispoñibles con

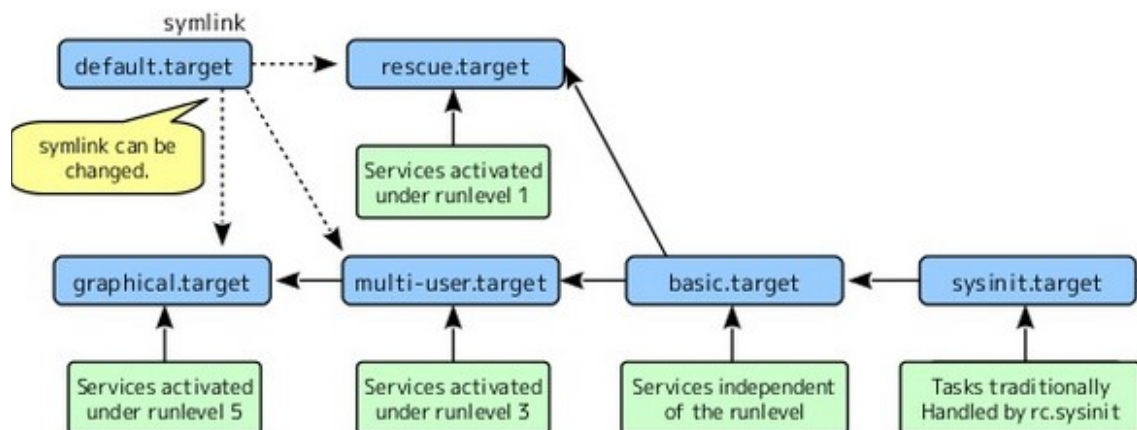
```
rojas@debian:~$ ls /lib/systemd/system/*.target
/lib/systemd/system/basic.target
/lib/systemd/system/printer.target
/lib/systemd/system/bluetooth.target
/lib/systemd/system/reboot.target
```

Podemos comprobar cal é o noso target por defecto

```
rojas@debianRojas:~$ ls -l /lib/systemd/system/default.target
lrwxrwxrwx 1 root root 16 Out 16 15:24 /lib/systemd/system/default.target ->
graphical.target
```

Os targets teñen dependencias entre eles. Para que un target se complete é preciso que outros se completen antes

```
rojas@debian:~$ more /lib/systemd/system/graphical.target
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
```



Podemos cambiar o target

```
rojas@debian:~$ systemctl isolate multi-user.target
```

Para cambiar o target por defecto

```
rojas@debian:~$ systemctl set-default multi-user.target
```

Comandos de información do Sistema

- Examinando tiempo de inicio del sistema

```
root@debianTesting:~# systemd-analyze
Startup finished in 1.307s (kernel) + 1.780s (userspace) = 3.088s
```

- Información sobre el equipo

```
root@debianTesting:~# hostnamectl
Static hostname: debianTesting
Icon name: computer-vm
Chassis: vm
Machine ID: d22291cfe0cb4a0a929d6e716bbb3fb2
Boot ID: b6c8778fdcf343a1963d222da1756155
Virtualization: oracle
Operating System: Debian GNU/Linux 9 (stretch)
Kernel: Linux 4.9.0-8-amd64
Architecture: x86-64
```

- Información sobre a configuración horaria

```
root@debianTesting:~# timedatectl
Local time: Lun 2019-03-18 00:20:13 CET
Universal time: Dom 2019-03-17 23:20:13 UTC
RTC time: Dom 2019-03-17 23:20:13
Time zone: Europe/Madrid (CET, +0100)
Network time on: yes
NTP synchronized: yes
RTC in local TZ: no
```

Examinando os logs do Sistema

Tradicionalmente para examinar os logs do sistema empregábase a carpeta /var/logs. Nesa carpeta temos distintos arquivos que almacenan información provinte de distintos orixes en formato texto. **rsyslog** era o encargado de xestionar os logs.

Pero a partir de **Debian 12 Bookworm**, rsyslog xa non se instala por defecto e os logs son xestionados por **systemd-journald**. Systemd intenta unificar a información de todos eses arquivos. Agora todos os logs son almacenados nun arquivo binario.

- Para examinar a información do sistema dende o último arranque

```
root@debian:~# journalctl
-- Logs begin at Thu 2020-01-16 18:09:08 CET, end at Thu 2020-01-16 18:55:49 CET
Xan 16 18:55:10 debianRojas kernel: Linux version 4.19.0-6-amd64 (debian-kernel@
Xan 16 18:55:10 debianRojas kernel: Command line:
BOOT_IMAGE=/boot/vmlinuz-4.19.
Xan 16 18:55:10 debianRojas kernel: x86/fpu: Supporting XSAVE feature 0x001:
'x8
```

Podemos movernos polo listado, e procurar unha cadea de texto.

- Para filtrar as mensaxes de erro

```
root@debian:~# journalctl -p err
```

- Podemos examinar os logs nun periodo de tempo

```
root@debian:~# journalctl --since 'yesterday' --until '00:00'
```

- Para ver as entradas máis recentes

```
root@debian:~# journalctl -r
```

- Logs por tipo
 - Logs dende o arranque actual

```
root@debian:~# journalctl -b
```
 - Logs de sistema

```
root@debian:~# journalctl -k
```
 - Para ver os logs dunha unit, por **exemplo un servizo**

```
root@debian:~# journalctl -u ssh
```

- Para ver os logs do kernel do último arranque

```
journalctl --dmesg
```

- Para ver os últimos arranques

```
journalctl -list-boots
```


Examinando os logs de inicio de sesión

Os inicios de sesión están xestionados por **systemd-logind**

- Para ver quen iniciou sesión localmente

```
journalctl -u systemd-logind | grep "New session"
```

- Para ver quen iniciou sesión localmente hoxe

```
journalctl -u systemd-logind --since "today" | grep "New session"
```

Os inicios de sesión vía ssh están xestionados por **ssh**

- Para ver os inicios de sesión vía ssh dende onte

```
journalctl -u ssh --since yesterday
```

- Para ver os inicios de sesión por ambas vías

```
journalctl -u systemd-logind -u ssh --since today
```

Referencias

- <https://opensource.com/article/20/8/journals-systemd>