Konfigurowanie i zastosowanie systemu Cron

Grzegorz Jacek Nalepa

24.8.2000, Kraków, *Revision* : 1.4

Streszczenie

Przedstawiony w artykule system Cron jest jednym z podstawowych narzędzi automatyzacji pracy systemu GNU/Linux czy Unix. System może przynieść nieocenione korzyści w ręku doświadczonego administratora. Jest on również przydatny dla zaawansowanych użytkowników systemu. Głównym zadaniem systemu jest cykliczne uruchamianie określonych zadań w podanym terminie. Artykuł prezentuje architekturę systemu Cron, a także prezentuje na przykładach jego konfigurowanie. Zaprezentowane są również narzędzia wspomagające pracę z systemem.

Spis treści

[1]Tekst ukazał się w: *Magazynie NetForum*, nr 10/2000, wydawanym przez Lupus.

[2]Kontakt z autorem: mail:gjn@agh.edu.pl

[3]Tytuł angielski: *Using and configuring Cron*

1. Wprowadzenie

Używanie i administrowanie systemem o tak dużych możliwościach jak GNU/Linux może być niejednokrotnie zadaniem złożonym. Istnieje jednak szereg standardowych narzędzi, które to zadanie niezwykle ułatwiają, pomagając w wykonywaniu rutynowych zadań. Takim narzędziem jest niewątpliwie system Cron, autorstwa Paula Vixie, będący jednym z podstawowych elementów systemu.

System Cron składa się z dwóch podstawowych części: serwera i klienta. Serwer, czyli demon *crond*, pracuje cały czas w systemie i na podstawie systemowych plików konfiguracyjnych oraz konfiguracji użytkowników cyklicznie uruchamia zadania. Program klient nosi nazwę *crontab* i służy do edycji pliku konfiguracyjnego użytkownika.

2. Konfigurowanie

Każdy użytkownik w systemie GNU/Linux ma dostęp do systemu Cron poprzez program *crontab*. Program umożliwia zakładanie, edycję i wyświetlanie pliku konfiguracyjnego dla danego użytkownika. Opis programu *crontab* można znaleźć w podręczniku `crontab(1)`. Polecenie korzysta z zewnętrznego edytora do edycji pliku. Uruchamiany edytor można wybrać przy pomocy zmiennej VISUAL lub EDITOR. Przeważnie domyślnym edytorem jest VI. Podstawowe wywołania `crontab` to:

- `crontab -e` – edycja pliku,
- `crontab -l` – wyświetlenie pliku,
- `crontab -r` – usunięcie pliku,
- `crontab -u user {-e|-l-r}` – edycja pliku zadanego użytkownika – wywołanie tylko dla administratora,
- `crontab plik` – użycie gotowego `pliku` jako `crontab`.

Całość konfiguracji dla pojedynczego użytkownika znajduje się w pliku `crontab`, którego dokładna składnia jest opisana w podręczniku *crontab(5)*. Plik zawiera szereg zadań (zleceń) dla systemu Cron, gdzie przez zadanie można rozumieć polecenie, które ma być wykonywane w dokładnie zdefiniowanym cyklu (na przykład każdego dnia). Każdy użytkownik może założyć jeden taki plik, i może się w nim znajdować wiele zadań, każde w osobnej linii. Jedna linia opisuje jedno zadanie i składa się z sześciu pól, rozdzielonych białymi spacjami. Pola w każdej linii oznaczają kolejno: minutę, godzinę dzień miesiąca, miesiąc, dzień tygodnia, ostatnie to nazwa polecenia do wykonania. Gwiazdka w miejscu wartości oznacza: „dla każdej wartości pola". W danym polu można podawać kilka wartości rozdzielonych przecinkami, natomiast użycie znaku „–" pozwala na podanie zakresu wartości. Podanie po zakresie znaku „/" i kolejnej wartości umożliwia podanie kroku w obrębie zakresu.

Opisane powyżej konfigurowanie Crona najlepiej zademonstrować na przykładach. Konfigurowanie należy rozpocząć przy pomocy polecenia `crontab -e`, które powoduje pojawienie się edytora z nowym, pustym plikiem. Jeżeli użytkownik wcześniej miał swój plik `crontab`, to edytor wyświetli jego zawartość. Ważne jest, aby po zakończeniu edycji zapisać plik pod domyślną nazwą proponowaną przez edytor, czyli w przypadku VI wyjść z edytora przez polecenie `:wq`. Edycję pliku można rozpocząć od pierwszej linii. Przykładowo, jeżeli chce się uruchamiać polecenie `who` co godzinę, linia w `crontab` może wyglądać następująco:

```
0 * * * *   who
```

