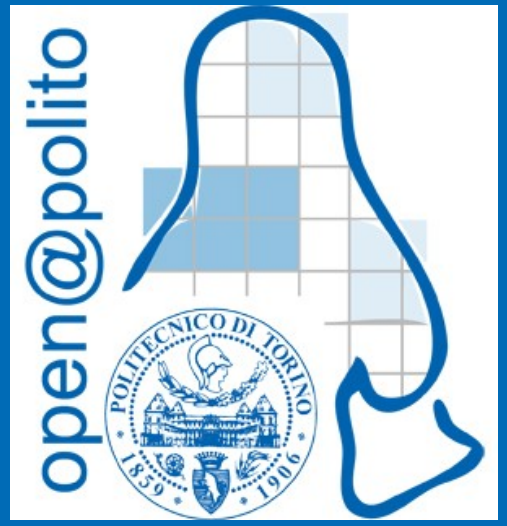
Gestione dei processi con Systemd

Di Rosario Antoci

Con il supporto di:



What is systemd? (easy)

systemd

system daemon

What is systemd? (easy)

E’ un init system e system manager open-source

per sistemi GNU/Linux:

- Avvio parallelo dei servizi.
- Gestione delle dipendenze.
- Registrazione centralizzata con journald

Sostituisce i vecchi sistemi di init (come SysVinit) con un design moderno ed efficiente.

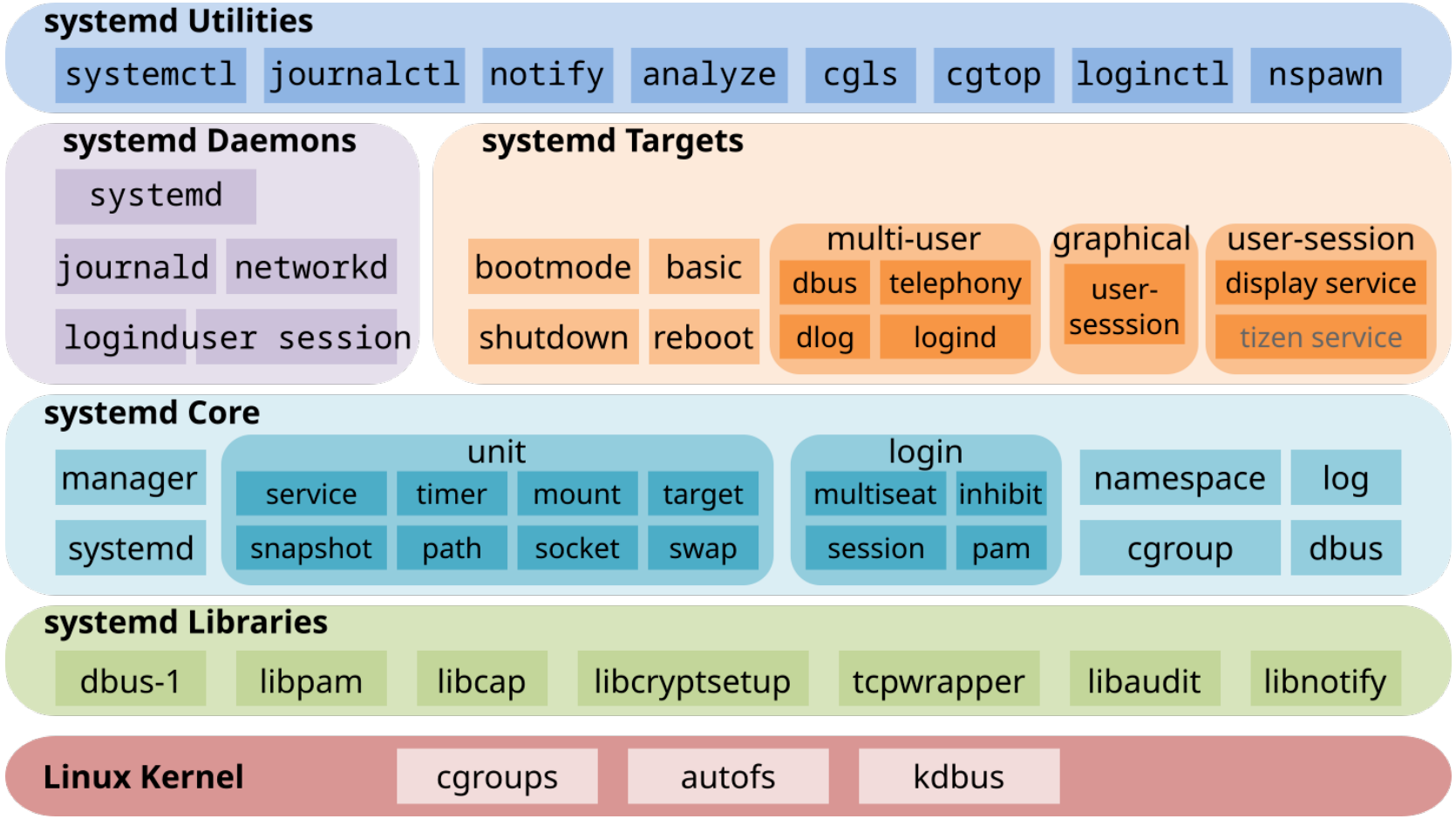
History

Rilasciato per la prima volta nel 2010

dallo sviluppatore tedesco Lennart Poettering

(in foto)

What is Systemd? (complex)



Architettura di systemd (easy)

Unit: **Systemd** organizza i **task** in **unit**. Tipi di unit comuni:

Service (.service): Gestisce i servizi.

