

## Usługi w Linuksie

### Wprowadzenie do zarządzania usługami w systemie Linux

W tle Linuksa stale działa wiele usług, takich jak usługi sieciowe i systemowe. Usługi działające w systemie Linux są również znane jako daemony.

`systemd` jest pakietem narzędzi wykorzystywanych przez większość współczesnych dystrybucji Linuksa. Wszystkie zadania systemowe mogą być kontrolowane poprzez `systemd`. Proces może być uruchomiony lub zakończony za pomocą tego narzędzia. Można również za jego pomocą wyświetlić wszystkie włączone i wyłączone usługi wraz z informacjami o nich.

#### Listowanie usług za pomocą `systemctl`

Najprostszym sposobem na wylistowanie usług w Linuksie, gdy system wykorzystuje `systemd`, jest użycie polecenia `systemctl`, po którym następuje komenda `list-units`. Można dodać opcję `--type=service` w celu ograniczenia wyników tylko do usług.

```
systemctl list-units --type=service
```

Aby wyświetlić listę wszystkich usług, zarówno aktywnych jak i nieaktywnych, należy użyć polecenia `systemctl list-units`, a następnie opcji `--all`:

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

#### Uruchamianie usług

```
systemctl start [service_name]
```

#### Zatrzymywanie usług

```
systemctl stop [service_name]
```

#### Włączenie uruchamiania usługi podczas startu systemu

```
systemctl enable [service_name]
```

#### Wyłączenie uruchamiania usługi podczas startu systemu

```
systemctl disable [service_name]
```

#### Sprawdzanie statusu usługi

```
systemctl status [service_name]
```

#### Przeglądanie logów produkowanych przez usługi

Logi produkowane przez różne usługi są zapisywane przez `systemd` poprzez daemona `journald`. Aby zobaczyć logi, które zebrał daemon `journald`, użyj polecenia `journalctl`.

Kiedy komenda jest użyta samodzielnie, każdy wpis w dzienniku będzie wyświetlony w pagerze (zazwyczaj mniejszym) do przeglądania. Najstarsze wpisy będą na górze:

```
journalctl
```

Prawdopodobnie w dzienniku znajduje się spora liczba komunikatów (dziesiątki lub setki tysięcy linii). To pokazuje jak wiele logów znajduje się w bazie.

#### Wyświetlanie logów z bieżącego startu

Dodanie flagi `-b` do `journalctl` spowoduje filtrowanie logów, pokazując tylko dane załogowane od ostatniego restartu systemu:

```
journalctl -b
```

#### Przeglądanie logów produkowanych przez konkretną usługę

```
journalctl -u [nazwa_uslugi]
```

## Tworzenie własnych usług w Linuksie

Czasami chcemy, żeby napisana aplikacja była kontrolowana przez `systemd`. W niektórych przypadkach warto, aby skrypt był uruchamiany ponownie, gdy zostanie zabity z jakiegoś powodu. W takich przypadkach `systemd` jest dobrym rozwiązaniem.

1. Utwórz plik o nazwie `[nazwa_uslugi].service` w `/etc/systemd/system` używając następującego szablonu:

```
[Unit]
Description=Opis usługi

[Service]
User=nazwa_uzytkownika (np. root)
ExecStart=bezwzgledna_sciezka_do_skryptu (np. /home/student/Desktop/script.sh)
Restart=always

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

2. Dla projektów języka Python, które wykorzystują środowisko wirtualne:

```
[Unit]
Description=Opis usługi

[Service]
User=nazwa_uzytkownika (np. root)
ExecStart=/home/user/virtualenv/bin/python main.py
Restart=always
# zastąp /home/user/virtualenv/bin/python swoim virtualenv i main.py swoim skryptem

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

3. Nadaj prawa do wykonywania skryptu poleceniem `chmod +x nazwa_skryptu`

4. Przeladuj pliki usług, aby uwzględnić nową usługę:

```
sudo systemctl daemon-reload
```

5. Uruchom swoją usługę:

```
sudo systemctl start your-service.service
```

Przykładowy serwis i skrypt:

my\_service.service:

```
[Unit]
Description=Opis uslugi

[Service]
User=jakub
ExecStart=/usr/bin/python3 /home/jakub/Desktop/print_messages.py
Restart=always

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

print\_messages.py:

```
#!/usr/bin/env python3
import time

if __name__ == '__main__':
    while True:
        print('Hello world')
        time.sleep(1)
```

lub dla basha:

```
#!/bin/bash
while $(sleep 1);
do
    echo "hello world"
done
```

Tworząc własny skrypt, upewnij się, że umieściłeś w nim shebang line (pierwszą linijkę w powyższych przykładach).

### 🔥🔥 Zadanie 1 🔥🔥

Utwórz skrypt w Pythonie o nazwie `print_messages.py`. Zawartość pliku została przedstawiona w poprzedniej części instrukcji. Uruchom skrypt z poziomu terminala z dwóch różnych lokalizacji:

- lokalizacji, w której znajduje się utworzony skrypt,
- z lokalizacji `/etc/systemd/system`

### 🔥🔥 Zadanie 2 🔥🔥

Utwórz usługę `my_service.service`. Zawartość pliku została przedstawiona w poprzedniej części instrukcji. Zrealizuj uruchomienie usługi.

### 🔥🔥 Zadanie 3 🔥🔥

Napisz aplikację w języku Python, która korzysta z biblioteki `click`. CLI (command line interface) aplikacji powinien przyjąć ścieżkę do jakiegoś pliku i czas w sekundach (tutaj nazwijmy go `n`). Aplikacja powinna zapisywać losową liczbę wraz z aktualnym czasem do podanego pliku co `n` sekund.

Napisz plik usługi, który będzie uruchamiał Twoją aplikację. Spróbuj dodać kilka instrukcji `print` w swoim kodzie, abyś mógł sprawdzić czy logi są zbierane poprawnie. Włącz (enable) i uruchom (start) swoją nową usługę. Sprawdź status i logi. Spróbuj zrestartować swój system. Usługa powinna uruchomić się automatycznie.

Możesz chcieć stworzyć wirtualne środowisko (`venv`) i zainstalować w nim bibliotekę `click`.

Zakończ zadanie zatrzymując i wyłączając usługę. Usuń plik usługi z `/etc/systemd/system`. Zapisz swój kod, ponieważ wykorzystasz go na kolejnych zajęciach.

## Źródła

- <https://devconnected.com/how-to-list-services-on-linux/>
- <https://www.digitalocean.com/community/tutorials/how-to-use-journalctl-to-view-and-manipulate-systemd-logs>
- <https://www.shubhamdijpt.com/blog/how-to-create-a-systemd-service-in-linux/>