Jak widać w polach oznaczających czas wystarczy podać jedną wartość. Jest ona podana w polu minut, a nie jak można by się spodziewać w polu godzin i oznacza w której minucie godziny będzie uruchamiane polecenie. Tak więc całość należy czytać następująco: „w pierwszej minucie (czyli zero minut po rozpoczęciu godziny), o każdej godzinie, każdego dnia miesiąca, w każdym miesiącu, każdego dnia tygodnia, uruchom „who". W praktyce oznacza to tyle, co „w pierwszej minucie godziny uruchom „who". Tak więc, jeżeli dopisało się tę linijkę do pliku `crontab` o godzinie 11:48, to polecenie `who` będzie uruchomione o godzinie 12:00, 13:00, 14:00 itd. (polecenie będzie uruchamiane codziennie). Tak więc aby uruchamiać polecenie co godzinę wystarczy podać dowolną wartość w polu minut.

Po tak gruntownym omówienie pierwszego przykładu, pokazanie jak uruchamiać polecenie raz dziennie będzie znacznie prostsze. Należy w tym celu podać określoną godzinę i minutę. Linia

```
0 8 * * *   who
```

Spowoduje uruchomienie `who` każdego dnia o 8:00.

Dni tygodnia są numerowane od 0 do 7, gdzie zarówno 0 i 7 są interpretowane jako niedziela. Dzięki temu zwolennicy poglądu, że tydzień rozpoczyna się niedzielę, mogą liczyć dni tygodnia od 0 do 6, a ci, którzy uważają, że w poniedziałek od 1 do 7. Dni tygodnia (podobnie jak dni miesiąca) mogą być podawane przy pomocy trzy literowych skrótów angielskich, np. mon, tue (jan, feb).

Oprócz konkretnych, pojedynczych wartości pól można podawać kilka wartości rozdzielonych przecinkami. Można również podać zakres wartości, czyli dwie liczby rozdzielone znakiem „`--`". Domyślnym krokiem w przypadku zakresu jest jeden, lecz można również podać inny. W poniższym przykładzie:

```
0,20,40 8-16 * * *   who
```

Polecenie `who` będzie uruchamiane codziennie, 3 razy na godzinę, ale tylko między 8 a 16.

Podając oprócz minuty i godziny dzień miesiąca powodujemy uruchamianie polecenia co miesiąc w zadanym dniu o zadanej godzinie. Dodając do tego określony miesiąc powodujemy uruchamianie raz na rok. Jak widać różnych kombinacji pól jest dość wiele i choć nie ma sensu ich tu szczegółowo omawiać warto rozumieć ich interpretację. Nietrywialnym przykładem może być podanie tylko pól minut i dnia miesiąca:

```
10 * 5 * * who
```

W tym przypadku `who` będzie uruchamiane co godzinę każdego piątego dnia miesiąca. Przykład:

```
20-40 0,8,16 1-30/2 * *   who
```

oznacza uruchamianie polecenia o godzinach od 0:20 do 0:40 co minutę, następnie od 8:20 do 8:40 co minutę (podobnie dla 16), co dwa dni. Wart odnotowania jest fakt, że pola dzień miesiąca i tygodnia łączą się, to znaczy podanie:

```
30 6 10 * 1   who
```

uruchamia polecenie 10. dnia miesiąca o 6:30 oraz w każdy piątek.

Na koniec przykładów warto się zastanowić, jaki efekt przyniesie podanie 5 gwiazdek w polu czasu. Oznacza ono mianowicie uruchamianie polecenia co minutę.

Problem, który nie był dotychczas omawiany dotyczy tego, jak są uruchamiane polecenia zlecane Cronowi. Początkujący użytkownicy po zainstalowaniu nowego `crontab`a niejednokrotnie oczekują pojawienia się efektów działania, na przykład uruchomienia `who`, „na ekranie", w

czasie ich otwartej sesji. Nic takiego się nie dzieje. Trzeba pamiętać, że polecenia uruchamiane przez Crona pracują w trybie nieinteraktywnym i nie są związane z aktualnie otwartymi sesjami pracy użytkowników. Oznacza to, że uruchamianie programów pracujących interaktywnie, które oczekują wprowadzania informacji z terminala, nie ma sensu. Wyniki pracy programów uruchamianych przez Cron, a dokładniej to, co jest wysyłane na ich standardowe wyjście, są przesyłane do użytkownika pocztą elektroniczną. Przykład takiego e-maila jest pokazany poniżej:

```
Date: Thu, 24 Aug 2000 08:25:00 +0200 (CEST)
From: root (Cron Daemon)
To: gjn
Subject: Cron <gjn@enterprise> who
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/home/gjn>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=gjn>

gjn   :0       Aug 24 06:30 (console)
gjn   ttyp0    Aug 24 06:30 (:0.0)
gjn   ttyp1    Aug 24 06:30 (:0.0)
root  ttyp5    Aug 24 06:39 (localhost)
```

3. Cron z perspektywy administratora

To, co dotychczas zostało powiedziane na temat Crona i polecenia `crontab` dotyczy każdego użytkownika systemu. Natomiast dla administratora jest również istotny sposób działania samego Crona, jego zaawansowana konfiguracja a także sposoby wykorzystania.

