

Systemd

- Objet : Présentation des commandes systemd
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Ce tuto vise à être un aide mémoire aux commandes systemd.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[doublon](#)
 - Création par [deuchdeb](#) le 05/07/2015
 - Testé par [deuchdeb](#) le 05/07/2015
 - Actualisé/compléter par [smolski](#) avec l'aide du forum le 21/08/2018
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#)¹⁾

Introduction

Systemd est une alternative au démon **init** de **System V** spécifiquement conçu pour le noyau **Linux**.

Il a pour but :

- d'offrir un meilleur cadre pour la gestion des dépendances entre services,
- de permettre le chargement en parallèle des services au démarrage,
- et de réduire les appels aux scripts shell.

Le projet a été lancé par **Lennart Poettering** en 2010 et publié sous licence **GNU LGPL version 2.11**.

Le nom de ce programme vient de « **system daemon** » : le daemon du système.

C'est devenu le système d'init par défaut de Debian depuis la version Jessie (Debian 8).

Installation

Déjà installé par défaut.

Et on dit que le libre c'est compliqué dit donc ! 😊

Utilisation

Un lien absolument nécessaire à consulter :

- [Créer et modifier des fichiers d'unité systemd.](#)

Merci **Severian** 😊

Post sur le forum df :

- <https://debian-facile.org/viewtopic.php?pid=274804#p274804>

Les journaux système

Remarque importante !

Seuls l'utilisateur « root » ou le groupe « systemd-journal » peuvent consulter le journal.

Visualiser tout le journal

```
journalctl
```

Visualiser le journal relatif à un service particulier (exemple cups)

```
journalctl -u cups
```

Visualiser le journal relatif à un processus particulier

```
journalctl _PID=1
```

Visualiser le journal relatif à un exécutable particulier (exemple dhcpcd)

```
journalctl /usr/sbin/dhcpcd
```

Visualiser le journal du jour

```
journalctl --since="today"
```

Visualiser le journal sur une période

```
journalctl --since="AAAA-MM-JJ hh:mm:ss" --until="AAAA-MM-JJ hh:mm:ss"
```

Visualiser le journal relatif au dernier boot (-b) avec les messages explicatifs (-x)

```
journalctl -xb
```

Visualiser les erreurs dans le journal (-p err) en commençant par la plus récente (-r)

```
journalctl -r -p err
```

Scripts auto

On peut activer automatiquement des scripts root au démarrage.

Exemple

Pour activer par exemple le script `/opt/monscript.sh` au démarrage de debian.

1. Créer le fichier `monscript.service` avec le code suivant :

```
cat >/etc/systemd/system/monscript.service <<EOF
[Unit]
Description=/opt/monscript.sh Compatibility
ConditionPathExists=/opt/monscript.sh
after=NetworkManager.service

[Service]
Type=forking
ExecStart=/opt/monscript.sh start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99

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

À noter le démarrage du réseau pour l'utilisation de l'application [Network Manager](#) :
`after=NetworkManager.service.`



Si vous n'utilisez pas l'application `network-manager`, Freemaster signale sur le forum cet argument à mettre à la place :
`After=network.target`

1. Créer le fichier `/opt/monscript.sh` ainsi :

```
cat > /opt/monscript.sh <<EOF
#!/bin/sh -e
# /opt/monscript.sh
```

```
commande <argument> <options> <etc>
```

```
exit 0  
EOF
```

2. Ajouter l'[autorisation d'exécution](#) au script /opt/monscript.sh :

```
chmod +x /opt/monscript.sh
```

3. Et activer le service :

```
systemctl enable monscript
```

Merci à Freemaster et vv222 sur le forum là :

- <https://debian-facile.org/viewtopic.php?pid=274785#p274785>

Que leurs pas soient à jamais parfumés de pétales de roses...

Les unités

Systemd gère des services système mais également des périphériques, des points de montage, des timers, et d'autres composants du système.

Le terme générique pour tous ces objets dans **systemd** est une unité (**Unit**).

La section **[Unit]** stocke l'information qui s'applique non seulement aux services mais également à tous les autres types d'unité **systemd**.

Convention de nommage

Chaque unité utilise un nom d'extension en rapport avec son type :

```
service : *.service  
groupe d'unités : *.target  
montage : *.mount  
socket : *.socket
```

Utilisation des unités

Lister les unités

```
systemctl list-units
```

Statut d'une unité

```
systemctl status kdm.service
```

Activer une unité (ici le gestionnaire graphique kdm)

```
systemctl enable kdm.service
```

Désactiver une unité (ici le gestionnaire graphique kdm)

```
systemctl disable kdm.service
```

Démarrer une unité (ici le gestionnaire graphique kdm)

```
systemctl start kdm.service
```

Arrêter une unité (ici le gestionnaire graphique kdm)

```
systemctl stop kdm.service
```

Regarder si il y a des unités qui posent problème

```
systemctl status
```

Gestion au niveau utilisateur

Pour gérer les services au niveau utilisateur, il faut ajouter l'option « --user ».
Par exemple désactiver le service « dropbox » ainsi :

