

Crontab

at

Jednorazowe zaplanowanie zadania – program at.

Jeśli potrzebujemy wykonać jakieś zadanie, czy też polecenie jednorazowo za jakiś czas (np. o określonej godzinie zamknąć automatycznie system, nawet wtedy, gdy nas przy nim nie ma), z pomocą przychodzi polecenie „at”. Polecenia at używamy na kilka sposobów. Najczęściej używany jest poniższy schemat:

1. Wywołujemy polecenie at z określeniem czasu wykonania, np. at 15:00
2. Pokaże się wiersz zachęty at> podajemy polecenie do wykonania np. poweroff
3. Naciskamy kombinację klawiszy Ctrl+d, aby wyjść z trybu wprowadzania zadania.

Wylistowanie zaplanowanych zadań : at -l, lub atq

Usuwanie zadań: atrm nr_zadania, lub at -r nr_zadania (nr znajduje się w pierwszej kolumnie która jest widoczna podczas listowania zadań).

Czas możemy podawać na wiele sposobów, np.

at midnight (o północy)

at 3pm tomorrow (jutro o 15:00)

at 15:00 + 4 days (za 4 dni o godzinie 15.00)

at 6am May 20 (20 maja o godz. 6 rano)

Problemy

Na początku korzystania z polecenia at możemy napotkać różne problemy.

Np. jeżeli zobaczymy komunikat „Can't open /var/run/atd.pid to signal atd. No atd running?” należy uruchomić demona atd, w następujący sposób:

```
/etc/init.d/atd start
```

Aby demon atd uruchamiał się przy starcie systemu wydaj polecenie: **chkconfig --level 345**

atd on

ĆWICZENIA – at.

Dodaj zadanie automatycznego zamknięcia systemu. System ma się zamknąć automatycznie o godz. 15.30.

- 1)Zaloguj się jako root,
- 2)Wydaj polecenie: at 15:30,
- 3) Wydaj polecenie: poweroff i wciśnij enter,
- 4) Wciśnij Ctrl+d, aby zakończyć wprowadzanie,

5) Sprawdź poleceniem `at -l`, lub `atq`, czy dodane zadanie zostało wyświetlone.

Dodaj zadanie automatycznego wykonania skryptu o nazwie `zamknij.sh`, w którym znajduje się polecenie zapisania użytkowników do pliku `кто_jest.txt`, którzy są zalogowani w systemie (polecenie `who`), oraz zamknięcia systemu (polecenie `poweroff`). Zaplanuj wykonanie zadania na godz. 16.30.

1) Należy napisać skrypt wykonujący powyższe zadania, oraz nadać prawo wykonania,

2) Należy zaplanować wykonanie zadania za pomocą polecenia `at`.

Crontab.

Poznane na poprzednich slajdach polecenie `at` służy do planowania i wykonywania zadań jednorazowych. W przypadku zadań wykonywanych cyklicznie korzystamy z innego narzędzia systemu LINUX, a mianowicie z `crona`. Jak nadmieniono powyżej narzędzie to służy do planowania cyklicznie wykonywanych zadań np. codziennego wykonywania kopii zapasowej plików i archiwizacji, zamykania systemu o określonej godzinie itp. Crontab jest odpowiednikiem harmonogramu zadań z Windows.

Uruchomienie „ręczne” :

`/etc/init.d/cron start`

Zatrzymanie :

`/etc/init.d/cron stop`

Zazwyczaj crontab uruchamia się domyślnie podczas startu systemu, więc jego „ręczne” startowanie jest zbyteczne.

Składnia polecenia `crontab`:

`crontab [-l], [-r], [-e]`

`l` – listuje zadania

`r` – usuwa zadania

`e` – tworzy lub otwiera wcześniej utworzony plik z zadaniami użytkownika

Zabronienie używania `crona` `/etc/cron.deny`. Dopisz do tego pliku dowolnego użytkownika i spróbuj wykonać polecenie `crontab` np. `crontab -e`. Zobaczysz:

```
linux-3kiy:/etc/init.d # su pkania
sh-4.1$ crontab -e
You (pkania) are not allowed to use this program (crontab)
See crontab(1) for more information
sh-4.1$ █
```

Struktura pliku z zadaniami:

min godz dzien_miesiaca miesiac dzien_tyg zadanie

Dane czasu do uruchomienia podawane są w jednym wierszu w pięciu kolumnach, dane te rozdzielone są przecinkami, jeśli mamy do czynienia z zakresem to wtedy występuje myślnik oznaczający zakres, gwiazdka oznacza wszystkie dostępne wartości. Po danych czasu uruchomienia podajemy zadanie do wykonania. Jeżeli zadań ma być kilka, rozdzielamy je znakiem „;”

Zakres poszczególnych pól:

Minuty : 0-59

Godziny : 0-23

Dni_miesiąca : 1-31

Miesiące : 1-12

Dni tygodnia 0 (niedziela) – 6 (sobota)

ĆWICZENIA – CRONTAB.

