

Współczesne systemy komputerowe

Automatyzacja i planowanie

Demon cron

- **cron (anacron)** jest **demonem** (serwisem), który co minutę sprawdza i uruchamia zadania zdefiniowane w plikach **crontab**.
- Można definiować dwa rodzaje zadań: **systemowe** i **użytkownika**.
- Zadania systemowe są zdefiniowane w pliku **/etc/crontab**.

```
root@debian:~# egrep ^[^#] /etc/crontab
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
17 * * * *root cd / && run-parts --report /etc/cron.hourly
25 6 * * *root test -x /usr/sbin/anacron || { cd / && run-parts --report
    /etc/cron.daily; }
47 6 * * 7root test -x /usr/sbin/anacron || { cd / && run-parts --report
    /etc/cron.weekly; }
52 6 1 * *root test -x /usr/sbin/anacron || { cd / && run-parts --report
    /etc/cron.monthly; }
```

- Po instalacji są zdefiniowane zadania które uruchamiają skrypty umieszczone w katalogach:
 - **/etc/cron.hourly**,
 - **/etc/cron.daily**,
 - **/etc/cron.weekly**,
 - **/etc/cron.monthly**.
- Zadania użytkowników są umieszczane w plikach **/var/spool/cron/tabs/user**.
- Do zarządzania zadaniami użytkownik może użyć polecenia **crontab**.

-e	utworzenie lub edycja pliku zadań (edytor nano lub vi)
-l	wyświetlenie bieżących zadań
-r	usunięcie wszystkich zadań

- Każda linia w pliku konfiguracyjnym definiuje jedno zadanie, pierwsze pięć kolumn to data i czas wykonania zadania, szósta to polecenie lub skrypt do wykonania.

pole	zakres
minuty	0-59
godziny	0-23
dni miesiąca	1-31
miesiące	0-12
dni tygodnia	0-7

- Zamiast wartości z zakresu można podać symbol *****, wtedy zadanie zostanie wykonane przy każdej zmianie daty lub czasu, zakresu lub kroku.

- Linia:

`*/10 8-17 * * 1-5 command`

definiuje wykonanie ***command*** od poniedziałku do piątku (***1-5***) co 10 minut (****/10***) pomiędzy godzinami 8⁰⁰ i 17⁰⁰ (***8-17***).

- Sprawdź zadania użytkownika **foo**.

```
foo@debian:~$ crontab -l
no crontab for foo
```

- Dodaj zadanie uruchamiane co minutę, być może przy pierwszym użyciu trzeba będzie zdefiniować edytor, (**nano** jest najłatwiejszy).

```
foo@debian:~$ crontab -e
no crontab for foo - using an empty one
Select an editor. To change later, run select-editor again.
 1. /bin/nano          <---- easiest
 2. /usr/bin/mcedit
 3. /usr/bin/vim.tiny

Choose 1-3 [1]: 1
```

- Dopisz do otwartego pliku linię:

```
* * * * * echo "=== cron ===" >> /home/foo/cron.log
```

sprawdź zadania.

```
foo@debian:~$ crontab -l
```

- Sprawdź działanie zadania poleceniem, poczekaj 2-3 minuty (aby zakończyć, wciśnij [CTRL+C]).

```
foo@debian:~$ tail -f ~/cron.log
```

- Usuń zadania.

```
foo@debian:~$ crontab -r
```

```
foo@debian:~$ crontab -l
no crontab for foo
```

Polecenie at

- Poleceniem **at** można definiować zadania mające wykonać się jeden raz o zaplanowanej porze.

atq	wyświetla zdefiniowane zadania
atrm <i>number</i>	usuwa zadanie <i>number</i>

- Zainstaluj pakiet **at**.

```
root@debian:~# apt update
```

```
root@debian:~# apt install at
```

- Sprawdź stan (powinien być uruchomiony: **active**), i ewentualnie uruchom demona **at**.

```
root@debian:~# systemctl status atd
● atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enab>
   Active: active (running) since Tue 2025-12-16 12:58:24 EST; 2min 19s ago
  Invocation: a0b69a9cd977487db56349462e9b4e39
     Docs: man:atd(8)
    Main PID: 3441 (atd)
       Tasks: 1 (limit: 2296)
      Memory: 484K (peak: 2.2M)
         CPU: 11ms
    CGroup: /system.slice/atd.service
            └─3441 /usr/sbin/atd -f
```

- Sprawdź datę i dodaj zadanie trzy minuty później (wpisywanie zakończ klawiszami [CTRL+D]).

```
root@debian:~# date
Tue Dec 16 01:03:17 PM EST 2025
```

```
root@debian:~# at 13:09
warning: commands will be executed using /bin/sh
at Tue Dec 16 13:09:00 2025
at> logger -p emerg "ała ma psa"
at> <EOT>
job 1 at Tue Dec 16 13:09:00 2025
```

- Sprawdź zadanie.

```
root@debian:~# atq
1 Tue Dec 16 13:09:00 2025 a root
```

- Zaczekaj 2-3 minuty, zadanie powinno się wykonać.

- Dodaj dwa kolejne zadania.

```
root@debian:~# at noon tomorrow
warning: commands will be executed using /bin/sh
at Wed Dec 17 12:00:00 2025
at> logger "ała mie lubi kotow"
at> <EOT>
job 2 at Wed Dec 17 12:00:00 2025
```

```
root@debian:~# at 14:00 tomorrow
warning: commands will be executed using /bin/sh
at Wed Dec 17 14:00:00 2025
at> logger "wiktor wektor"
at> <EOT>
job 3 at Wed Dec 17 14:00:00 2025
```

- Sprawdź i usuń zadania.

```
root@debian:~# atq
2 Wed Dec 17 12:00:00 2025 a root
3 Wed Dec 17 14:00:00 2025 a root
```

```
| root@debian:~# atrm 2
```

```
| root@debian:~# atrm 3
```

```
| root@debian:~# atq
```