

systemd - Système D ?

Matthieu Herrb



Capitoul, 13 Novembre 2015

<http://homepages.laas.fr/matthieu/talks/systemd.pdf>



Ce document est sous licence

Creative Commons Paternité - Partage à l'Identique 3.0 non transposé.

Le texte complet de cette licence est disponible à l'adresse :

<http://creativecommons.org/licenses/by-sa/3.0/>

Agenda

- 1 Introduction
- 2 Commandes de base
- 3 Configuration d'un service
- 4 Conclusion

Agenda

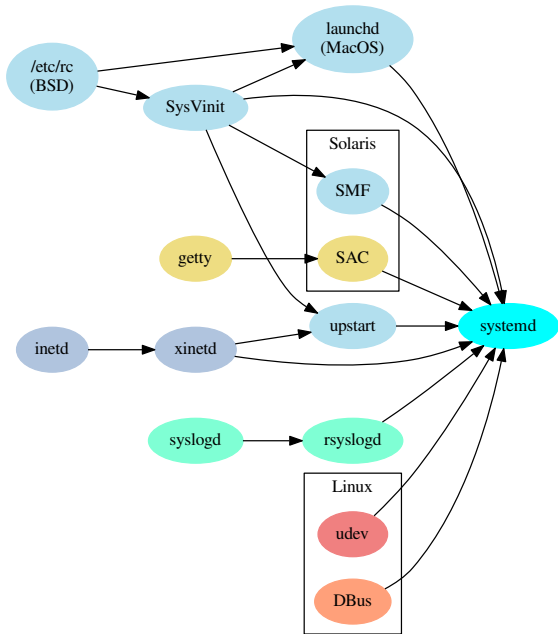
- 1 Introduction
- 2 Commandes de base
- 3 Configuration d'un service
- 4 Conclusion

Nouveau système de gestion du système pour Linux

- Parallèle → démarrage rapide
- Dynamique → ajout retrait matériel à chaud
- Sécurisé (?) → gère cgroups, chroot, containers,...

Conséquences :

- Nouvelles commandes d'administration
- Nouveaux problèmes
- Intégration de services « maison » ?



Composants

- systemd - gestion du système
- systemd-udev - gestion des périphériques
- system-journald - gestion des journaux
- systemd-logind - gestion des sessions utilisateurs
- systemd-networkd - gestion réseau
- systemd-...

Unités :

- services
- ressources (devices, montages, partitions,...)
- groupes d'autres unités (targets)
- déclencheurs (triggers) timers, ...

Agenda

- 1 Introduction
- 2 Commandes de base**
- 3 Configuration d'un service
- 4 Conclusion

B A BA systemd

- `systemctl disable sshd`
- `systemctl enable sshd`
- `systemctl start sshd`
- `systemctl stop sshd`
- `systemctl restart sshd`
- `systemctl reload sshd`
- `systemctl status sshd`

- `journalctl -e` (end)
- `journalctl -f` (follow)
- `journalctl -b` (this boot)
- `journalctl -u sshd` (unit)

Systemd comparé à init

Runlevel	Target	lien vers
Runlevel 0	runlevel0.target	poweroff.target
Runlevel 1	runlevel1.target	rescue.target
Runlevel 2	runlevel2.target	multi-user.target
Runlevel 3	runlevel3.target	multi-user.target
Runlevel 4	runlevel4.target	multi-user.target
Runlevel 5	runlevel5.target	graphical.target
Runlevel 6	runlevel6.target	reboot.target

```
systemctl get-default  
systemctl set-default multi-user.target
```

(/etc/systemd/system/default.target, ex /etc/inittab)

```
systemctl poweroff  
systemctl reboot
```

Manipulation des unités

```
systemctl list-units  
systemctl list-units -t service  
systemctl list-unit-files -a  
systemctl list-unit-files -a -t service
```

Trucs & astuces

```
systemd-analyze
systemd-analyze blame
systemd-analyze critical-chain
systemd-analyze critical-chain expenseit-web
systemctl list-dependencies expenseit-web
systemctl kill expenseit-web
journalctl /usr/sbin/sshd
journalctl -b -1
journalctl --since=today
journalctl -p err
journalctl -u expenseit-web -e -b
journalctl -u expenseit-web -f
systemctl show mysqld | grep CPUShares
systemctl suspend
systemctl hibernate
```