Dodaj zadanie automatycznego zamykania systemu. System ma się zamykać codziennie o godz. 14.30.

1)Zaloguj się jako root,

2)Wydaj polecenie: crontab –e

3) W otwartym pliku dopisz linię:

30 14 * * * /sbin/halt (pamiętaj o podaniu pełnej ścieżki, w przeciwnym wypadku polecenie nie zadziała !!!)

4) Zapisz plik

5) Sprawdź poleceniem crontab –l, czy dodane zadanie zostanie wyświetlone.

Dodaj zadanie automatycznego dopisywania do pliku test_crontab.txt, następującego tekstu „to jest wpis z crontaba” co dziesięć minut, w godzinach od 9-14.

- 1)Zaloguj się jako root,
- 2)Wydaj polecenie: crontab –e
- 3) W otwartym pliku dopisz linię:
0,10,20,30,40,50 9-14 * * * echo „to jest wpis z crontaba” >> test_crontab.txt
(wpisz 0,10,20 ... itd. Można zastąpić również wpisem **0-59/10**)
- 4) Zapisz plik
- 5) Sprawdź poleceniem crontab –l, czy dodane zadanie zostanie wyświetlone.

Co dwie godziny dopisuj do pliku kto_jest.txt, wynik polecenia „who” w godzinach od 10.10 do 18.10.

- 1)Zaloguj się jako root,
- 2)Wydaj polecenie: crontab –e
- 3) W otwartym pliku dopisz linię:
10 10-18/2 * * * who >> kto_jest.txt
- 4) Zapisz plik
- 5) Sprawdź poleceniem crontab –l, czy dodane zadanie zostanie wyświetlone.

Stwórz skrypt o nazwie skrypt.sh o następującej treści:

```
#!/sbin/sh  
echo `who` >> kto_byl.txt  
/sbin/poweroff
```

(nie zapomnij o całej ścieżce, w przeciwnym razie polecenie nie zostanie wykonane !!!)

Następnie zaplanuj jego wykonanie o godzinie 15.00

- 1)Zaloguj się jako root,
- 2)Wydaj polecenie: crontab –e
- 3) W otwartym pliku dopisz linię:
0 15 * * * /root/skrypt.sh
- 4) Zapisz plik
- 5) Sprawdź poleceniem crontab –l, czy dodane zadanie zostanie wyświetlone.

Wykonajmy teraz to samo ćwiczenie co na poprzednim slajdzie, ale bez użycia zewnętrznego skryptu. Obydwa zadania mają się znajdować bezpośrednio w cronie. Pamiętaj, aby oddzielić

zadania wykonywane bezpośrednio po sobie o zaplanowanej porze używamy znaku rozdzielającego „;”, (średnik).

Następnie zaplanuj jego wykonanie o godzinie 15.00

1) Zaloguj się jako root,

2) Wydadź polecenie: `crontab -e`

3) W otwartym pliku dopisz linię:

```
00 15 * * * echo `who` >> ~/kto_byl.txt; /sbin/poweroff
```

4) Zapisz plik

5) Sprawdź poleceniem `crontab -l`, czy dodane zadanie zostanie wyświetlone.

Crontab – definiowanie własnego pliku z zadaniami.

Nic nie szkodzi na przeszkodzie, aby zdefiniować własny plik z tablicami zadań dla crona.

Aby zrezygnować z domyślnego pliku zadań wykonujemy następujące kroki.

1) Tworzymy plik np. `vi ~/zadania_cron`

2) Wpisujemy planowane zadania, jak na poprzednich slajdach

3) Wydadźmy polecenie `crontab ~/zadania_cron` (czyli wywołujemy crontaba i po spacji wskazujemy nowy plik z tablicami zadań).

Jeżeli zmieniasz coś w tym pliku po jego zapisaniu wydaj ponownie polecenie `crontab ~/zadania_cron`