

LINUX -> Zarządzanie usługami - Systemd, systemctl.

Dodał / zmodyfikował: pkania, w dniu: 2015-10-15

W "nowszych" dystrybucjach Linux-a (tj. w tych wydawanych po 13.10.2012r.) natkniemy się na inny sposób zarządzania serwisami / usługami niż dotychczas. Mowa tu o **Systemd**. Systemd, tak jak jego poprzednik SysVinit odpowiada w głównej mierze za sterowanie procesami rozruchowymi oraz usługami.

SysVinit zawierał polecenia typu: `chkconfig` oraz `service`. Za pomocą tych poleceń dodawaliśmy program do autostartu, uruchamialiśmy go, sprawdzaliśmy aktualny stan usługi itd.

Wymienione powyżej polecenia nadal spełniają swoją rolę (pozostawione pewnie ze względu na nawyki administratorów), ale nie wiadomo jak długo będą one wspierane i dlatego należy zapoznać się z nowym sposobem zarządzania usługami.

Do zarządzania głównie wykorzystujemy polecenie **systemctl**. Polecenie `systemctl` wydane bez żadnych opcji wyświetli wszystkie aktywne (uruchomione) usługi. Możemy też dodać opcję `-all` spowoduje ona wyświetlenie wszystkich usług (nawet tych nieaktywnych).

Dodanie / usunięcie usługi do / z "autostartu".

Schemat użycia: **`systemctl enable/disable nazwaUsługi.service`**

Przykłady:

`systemctl enable httpd.service` (dodanie ".service" na dzień dzisiejszy jest opcjonalne).
`systemctl disable httpd.service`

Zarządzanie manualne usługami.

Schemat użycia: **`systemctl start / stop / restart / status nazwaUsługi.service`**

Przykłady:

```
systemctl start httpd.service
```

```
systemctl stop httpd.service
```

```
systemctl status httpd.service
```

Analiza przykładowego wyniku polecenia *systemctl enable*.

```
www:~ # systemctl enable nmb.service
```

```
ln -s '/usr/lib/systemd/system/nmb.service' '/etc/systemd/system/multi-  
user.target.wants/nmb.service'
```

Jak widać na powyższym przykładzie w wyniku tego polecenia został utworzony link symboliczny w lokalizacji `/etc/systemd/system/multi-user.target.wants/nmb.service` wskazujący na plik właściwy: `/usr/lib/systemd/system/nmb.service`.

Analiza przykładowego wyniku polecenia *systemctl status*.

```
www:~ # systemctl status nmb.service
```

```
nmb.service - Samba NMB Daemon
```

```
Loaded: loaded (/usr/lib/systemd/system/nmb.service; disabled)
```

```
Active: inactive (dead)
```

Napis w tym kolorze wskazuje na lokalizację skryptu do zarządzania daną usługą (w tym przypadku nmb).

Ten kolor mówi, że usługa na ten moment nie została dodana do "autostartu".

Ten kolor powiadamia nas o tym, że usługa w tym momencie nie jest uruchomiona.

Zawartość skryptu `/usr/lib/systemd/system/nmb.service`:

```
[Unit]
```

```
Description=Samba NMB Daemon
```

```
After=syslog.target network.target
```

```
[Service]
```

```
Type=notify
```

```
NotifyAccess=all
```

```
Environment=KRB5CCNAME=/run/samba/krb5cc_samba
PIDFile=/run/samba/nmbd.pid
EnvironmentFile=-/etc/sysconfig/samba
ExecStart=/usr/sbin/nmbd $NMBDOPTIONS
ExecReload=/usr/bin/kill -HUP $MAINPID
```

[Install]

```
WantedBy=multi-user.target
```

Jak można zauważyć budowa takiego skryptu jest dużo prostsza i czytelniejsza, niż skryptów znanych z SysVinit, w skład których wchodził np. język bash itd.

W tym przypadku mamy jawny podział na sekcję, a każda z nich posiada zdefiniowane zmienne wraz z ich wartościami.

[Unit] After określa moment uruchomienia usługi (w tym przypadku po uruchomieniu sysloga oraz obsługi sieci).

[Service] opcje serwisu np. dla polecenia start odwołaj się do demona /usr/sbin/nmbd z ew. opcjami.

Sekcja [Install] wskazuje na poziom pracy w jakim usługa ma być uruchamiana (związane z poleceniem `systemctl enabled`). W tym przypadku link symboliczny do skryptu startowego usługi zostanie umieszczony w ścieżce: `/etc/systemd/system/multi-user.target.wants`, co odpowiada "dawnemu" poziomowi (runlevel3) multi-user bez grafiki.

Poniżej przedstawiono znaczenie poszczególnych "targetów" oraz ich odpowiedniki (runlevel) znane z SysVinit:

Zamknięcie systemu: `poweroff.target` -> odpowiednik runlevel0 w SysVinit

Wiersz poleceń w trybie rescue(jeden użytkownik): `rescue.target` -> runlevel1

Poziom wielu użytkowników (multi-user) bez grafiki: `multi-user.target` -> runlevel2 (wielu użytkowników bez usług sieciowych), runlevel3 (wielu użytkowników z usługami sieciowymi), runlevel4 (nieużywany)

Poziom wielu użytkowników (multi-user) z interfejsem graficznym: `graphical.target` -> runlevel5

Zamknięcie, restart systemu: `reboot.target` -> runlevel6