Agenda

- 1 Introduction
- 2 Commandes de base
- 3 Configuration d'un service**
- 4 Conclusion

`http://in.waw.pl/git/login-article-example/`

Description d'une unité « service » :

```
# /etc/systemd/system/hasher.service
[Unit]
Description=Text hashing service
Documentation=https://example.com/hasher
[Service]
ExecStart=/usr/bin/python -m hasher
```

Démarrage :

```
systemctl start hasher
```

Arrêt :

```
systemctl stop hasher
```

Fichiers de configuration

Systemd cherche dans l'ordre dans:

- 1 `/etc/systemd/system/`,
pour sysadmin
- 2 `/run/systemd/system/`,
écrasé à chaque reboot
- 3 `/lib/systemd/system/`,
paquets système - ne pas modifier
- 4 `/usr/lib/systemd/system/` (?)

“Droppins” :

exemple : `/etc/systemd/system/hasher.service.d/`
répertoire qui contient des bouts de config en plus du fichier
`hasher.service`.

Drappins - exemple

```
systemctl edit --system hasher
```

→ /etc/systemd/system/hasher.service.d/override.conf

```
[Service]
ExecStart=
ExecStart=/usr/bin/python3 -m hasher
```

Puis

```
systemctl daemon-reload
systemctl restart hasher
```

Voir la définition d'une unité de service :

```
systemctl cat hasher
```


Activation par socket

Systemd pré-crée tous les sockets utilisés par les services
⇒ plus besoin de décrire l'ordre d'activation...

Géré par des fichiers `.socket` :

```
# /etc/systemd/system/haser.socket
[Unit]
Description=Text hashing service socket
[Socket]
ListenStream=9001
```

Supporte sockets IPv4, IPv6, Unix, pipes nommés, etc.

Démarrage automatique

Services groupés en *targets*

Par défaut pour démarrage : `multiuser.target`

```
# /etc/systemd/system/hasher.service
[Unit]
Description=Text hashing service
Requires=hasher.socket
[Service]
...
[Install]
WantedBy=multi-user.target
```

Activation d'un service

```
$ systemctl enable hasher.service  
Created symlink from  
/etc/systemd/system/multi-user.target.wants/hasher.service  
to /etc/systemd/system/hasher.service.
```

Toutes les sorties des services sont interceptés par *journald*.

```
systemctl start hasher
journalctl -u hasher
-- Logs begin at Sun 2015-11-08 19:36:01 CET,
   end at Wed 2015-11-11 15:49:19 CET
Nov 11 15:49:13 paperthin systemd[1]:
   Started Text hashing service.
Nov 11 15:49:13 paperthin python3[20937]: /usr/bin/python3:
   No module named hash
Nov 11 15:49:13 paperthin systemd[1]: hasher.service:
   Main process exited, code=
Nov 11 15:49:13 paperthin systemd[1]: hasher.service:
   Unit entered failed state.
Nov 11 15:49:13 paperthin systemd[1]: hasher.service:
   Failed with result 'exit-c'
```

Définition de l'uid qui exécute le service, et restriction des droits

```
# /etc/systemd/system/hasher.service
...
[Service]
User=hasher
NoNewPrivileges=yes
ProtectHome=yes
ProtectSystem=full
PrivateTmp=yes
PrivateNetwork=yes
```

Agenda

- 1 Introduction
- 2 Commandes de base
- 3 Configuration d'un service
- 4 Conclusion**

Conclusion

- Systemd a gagné → nouveau standard de fait
- Pas encore complètement sec
- Demande des modifications dans services / applications existants
- Devrait rendre le système plus homogène...
mais aujourd'hui : plein de couches de compatibilité
- Documentation abondante mais pas pas toujours utilisable
- Complexité problématique
- Linux est-il encore un UnixTM ?

→ Va obliger à revoir tous les cours d'administration et de développement système Linux

Bref, pas fini de rire...

Questions ?

- *Daemon Management Under Systemd*, Zbigniew Jędrzejewski-Szmek & Jóhann B. Guðmundsson, ;Login: June 2015, Vol. 40 no. 3
- <http://www.freedesktop.org/wiki/Software/systemd/>
- <https://linuxfr.org/news/évolutions-techniques-de-systemd>
- <http://www.h-online.com/open/features/Control-Centre-The-systemd-Linux-init-system-1565543.html>
- <http://martin.kopta.eu/activity/talks/systemd/systemd.txt>