```
systemctl --user disable dropbox
```

Les cibles (targets) : la méthode systemd pour les niveaux d'exécutions

Il est possible de regrouper plusieurs unités dans une unité spéciale dont le nom d'extension est *target* (cible).

Cela permet de reproduire le fonctionnement des *niveaux d'exécution* (*runlevels* de SysVinit).

Pour lister toutes les cibles²⁾ :

```
systemctl list-units --type=target -all
```

Voici un tableau de correspondance avec l'ancien système :  : tableau à contrôler :

Niveau d'exécution	Cible systemd	Notes
0	poweroff.target	Arrêt de l'ordinateur
1	rescue.target	Mode mono-utilisateur
2, 3, 4	multi-user.target	Mode multi-utilisateur en mode texte
5	graphical.target	Mode multi-utilisateur en mode graphique
6	reboot.target	Redémarrage de l'ordinateur
emergency	emergency.target	Mode de secours

Par défaut, lorsque un gestionnaire de connexion est installé, la cible par défaut est `graphical.target`. Pour changer la cible par défaut au démarrage :

- Avec `systemctl` :

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

- Ou bien, lors de l'amorçage de l'ordinateur, en ajoutant directement à la fin de la ligne `boot` :

```
systemd.unit=multi-user.target
```

Diagnostic système

Liens entre les services créés par systemd et les processus

```
systemd-cgls
```

Cela va créer une arborescence qui permettra de faire le lien entre les services et les processus.

les performances de démarrage

quelques outils intégré à systemd pour analyser le démarrage du système

temps simple du démarrage du système :

```
systemd-analyze
```

```
Startup finished in 1.456s (kernel) + 6.752s (userspace) = 8.208s
```

temps détaillé de démarrage du système :

```
systemd-analyze blame
```

```
1min 4.746s nmaj.service
5.427s exim4.service
386ms media-Divertissement.mount
177ms ModemManager.service
162ms keyboard-setup.service
115ms media-docs.mount
115ms accounts-daemon.service
97ms ufw.service
92ms rsyslog.service
85ms systemd-logind.service
84ms systemd-fsck@dev-disk-
by\x2duuid-8a01a86d\x2d67be\x2d46d7\x2d821f\x2def50f7
84ms lm-sensors.service
82ms alsa-restore.service
81ms console-setup.service
78ms irqbalance.service
73ms systemd-fsck@dev-disk-by\x2duuid-BC47\x2d8202.service
72ms media-divers.mount
54ms networking.service
51ms systemd-resolved.service
44ms lightdm.service
43ms systemd-tmpfiles-setup-dev.service
38ms systemd-networkd.service
37ms kbd.service
32ms systemd-tmpfiles-clean.service
31ms user@1000.service
30ms systemd-fsck@dev-disk-by\x2duuid-
bc2b1652\x2d35c6\x2d4688\x2db7d2\x2d11de53
27ms tmp.mount
26ms resolvconf.service
23ms polkitd.service
22ms systemd-setup-dgram-qlen.service
21ms udisks2.service
20ms systemd-backlight@backlight:acpi_video0.service
19ms systemd-udev-trigger.service
19ms home.mount
19ms dev-mqueue.mount
18ms systemd-random-seed.service
17ms sys-kernel-debug.mount
16ms user@110.service
16ms rdnssd.service
15ms hddtemp.service
14ms dev-hugepages.mount
14ms var.mount
13ms hdparm.service
13ms systemd-modules-load.service
13ms keymap.service
12ms systemd-user-sessions.service
11ms dev-disk-by\x2duuid-
c67a739b\x2d917c\x2d4b33\x2dbf0d\x2dd72a0321d356.swap
10ms boot-efi.mount
```

```
10ms gdomap.service
 9ms rc-local.service
 9ms systemd-journal-flush.service
 8ms systemd-remount-fs.service
 8ms kmod-static-nodes.service
 8ms dev-sda5.swap
 7ms systemd-update-utmp.service
 6ms systemd-sysctl.service
 5ms systemd-tmpfiles-setup.service
 4ms systemd-update-utmp-runlevel.service
 4ms systemd-udev.service
 3ms rtkit-daemon.service
 2ms sys-fs-fuse-connections.mount
 1ms udev-finish.service
```

```
systemd-analyze plot > plot.svg
```



Systemd en mode graphique

paquet : **systemd-ui**

<https://packages.debian.org/jessie/systemd-ui>

Installation

```
apt-get install systemd-ui
```

Ce tuto vient très largement de **Yopland** ici: http://www.planet-libre.org/?post_id=18293

Sources d'informations sur systemd

- <http://linuxfr.org/news/systemd-pour-les-administrateurs-partie-1-et-2>
- <http://linuxfr.org/news/systemd-pour-les-administrateurs-parties-3-4-et-5>
- <http://www.funix.org/fr/linux/startup.htm>

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

actives ou non

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:systeme:systemd>



Last update: **28/04/2023 09:06**