

Współczesne systemy komputerowe

Automatyzacja

1. Démon cron.

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

```
suse:~ # cat /etc/crontab
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root
#
# check scripts in cron.hourly, cron.daily, cron.weekly, and cron.monthly
#
-*/15 * * * * root test -x /usr/lib/cron/run-crons && /usr/lib/cron/run-crons
>/dev/null 2>&1
```

- Po instalacji jest zdefiniowane jedno zadanie (skrypt: */usr/lib/cron/run-crons*), który uruchamia 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 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, zakres lub krok;
- 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).

- Dodaj zadanie uruchamiane co minutę jako zwykły użytkownik:

```
* * * * * finger >> ~/users.log
```

sprawdź zadania;

```
suse:~ # crontab -e
no crontab for user - using an empty one
crontab: installing new crontab
user@suse:~> crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.XXXXdKE6yj installed on Tue Apr  2 00:29:59 2013)
# (Cron version V5.0 -- $Id: crontab.c,v 1.12 2004/01/23 18:56:42 vixie Exp $)
* * * * * finger >> ~/users.log

user@suse:~>
```

- Sprawdź działanie zadania poleceniem **tail -f ~/users.log**, aby zakończyć, wciśnij [CTRL C];
- Usuń zadania;

```
user@suse:~> crontab -r
user@suse:~> crontab -l
no crontab for user
```

2. 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>

- Sprawdź stan, i ewentualnie uruchom demona **at**;

```
user@suse:~> su -
Password:
suse:~ # rcatd status
Checking for at daemon: unused
suse:~ # rcatd start
Starting service at daemon done
suse:~ # rcatd status
Checking for at daemon: running
```

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

```
suse:~ # date
Tue Apr  2 15:15:05 CEST 2013
suse:~ # at 15:18
warning: commands will be executed using /bin/sh
```

```
at> finger >> /var/log/messages
at> <EOT>
```

- Sprawdź zadanie;

```
job 1 at 2013-04-02 15:18
suse:~ # atq
1      2013-04-02 15:18 a root
```

- Sprawdź wykonanie zadania (trzy minuty później);

```
suse:~ # date
Tue Apr  2 15:18:13 CEST 2013
suse:~ # tail -n 5 /var/log/messages
Apr  2 15:01:29 suse -- MARK --
Login      Name          Tty      Idle   Login Time   Where
user       user          *:0      -      Sun 10:06
user       user          pts/0    1      Sun 10:12 :0.0
user       user          pts/1    18:10  Sun 10:13 :0.0
```

- Dodaj dwa kolejne zadania;

```
suse:~ # at noon tomorrow
warning: commands will be executed using /bin/sh
at> date >> /var/log/messages
at> <EOT>
job 2 at 2013-04-03 12:00
suse:~ # at 14:00 tomorrow
warning: commands will be executed using /bin/sh
at> date >> /var/log/messages
at> <EOT>
job 3 at 2013-04-03 14:00
```

- Sprawdź i usuń zadania;

```
suse:~ # atq
3      2013-04-03 14:00 a root
2      2013-04-03 12:00 a root
suse:~ # atrm 2
suse:~ # atrm 3
suse:~ # atq
```