

Dzień z życia administratora: Fcron

Jak w zegarku

Usługa cron uruchamia zadania o określonym czasie. To stare, dobre narzędzie działa bez zarzutu – niemal jak szwajcarski zegarek – można mieć pewność, że o określonej godzinie, bez względu na okoliczności, zadanie zostanie wykonane – niestety okazuje się, że nie zawsze jest to dobre.

CHARLY KÜHNAST



Rutynowe zadania, takie jak archiwizacja plików dziennika, usuwanie plików tymczasowych czy też automatyczne wykonywanie kopii zapasowych, to tradycyjne zadania dla demona cron. Cron działa znakomicie, dopóki nie dojdzie do sytuacji, w której jedno zadanie przedłuża się, a należy już uruchomić kolejne. Nie raz w przeszłości myślałem o tym, jak bardzo przydałby się program, który będzie działał inteligentnie w takich sytuacjach. Wreszcie takie narzędzie znalazłem – program Fcron [1].

Fcron w obecnej wersji 2.0.1 osiągnął status aplikacji stabilnej, jego autorem jest Thibault Godouet, który chce, aby jego program stał się konkurentem znanego wszystkim programowi vixie cron. W tym celu Thibault wyposażył swój program w kilka niespotykanych wcześniej funkcji. Instalacja programu jest trywialna – po rozpakowaniu archiwum liczącego 142 Kb, wykonujemy polecenia:

```
./configure && make && \nmake install
```

Jeśli sobie tego życzysz, program instalacyjny może też skopiować odpowiedni skrypt startowy do automatycznego uruchamiania demona. Analogicznie do nazewnictwa cron-a, fcron używa pliku fcrontab do przechowywania tabeli zadań. W celu edycji tego pliku (czyli listy zadań) należy wykonać polecenie

fcrontab -e. To jednak już ostatnie podobieństwo do cron-a. Mimo że struktura i składnia fcrontab jest oparta na crontab, są jednak znaczące różnice.

Na początek nakazmy fcron wykonać to samo, co robi zazwyczaj cron: uruchamiać polecenie /bin/date na pięć minut przed pełną godziną. fcrontab powinien zawierać poniższy wpis:

```
5 0 * * * /bin/date
```

Ciekawą cechą fcron-a jest możliwość uruchamiania poleceń w odniesieniu do czasu pracy serwera (uptime), założmy, że chcemy uruchamiać polecenie /bin/date co dwie i pół godziny, licząc od chwili uruchomienia systemu:

```
@ 2h30m /bin/date
```

Bądź uprzejmy!

Dla programu, który generuje duże obciążenie systemu, można zmienić priorytet zadania poleceniem nice:

```
&nice(10) 0 1 * * * /usr/local \n/shellscripts/MyBackupScript.sh
```

Poniższy wpis uruchamia polecenie co godzinę między godzinami 14.00 a 10.00.

```
0 14-22 * * * /bin/date
```

Jeśli polecenie ma być uruchamiane co dwie godziny, musimy zmienić wpis następująco:

```
0 14-22/2 * * * /bin/date
```

Fcron obsługuje również wyjątki: jeśli chcesz uruchamiać polecenie w każdy dzień tygodnia za wyjątkiem soboty, musisz wprowadzić tylko drobną zmianę:

```
0 14-22/2 * * *-0 /bin/date
```

Innymi słowy, fcron traktuje elementy rozpoczynające się od tyldy (~) jak wyjątki. Możliwość sprawdzania obciążenia systemu jest jedną z najlepszych cech fcron-a. Poniższy przykład uruchamia polecenie o nazwie /bin/heavyjob/ raz na godzinę – ale tylko wtedy, gdy średnie obciążenie systemu (loadavg) w okresie pięciu minut jest poniżej 1.5:

```
@1avg5(1.5) 1h /bin/heavyjob
```

Bardzo przydatna jest pomoc w postaci podręcznika dokumentacji systemowej (man), który szczegółowo opisuje poszczególne opcje fcron-a, wraz z przykładami. Nie wiem jak będzie z Wami, ale dla mnie stary dobry program cron przeszedł do historii.

INFO

[1] Fcron: <http://fcron.free.fr>