
OPERACJE NA PROCESACH (CD): SYGNAŁY, LIMITY ZASOBÓW, PRIORYTETY

JAKUB SZCZYRK
235477

PONIEDZIAŁEK 13:15-14:00

25 LISTOPADA 2018

Spis treści

1	Przechwytywanie sygnałów	2
2	Zawieszanie i wznowianie procesu	2
3	Ulimit	2
4	Wartości priorytetów procesów i ich liczby nice	3

1 Przechwytywanie sygnałów

Napisany skrypt, który co trzy sekundy wyświetla bieżącą godzinę. Polecenia **SIGINT** i **SIGQUIT** nie zatrzymały procesu, natomiast **SIGILL** oraz **SIGFPE** przerwało wykonywanie procesu. Po zastosowaniu w skrypcie komendy `trap` przechwytywającej 15 sygnałów (poniższy skrypt), żaden z nich nie mógł zabić procesu, z wyjątkiem **SIGKILL**.

```
#!/bin/bash

echo "pid $$"

#trap 'echo " 1SIG_echo"' SIGINT
#trap 'echo " 2SIG_echo"' SIGQUIT
#trap 'echo " 3SIG_echo"' SIGFPE
#trap 'echo " 4SIG_echo"' SIGILL
#trap 'echo " 5SIG_echo"' SIGHUP
#trap 'echo " 6SIG_echo"' SIGTRAP
#trap 'echo " 7SIG_echo"' SIGCHLD
#trap 'echo " 8SIG_echo"' SIGBUS
#trap 'echo " 9SIG_echo"' SIGABRT
#trap 'echo "10SIG_echo"' SIGKILL
#trap 'echo "11SIG_echo"' SIGUSR1
#trap 'echo "12SIG_echo"' SIGUSR2
#trap 'echo "13SIG_echo"' SIGPIPE
#trap 'echo "14SIG_echo"' SIGALRM
#trap 'echo "15SIG_echo"' SIGTERM

while : ; do
    DATE=$(date '+%Y-%m-%d %H:%M:%S')
    echo $DATE
    sleep 3
done
```

Rysunek 1: script.sh

2 Zawieszanie i wznowianie procesu

Zawieszania procesu i wznowiania jest możliwe za pomocą sygnałów:

- `killall -19 script.sh` dla **SIGSTOP** (zawieszenie)
- `killall -18 script.sh` dla **SIGCONT** (wznowienie)

Zatrzymać proces można także za pomocą `Ctrl+Z`, jednakże nie da się go wznowić sygnałem **SIGCONT**, aby to zrobić trzeba użyć polecenie `fg [jobs_id]` albo `bg [jobs_id]`, z tą różnicą, że po `bg` nie da się go zatrzymać `Ctrl+Z`.

SIGCONT, `bg` i `fg` dają taki sam efekt.

3 Ulimit

Funkcja `ulimit` pozwala ustawić maksymalną liczbę procesów użytkownika. Po zastosowaniu polecenia `ulimit` program `zad3.sh` wykonał pętlę tylko raz i wyświetlił nazwę programu. Oznacza to, że działanie funkcji `ulimit` było respektowane.

```
#!/bin/bash

echo $0

sleep 5
while : ; do
    ./zad3.sh
    kill -19 $$
done
```

Rysunek 2: zad3.sh

4 Wartości priorytetów procesów i ich liczby nice

Polecenie nice służy do uruchomienia programu z określonym priorytetem:

```
nice -n 15 [polecenie]
```

Takie polecenie działa tylko dla uruchomienia programu.

Żeby zmienić wartość priorytetu nice dla już działającego programu czy procesu – musimy skorzystać z narzędzia renice:

```
renice 0 [PID]
```

Polecenie nice działa z nazwami programów, a renice z identyfikatorami procesów PID.