Timer (.timer): Pianifica le attività.

Target (.target): Raggruppa unità (es. multi-user.target per la modalità multiutente).

Socket (.socket): Per l'attivazione basata su socket.

Strumenti a riga di comando:

systemctl: Gestisce e interroga lo stato delle **unit**.

journalctl: Accede ai **log**.

Practical setup

systemctl –version

systemctl list-units

init system ?

In un sistema UNIX o UNIX-like (tra cui Linux),

L' **init** e' quel particolare **programma** che viene lanciato per primo

all'avvio del sistema operativo

Il suo **Process ID** e' e sarà sempre 1 (**PID 1**), ed e' per definizione

l'antenato, diretto o indiretto, di tutti i processi del sistema

E' responsabile dell'avvio e della gestione di tutti gli altri processi

processo

E’ un’ **istanza** di un programma **in esecuzione**

Ha un **Process ID (PID)** che lo identifica all’interno della lista dei processi del sistema

Ha delle risorse **dedicate**

(memoria, CPU, e quant’altro)

Puo’ essere generato da un utente o da un altro processo

Perché systemd

E’ nato con l’obiettivo di **uniformare**

la gestione dei sistemi Linux...

...e ci e’ riuscito!

A partire dal 2010, e’ progressivamente diventato l’**init system**

piu’ adottato dalle distribuzioni Linux

Red Hat Enterprise Linux (RHEL) ha iniziato ad adottare

systemd dalla versione 7.0

Linux prima di systemd

```
$ /etc/init.d/httpd reload
$ /etc/init.d/sshd start
$ /etc/init.d/crond restart
```

Ciclo di vita di una unit

Stati:

Active: Il servizio è in esecuzione.

Inactive: Il servizio è fermo.

Failed: Il servizio ha incontrato un errore.

Dipendenze delle unità: Usa **After**=, **Requires**= e **Wants**= per gestire le dipendenze.

Ciclo di vita di una unit

Stati:

Active: Il servizio è in esecuzione.

Inactive: Il servizio è fermo.

Failed: Il servizio ha incontrato un errore.

Dipendenze delle unità: Usa **After**=, **Requires**= e **Wants**= per gestire le dipendenze.

Cos'è una Service Unit?

Un file **.service** descrive un servizio gestito da systemd.

Struttura di un file di servizio:

[Unit]: Metadati e dipendenze.

[Service]: Comportamento del servizio (comandi, policy di riavvio).

[Install]: Opzioni di abilitazione.

Practical setup

Crea un file per lo script hello.py

```
#!/usr/bin/env python3
print("Hello da WEEE Open & Netstudent!")
```

Assegna i permessi di esecuzione:

```
chmod +x hello.py
```

Practical setup

Crea la service unit hello.service

[Unit]

Description=Hello World Service

[Service]

ExecStart=/tmp/hello.py

Restart=no

[Install]

WantedBy=multi-user.target

Practical setup

Esegui un comando alla volta chiedendoti cosa esegue ogni singola azione:

```
sudo cp hello.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl start hello.service
sudo systemctl enable hello.service
```

Comandi utili

systemctl status hello.service

systemctl stop hello.service

systemctl reset-failed hello.service
journalctl -u hello.service

Giochiamo con i timer

Cos'è una Timer Unit?

Un file **.timer** pianifica le attività in base al tempo o agli intervalli.

Casi d'uso comuni:

- Automazione dei backup.
- Pianificazione dell'esecuzione di script.
- Necessita’ di maggiore controllo rispetto a **crond**

Giochiamo con i timer

Sezioni fondamentali per la **.timer** unit:

[Unit]: Metadati.

[Timer]: Dettagli di pianificazione (OnCalendar=, OnBootSec=, OnUnitActiveSec=).

Practical setup

Crea la timer unit hello.timer:

```
[Unit]
Description=Esegui Hello Service ogni minuto

[Timer]
OnBootSec=10s
OnUnitActiveSec=1min

[Install]
WantedBy=timers.target
```

Practical setup

```
sudo cp hello.timer /etc/systemd/system/
sudo systemctl start hello.timer
sudo systemctl enable hello.timer
```

Altri comandi utili

systemd-analyze

systemd-analyze blame

Key-value avanzati

```
[Timer]
OnCalendar=daily
OnUnitActiveSec=5m
OnStartupSec=60m
RandomizedDelaySec=60m
Persistent=true
OnCalendar=*-*-* 06..18:00/30 <=== Chi mi sa rispondere?
```

Domande?

Grazie per l’attenzione!

Segui torino.ils.org & Co.

Appuntamento settimanale:

Sportello di Assistenza Informatica:

tutti i mercoledi

dalle 18:00 alle 19:30

presso la Casa del Quartiere

via Oddino Morgari 14, Torino