

Z braku dobrych odpowiedników na określenie czynności do wykonania, która w terminologii Unixowej zwana jest **task**, będziemy ją nazywać **zadanie**.

Nie zawsze dobrym rozwiązaniem jest uruchomienie programu w momencie kiedy siedzimy przed terminalem. Program do analizy dużej ilości danych, który zwykle potrzebuje 6-7 godzin, aby wyprodukować wyniki, do tego mocno obciąża komputer, lepiej uruchomić na noc, tak aby rano mieć gotowe wyniki. Na systemach, na których pracuje jednocześnie wiele osób, takie programy powinniśmy starać się uruchamiać w godzinach nocnych, gdy obciążenie systemu jest mniejsze i nie będzie to innym przeszkadzać. Drugi przykład, gdy chcemy wykonać jakiś program określonego dnia, o określonej godzinie, to pobieranie aktualizacji, o których wiemy, że pojawią w wyznaczonym, późniejszym terminie.

W Unixie system pozwalający planować zadania, ustawiać harmonogram zadań oraz wykonujący zadania w określonym terminie to **cron**.

Zadania do wykonania możemy podzielić na trzy kategorie:

1. pojedyncze programy, które chcemy wykonać jednorazowo w późniejszym terminie (at),
2. grupy programów, które potrzebujemy wykonać jako zestaw (batch),
3. programy, które wykonujemy cyklicznie, w regularnych odstępach czasu (crontab).

Obecnie coraz rzadziej mamy do czynienia z pierwszymi dwiema kategoriami. Do zaplanowania wykonania zadania w określonym terminie służy polecenie `at`. Polecenie to uruchamiamy podając termin wykonania zadania. Może to być konkretna data z godziną lub tekst w określonym formacie, opisujący taki termin, na przykład:

```
12:30 tuesday
next monday
midnight
now + 30 minutes
23:00 +2 months
23:59 dec 31, 2020
```

Po uruchomieniu polecenia `at` z podanym poprawnym terminem wykonania zadania pojawi się znak zachęty i możemy wpisywać polecenia do wykonania. Nie musi to być jeden konkretny program. Możemy tam wpisać wręcz cały, dłuższy skrypt z wieloma poleceniami do wykonania. Wpisywanie poleceń kończymy poprzez `Ctrl+D`, czyli znak końca transmisji `EOT` (End Of Transmission).

```
sirius$ at 23:59 dec 31, 2020
at> echo "Don't forget to open a champagne"
at> <EOT>
commands will be executed using /bin/bash
job 1609455543.a at Cz gr 31 23:59:00 20
```

Dodając zadania poleceniem `at` tworzymy kolejkę (queue) zadań do wykonania. Można ją obejrzeć przy pomocy `atq` albo `at -l`. Polecenie `atq` jest nieco bardziej wygadane i pokazuje więcej informacji:

```
sirius$ atq
```

Rank	Execution Date	Owner	Job	Queue	Job Name
1st	Dec 14, 2020 16:52	mariusz	1607961136.a	a	stdin
2nd	Dec 15, 2020 12:30	mariusz	1608031800.a	a	stdin
3rd	Dec 15, 2020 14:48	mariusz	1608040119.a	a	stdin
4th	Dec 31, 2020 23:59	mariusz	1609455542.a	a	stdin

```
5th Dec 31, 2020 23:59 mariusz 1609455543.a a stdin
6th Mar 14, 2021 23:59 mariusz 1615762740.a a stdin
```

Jako zwykły użytkownik zobaczymy tylko własne zadania. Administrator `root` może obejrzeć zadania wszystkich użytkowników. Z opcją `-c` program `atq` wyświetli zadania w kolejności ich tworzenia, nie wykonania. Natomiast z opcją `-n` wyświetli tylko ilość zadań w kolejce. Lista zadań wyświetlana przez `at -l` wygląda następująco:

```
sirius$ at -l
1608031800.a Wt gr 15 12:30:00 2020
1608040119.a Wt gr 15 14:48:39 2020
1607961136.a Pn gr 14 16:52:16 2020
1615762740.a N mrz 14 23:59:00 2021
1609455542.a Cz gr 31 23:59:02 2020
1609455543.a Cz gr 31 23:59:03 2020
```

Jak widać każde zadanie ma swój unikalny identyfikator. Do usunięcia zadania z kolejki służy `atrm` albo `at -r`. W obu przypadkach należy podać identyfikator zadania do usunięcia. Chyba, że zawołamy `atrm -a`, co usuwa wszystkie zadania.

```
sirius$ atrm 1608031800.a
1608031800.a: removed
```

Polecenie `batch` jest to w rzeczywistości program `at` z opcją `-b`. Po uruchomieniu, podobnie jak `at`, pozwala wprowadzać polecenia do wykonania. Będą one wykonywane równolegle bądź sekwencyjnie w zależności od ustawień w konfiguracji.

