

Systemy operacyjne

Mariusz Żynel

`mariusz@math.uwb.edu.pl`

`http://math.uwb.edu.pl/~mariusz/`

Uniwersytet w Białymstoku

2023/2024

- Słowo **zadanie** w kontekście systemu operacyjnego ma kilka znaczeń:
 - dowolny proces albo wykonywany program w systemie (task, zadanie)
 - proces skojarzony z konkretnym terminalem (job, robota)
- Zadanie tworzymy uruchamiając dowolny program w terminalu
- Każde zadanie w shellu, podobnie jak proces, ma przypisaną unikalną liczbę całkowitą, służącą do identyfikowania zadań
- Identyfikatorów zadań używamy poprzedzając je znakiem %
- Każde zadanie może być w jednym z trzech stanów:
 - uruchomione na pierwszym planie (foreground)
 - uruchomione w tle (background)
 - zatrzymane
- Zadania działające w tle nie mają dostępu do standardowego wejścia, nie mogą zatem pobierać danych z terminala
- Program uruchamiamy w tle podając znak & na końcu polecenia

`jobs` wyświetla listę zadań wykonywanych w tle lub zatrzymanych

`fg` przenosi wskazane zadanie na pierwszy plan (foreground)

`bg` przenosi wskazane zadanie w tło (background)

`stop` zatrzymuje zadanie działające w tle (tylko `sh`, `csh`, `ksh`)

`notify` powiadamia użytkownika o zmianach stanu zadań (tylko `csh`)

Uruchamianie programów w ustalonym terminie

- Zadania czasochłonne lub wymagające wielu zasobów na systemach, gdzie pracuje wiele osób w tym samym czasie, uruchamiamy w godzinach nocnych, gdy obciążenie systemu jest mniejsze
- Gdy wiemy, że plik, który chcemy pobrać, pojawi się w późniejszym terminie, zadanie do jego pobrania można uruchomić określonego dnia, o określonej porze
- W Unixie system pozwalający planować zadania, ustawiać harmonogram zadań oraz wykonujący zadania w określonym terminie to `cron`

Dopisywanie zadań do kolejki

at wczytuje polecenia ze standardowego wejścia i grupuje je w zadanie do wykonania o późniejszym terminie

batch wczytuje polecenia do wykonania ustawiając je w specjalnej kolejce, wykonanie może być opóźnione przez podanie limitów na ilość zadań możliwych do wykonania równoległe, parametry określa się w pliku `/etc/cron.d/queuedefs`

```
sirius$ at midnight
at> wget -c -nd ftp://solaris-x86.org/jumpstart.iso
at> <EOT>
commands will be executed using /bin/bash
job 1589839200.a at Wt maj 12 00:00:00 2020
```

```
sirius$ at 3:30 friday
at> make build/illumos/kernel
at> <EOT>
commands will be executed using /bin/bash
job 1590111000.a at Pt maj 24 03:30:00 2020
```

Zarządzanie kolejką zadań

`atq` wyświetla listę zadań do wykonania przez aktualnego użytkownika, jeśli program uruchomiony jest przez uprzywilejowanego użytkownika wyświetla pełną listę wszystkich zadań, lub wszystkich zadań podanego użytkownika

`atrm` usuwa zadanie o podanym numerze z kolejki do wykonania, tylko uprzywilejowany użytkownik może usuwać zadania innych użytkowników

```
sirius$ atq
```

Rank	Execution Date	Owner	Job	Queue	Job Name
1st	May 12, 2020 00:00	mariusz	1589839200.a	a	stdin
2nd	May 15, 2020 03:30	mariusz	1590111000.a	a	stdin

```
sirius$ atrm 1590111000.a  
1590111000.a: removed
```

- Jest wiele zadań, które administrator powinien wykonywać na serwerze w regularnych odstępach czasu, na przykład:
 - tworzenie kopii bezpieczeństwa
 - rotacja dzienników
 - usuwanie zbędnych plików tymczasowych
 - zmiana czasu z zimowego na letni lub na odwrot
- W Unixie system pozwalający planować zadania, ustawiać harmonogram zadań oraz wykonujący zadania w określonym terminie to `cron`
- Ze względów bezpieczeństwa lista katalogów `$PATH`, z których uruchamiane mogą być programy jest ograniczona z poziomu `cron`

`crontab` zarządza harmonogramem zadań użytkownika

-e edycja pliku cron danego użytkownika

-l wyświetla zawartość pliku cron

Składnia pliku cron:

- minuta (0-59)
- godzina (0-23)
- dzień miesiąca (1-31)
- miesiąc (1-12)
- dzień tygodnia (0-6, 0 = niedziela)
- nazwa programu ze ścieżką i parametrami do wykonania

Przykład pliku cron

```
# 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
#
# 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/svcadm restart spamd
#
# Cleanup
#
45 3 * * * /opt/bin/cleanup
```