

## Service manager

**systemd** - wiele dystrybucji Linuksa używa go do zarządzania ustawieniami systemu i usługami. Organizuje on zadania w komponenty zwane unitami, a grupy jednostek w dependencies, których można używać do tworzenia zależności od innych usług systemowych i zasobów.

**systemctl** - komenda, która pozwala administratorowi systemu kontrolować **systemd** i menadżera usług. Może sprawdzać stan jednostek i celów, a także uruchamiać, zatrzymywać i rekonfigurować je.

### Ćwiczenie: Zarządzanie usługami z systemd i systemctl

Jak startować, zatrzymywać, resetować, przeładowywać pliki jednostek:

```
sudo systemctl start apache2.service
# uruchamianie usługi
sudo systemctl start apache2
# alternatywny, skrócony zapis

sudo systemctl status apache2.service
# sprawdzanie czy usługa jest uruchomiona
# w celu wyjścia z komendy klikamy `q`

sudo systemctl stop apache2.service
# zatrzymywanie usługi

sudo systemctl restart apache2.service
# restartowanie usługi

sudo systemctl reload apache2.service
# ponowne załadowywanie ustawień konfiguracyjnych, działa o ile aplikacja jest w
# stanie ponownie załadować konfigurację, bez restartowania

sudo systemctl reload-or-restart apache2.service
# ponowne załadowywanie ustawień konfiguracyjnych lub restartowanie usługi, w
# zależności czy program potrafi przeładować ustawienia konfiguracyjne w trakcie
# działania
```

Aktywowanie i dezaktywowanie plików jednostek oraz usług:

```
sudo systemctl enable apache2.service
# włącza usługę (przy rozruchu systemu)

sudo systemctl disable apache2.service
# wyłącza usługę (nie będzie ona uruchamiana przy rozruchu systemu)

ls /etc/systemd/system
# lokalizacja plików systemowych systemd
```

Sprawdzanie czy usługa jest aktywna i czy jest włączona:

```
sudo systemctl is-active apache2.service
# sprawdzanie, czy usługa jest obecnie aktywna
```

```
sudo systemctl is-enabled apache2.service
# sprawdzanie, czy usługa jest uruchamiana przy starcie systemu, czy nie
sudo systemctl is-failed apache2.service
# sprawdzanie, czy usługa nie jest w stanie awarii
```

Wyświetlanie jednostek, opcje wyświetlania:

```
systemctl list-units
# wyświetlanie listy aktywnych i załadowanych serwisów (jednostek usług/unitów)

systemctl list-units --all
# wyświetlanie listy aktywnych i nieaktywnych oraz niezaładowanych unit'ów
```

Filtrowanie wyników po stanie i typie jednostki

```
# Filtrowanie wyników po stanie jednostki
systemctl list-units --state=active
systemctl list-units --state=inactive
systemctl list-units --state=loaded

# Filtrowanie wyników po typie jednostki
systemctl list-units --type=service
```

Wyświetlanie zawartości plików unit

```
systemctl list-unit-files
# wyświetlanie plików jednostkowych (unit files)

systemctl cat apache2.service
# wyświetlanie pliku unit usługi apache2

systemctl list-dependencies
# drzewo zależności unit'ów

systemctl list-dependencies sshd.service
# wyświetlanie wszystkich zależności usługi sshd
systemctl list-dependencies sshd.service --all
# wyświetlanie wszystkich zależności usługi sshd

systemctl list-dependencies sshd.service --reverse
# wyświetlanie odwrotnych zależności usługi sshd

systemctl list-dependencies sshd.service --before
systemctl list-dependencies sshd.service --after

systemctl show sshd.service
# lista właściwości ustawionych dla danej jednostki (w formacie key-value)

systemctl show sshd.service -p Conflicts
# wyświetlanie konfliktów jednostki sshd
```

Maskowanie jednostki

```
sudo systemctl mask apache2.service
# maskowanie jednostki
sudo systemctl list-unit-files | grep apache2
# sprawdzanie, czy jednostka została zamaskowana
sudo systemctl start apache2.service
# próba uruchomienia zamaskowanej jednostki usługi
sudo systemctl unmask apache2.service
# usunięcie symlink'a /dev/null, które pozwala na ponowną aktywację lub uruchomienie jednostki
```

## Edycja plików jednostek

```
sudo systemctl edit apache2.service
# edycja pliku unit
sudo rm /etc/systemd/system/apache2.service
# usuwanie pliku unit

sudo systemctl daemon-reload
# przeładowywanie plików konfiguracyjnych całego systemd
```

## Cele (targets) - Opisują stan systemu lub synchronizacji

### Lokalizacja plików target

```
locate swap.target
# lokalizacja przykładowego pliku jednostki target
locate \*.target
# lokalizacja wszystkich plików target
```

### Sprawdzanie domyślnego celu i jego ustawianie

```
sudo systemctl get-default
sudo systemctl set-default target-file.target
```

### Wyświetlanie target'ów

```
sudo systemctl list-unit-files --type=target
# wyświetlenie wszystkich plików target

sudo systemctl list-units --type=target
# wyświetlenie wszystkich aktywnych plików target
```

### Modyfikacja target'ów

```
systemctl list-dependencies multi-user.target
# wyświetlenie zależności targetu multi-user.target
systemctl isolate multi-user.target
# odizolowanie targetu multi-user.target, tym samym przejdziemy do wiersza poleceń i zamkniemy środowisko graficzne
sudo systemctl set-default graphical.target
# przywrócenie poprzedniego target'u
sudo reboot
```

## Skróty dla ważnych zdarzeń w systemie

```
sudo systemctl rescue  
# przełączenie systemu w tryb ratunkowy  
  
sudo systemctl halt  
# zatrzymanie systemu  
  
sudo systemctl poweroff  
# zamknięcie systemu  
  
sudo systemctl reboot  
# ponowne uruchomienie systemu
```