Jak już powiedziano, Cron składa się z klienta – omawianego już programu `crontab` – oraz serwera, wspomnianego demona `crond`. Klient może być uruchamiany przez każdego użytkownika systemu do edycji jego własnego pliku. Dodatkowo administrator ma możliwość edycji pliku dowolnego użytkownika, w tym użytkowników systemowych, podając w poleceniu opcję `-u użytkownik`. Wszystkie pliki `crontab` znajdują się w jednym katalogu, przeważnie `/var/spool/cron/crontabs`. Każdemu użytkownikowi, który założył własny `crontab` odpowiada plik o takiej nazwie jak nazwa jego konta. Należy w tym miejscu wyjaśnić, że po uruchomieniu polecenia `crontab`, nie edytuje się plików z tego katalogu, lecz ich kopie, każdorazowo wykonywane przez `crontab`. Po dokonaniu edycji kopii `crontab` sprawdza składnię pliku i sygnalizuje błąd. Dopiero pozbawiona błędów składniowych nowa wersja `crontab`a jest kopiowana do pliku w katalogu `/var/spool/cron/crontabs`. Przykładowo, zawartość pliku z `/var/spool/cron/crontabs/gjn` związanego z podanym powyżej przykładem e-maila jest następująca:

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.XXXXa00893 installed on Thu Aug 24 08:23:25 2000)
# (Cron version -- $Id: Cron.tex,v 1.4 2002/06/08 10:22:26 gjn Exp $)
25 8 * * * who
```

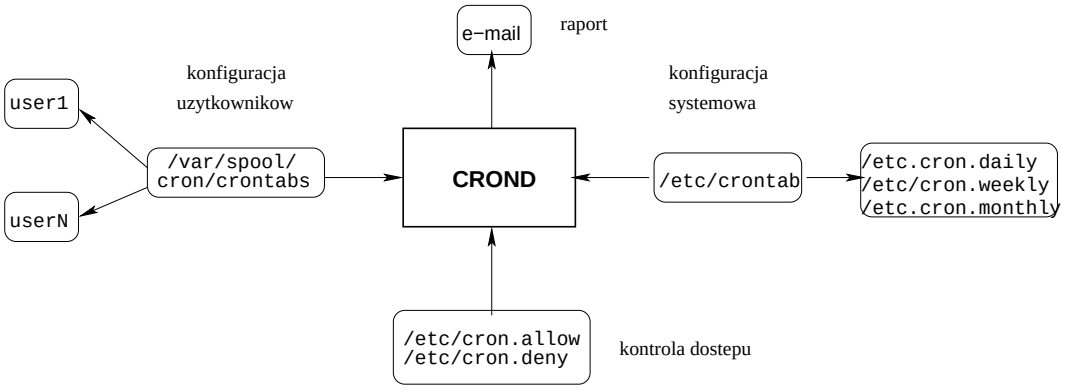
Zawiera informacje o tym, kiedy był modyfikowany plik i jaka była nazwa kopii w `/tmp`.

Oprócz plików użytkowników system Cron ma jeden systemowy plik konfiguracyjny, `/etc/crontab`, który może być edytowany bezpośrednio przez administratora. Typowy przykład takiego pliku jest pokazany poniżej:

```
# /etc/crontab: system-wide crontab
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# m h dom mon dow user  command
25 6    * * *   root    run-parts --report /etc/cron.daily
47 6    * * 7   root    run-parts --report /etc/cron.weekly
52 6    1 * *   root    run-parts --report /etc/cron.monthly
```

Zwraca uwagę pojawienie się po polach określających czas dodatkowego, siódmego, pola związanego z użytkownikiem. Pozwala ono administratorowi na określenie z jakimi uprawnieniami będzie uruchamiane podane na końcu polecenie. W tym pliku można umieszczać listę zadań administracyjnych dla Crona, związanych z utrzymaniem systemu. Jednak od dawna stosuje się bardziej elastyczny mechanizm, pokazany powyżej. Plik `/etc/crontab` zawiera zadania związane z częstotliwością uruchamiania, standardowo uruchamiane każdego dnia, tygodnia i miesiąca. Powyższy plik pochodzący z systemu Debian/GNU pokazuje taką konfigurację. Polecenie `run-parts` wykonuje wszystkie polecenia wykonywalne z podanego katalogu. Różne pakiety oprogramowania umieszczają w trakcie instalowania własne zadania dla Crona w postaci wykonywalnych plików (najczęściej skryptów w Sh lub Perlu) w katalogach `/etc/cron.*`. Te pliki pozwalają na automatyczne generowanie raportów z pracy programu, lub rotację plików rejestrowych.

Zarys omówionej architektury Crona jest pokazany na Rysunku 1.



Rysunek 1: Architektura systemu Cron

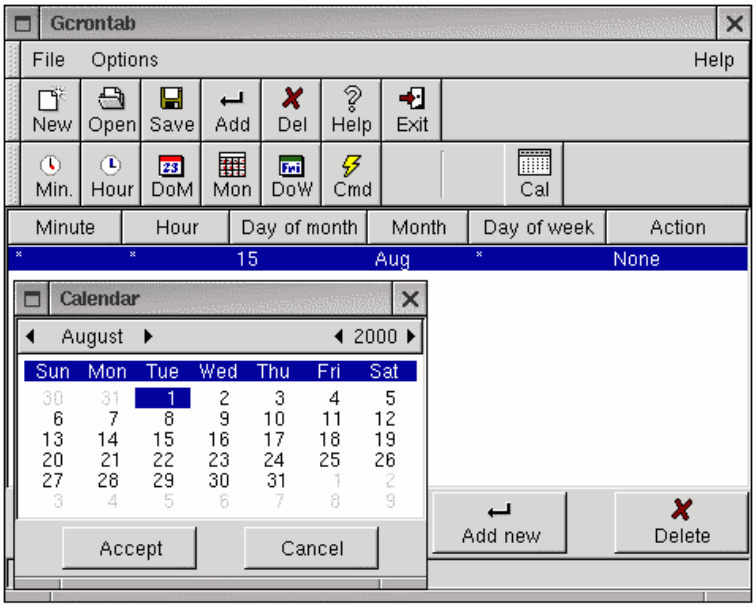
Należy powiedzieć również kilka słów na temat zagadnień związanych z bezpieczeństwem systemu. Po pierwsze program `crontab` pracuje z uprawnieniami administratora (ang. *suid root*). Oznacza to, że może być teoretycznie użyty w przypadku różnego rodzaju włamań. Po drugie, administrator może ograniczyć dostęp użytkowników do systemu Cron. Kontrola dostępu odbywa się przez dopisanie reguł *allow/deny* dla grup użytkowników w plikach `/etc/cron.allow, /etc/cron.deny`. Jest to szczegółowo opisane w podręczniku. Na koniec trzeba pamiętać o właściwych prawach dostępu do pliku `/etc/crontab` i plikach w katalogach `/var/spool/cron`, oraz `/etc/cron.*`.

4. Zastosowania i interfejsy

Zastosowania systemu są rozliczne i są właściwie ograniczone wyłącznie wyobraźnią administratora i poniekąd twórców dystrybucji GNU/Linuxa. Najczęściej wykorzystuje się Crona do rotacji plików rejestrowych, czy wstępnej ich analizy. Może być również wykorzystywany

do cyklicznego sprawdzania systemu plików, na przykład ilości wolnego miejsca, czyszczenia katalogu `/tmp`, czy sprawdzania spójności plików systemowych. Wielu administratorów wykorzystuje Crona również do tworzenia kopii zapasowych. W praktyce każda czynność, która jest wykonywana systematycznie według pewnego określonego algorytmu może być wykonywana przez system Cron.

W związku z rozlicznymi zastosowaniami Cron powstał szereg programów wspomagających pracę z nim, w tym zapewniających graficzny interfejs użytkownika. W takie narzędzie są wyposażone zintegrowane środowiska graficzne, takie jak GNOME czy KDE. Oprócz tego istnieją programy nie związane z tymi środowiskami. Przykładem takiego narzędzia jest *gcrontab* dostępny pod adresem `http://www.arquired.es/users/aldelgado/proy/gcrontab`. Interfejs *gcrontaba* jest oparty o popularną bibliotekę Gtk+ i zapewnia funkcjonalność systemowego `crontab`. Przykład pracy z programem jest pokazany na Rysunku 2. Choć tego typu interfejsy



Rysunek 2: Praca z gcrontabem

są przydatne dla początkujących, to wydaje się, że standardowe narzędzia oparte o interfejs znakowy są bardziej uniwersalne i dużo wydajniejsze.

5. Podsumowanie

System Cron jest jednym z podstawowych elementów współczesnych systemów GNU/Linux i Unix. Wraz z narzędziami takimi jak Syslog stanowi podstawę do automatycznego monitorowania i nadzoru nad systemem. W związku z tym, każdy nawet średnio zaawansowany administrator powinien rozumieć zasadę jego działania i umieć się nim posługiwać.