Konserwacja serwera wymaga regularnego wykonywania pewnych czynności, na przykład:

- tworzenia kopii zapasowych,
- rotacji dzienników,
- usuwania zbędnych plików tymczasowych,
- zmiany czasu z zimowego na letni i na odwrót.

To jest, według naszego podziału, trzecia kategoria zadań. Harmonogram zadań cyklicznych to w Unixie tak zwany **crontab**. Polecenie `crontab` służy do zarządzania harmonogramem zadań użytkownika. Z opcją `-l` wyświetla zawartość tablicy `crontab`, natomiast z opcją `-e` pozwala edytować tę tablicę.

Tablica `crontab` zawiera 6 kolumn:

1. minuta (0-59),
2. godzina (0-23),
3. dzień miesiąca (1-31),
4. miesiąc (1-12),
5. dzień tygodnia (0-6, 0 to niedziela),
6. ścieżka do programu i lista ewentualnych parametrów.

Każdy wiersz pliku `crontab` definiuje jedno zadanie. Linie rozpoczynające się od znaku `#` są ignorowane i służą do umieszczania komentarzy. Każde z pól 1-5 może zawierać konkretną wartość, zakres wartości rozdzielonych znakiem `-`, albo listę wartości rozdzielonych przecinkami. Na poziomie systemu `cron`, ze względów bezpieczeństwa lista katalogów w zmiennej `$PATH`, z których uruchamiane mogą być programy jest ograniczona do minimum. Należy o tym pamiętać pisząc skrypty uruchamiane z `crontab`. Należy albo odpowiednio zmodyfikować `$PATH`, albo wołać programy podając pełną ścieżkę.

Przykładowy plik crontab może wyglądać tak:

```
# minute hour day month weekday command
#
# Rotate logs
#
10 3 * * * /usr/sbin/logadm
#
# Adjust the real time clock if and when daylight savings time changes
#
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
#
# Backup
#
0 2 * * * /opt/bin/backup
0 5 * * 0 /opt/bin/backup-system
5 * * * * /opt/bin/backup-mysql
15 8,12,16 * * 1-5 /opt/bin/backup-firebird
#
# Rotate HTTP logs
#
0 4 * * 0 /opt/cfw/sbin/processhttpdlogs
59 23 31 12 * /opt/cfw/sbin/processhttpdlogs-ny
#
# SpamAssassin updates
#
30 3 * * * /opt/cfw/bin/sa-update --nogpg && /usr/sbin/svcdm restart spamd
#
# Cleanup
#
45 3 * * * /opt/bin/cleanup
```

- Zadanie `logadm`, służące do rotacji dzienników systemowych, uruchamiane jest codziennie o 3:10.
- Drugie zadanie, `rtc` służy do przestawiania czasu z zimowego na letni i na odwrót. Wykonywane jest codziennie o 2:01.
- Skrypt `backup` uruchamiany jest codziennie o 2:00, skrypt `backup-system` o 5:00 tylko w niedzielę bo trwa długo, a wczesnym rankiem w niedzielę obciążenie jest mniejsze.
- Skrypt `backup-mysql` jest wykonywany co godzinę, 5 minut po pełnej godzinie. To gwarantuje nam, że w przypadku katastrofy, możemy cofnąć się do zawartości bazy MySQL sprzed godziny.
- Baza danych Firebird SQL jest duża i zbyt częste tworzenie kopii zapasowej powoduje spowolnienie pracy serwera. Dlatego kopie te wykonywane są trzy razy w ciągu dnia od poniedziałku do piątku włącznie o 8:15, 12:15 i 16:15, przy pomocy skryptu `backup-firebird`.
- Skrypt `processhttpdlogs`, do rotacji logów serwera HTTP Apache, wykonywany jest o godzinie 4:00 w niedzielę.
- Natomiast `processhttpdlogs-ny` wykonywany jest raz w roku 31 grudnia o 23:59, czyli minutę przed zmianą roku w dacie, aby podzielić statystyki serwera HTTP na poszczególne lata.
- Program `sa-update` do aktualizacji reguł programu antyspamowego SpamAssassin uruchamiany jest codziennie o 3:30.
- Skrypt usuwający zbędne pliki tymczasowe `cleanup` wołany jest codziennie o 3:45.