Współczesne systemy komputerowe
Zarządzanie procesami

Monitorowanie procesów

- Do wyświetlenia procesów można użyć polecenia **ps**

```
root@debian:~# ps
  PID TTY          TIME CMD
 1223 pts/0    00:00:00 su
 1224 pts/0    00:00:00 bash
 1232 pts/0    00:00:00 ps
```

a	procesy kontrolowane przez terminal
x	dodatkowo, procesy nie kontrolowane przez terminal
w	opis szczegółowy
u	sortowanie według użytkowników
f	format hierarchiczny
l	opis szczegółowy (w jednej linii)

- Procesy niezależne od terminali (*TTY*)

```
root@debian:~# ps x
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S      0:00 [kthreadd]
    3 ?        S      0:00 [ksoftirqd/0]
    4 ?        S      0:00 [kworker/0:0]

...

 1196 ?        Ss     0:00 /usr/sbin/cupsd -l
 1197 ?        Ssl    0:00 /usr/sbin/cups-browsed
 1223 pts/0    S      0:00 su -
 1224 pts/0    S      0:00 -su
 1237 pts/0    R+     0:00 ps x
```

- Procesy użytkownika *foo*

```
root@debian:~# ps -lU foo
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001   611     1  0  80   0 -  2376 ep_pol ?        00:00:00 systemd
5 S  1001   612   611  0  80   0 -  7012 sigtim ?        00:00:00 (sd-pam)
1 S  1001   618     1  0  80   0 -  9821 poll_s ?        00:00:00 gnome-keyring-
4 S  1001   621   604  0  80   0 -  7387 poll_s tty2     00:00:00 gdm-x-session

...

0 S  1001   960   941  0  80   0 - 34499 poll_s ?        00:00:00 evolution-addr
0 S  1001  1006   611  0  80   0 - 34877 poll_s ?        00:00:01 gedit
0 S  1001  1163   611  0  80   0 - 22711 poll_s ?        00:00:01 gnome-terminal
0 S  1001  1169  1163  0  80   0 -  1600 wait   pts/0    00:00:00 bash
4 S     0  1223  1169  0  80   0 -  1560 wait   pts/0    00:00:00 su
```

- Procesy użytkownika *root*

```
root@debian:~# ps -lU root
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0     1     0  0  80   0 -  6749 ep_pol ?        00:00:01 systemd
1 S     0     2     0  0  80   0 -     0 kthrea ?        00:00:00 kthreadd
1 S     0     3     2  0  80   0 -     0 smpboo ?        00:00:00 ksoftirqd/0
1 S     0     4     2  0  80   0 -     0 worker ?        00:00:00 kworker/0:0

...

4 S     0  1196     1  0  80   0 -  3663 ep_pol ?        00:00:00 cupsd
4 S     0  1197     1  0  80   0 -  8856 poll_s ?        00:00:00 cups-browsed
4 S     0  1224  1223  0  80   0 -  1736 wait   pts/0    00:00:00 bash
1 S     0  1243     2  0  80   0 -     0 worker ?        00:00:00 kworker/0:1
0 R     0  1248  1224  0  80   0 -  1918 -      pts/0    00:00:00 ps
```

- Właściwości procesu

`UID`	numer ID użytkownika
`PID`	numer ID procesu
`PPID`	numer ID rodzica procesu
`TTY`	numer terminala (jeśli kontrolowany przez terminal)
`PRI`	priorytet (mniejsza wartość, więcej czasu procesora)
`NI`	priorytet *nice*
`STAT`	status procesu
`TIME CPU`	czas procesora
`COMMAND`	polecenie

- Status procesu

`R`	proces uruchomiony (*runnable*)
`S`	proces czeka na zdarzenie (*sleeping*)
`D`	proces czeka i nie może być zakończony (*uninterruptable sleep*)
`T`	proces jest wstrzymany (*traced, stopped*)
`X`	proces jest zabity (*dead*)
`Z`	proces zakończony, którego zamknięcie nie zostało obsłużone przez rodzica (*zombie*)

- Wyświetl procesy w formie drzewa

```
root@debian:~# pstree
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─{gdbus}
        │                └─{gmain}

...

        │          ├─{gmain}
        │          └─{probing-thread}
        ├─upowerd─┬─{gdbus}
        │         └─{gmain}
        └─wpa_supplicant
```

Procesy w tle

- Zainstaluj i uruchom program **xosview**

```
root@debian:~# apt install xosview
Reading package lists... Done
Building dependency tree
Reading state information... Done

...
```

```
foo@debian:~# xosview
```

- Po uruchomieniu programu terminal nie jest dostępny
- Ułóż okna programu **xosview** i terminalu tak, aby były widoczne, wstrzymaj proces programu **xosview** wciskając **[CTRL Z]**
- Wyświetl procesy w tle

```
foo@debian:~$ jobs
[1]+  Stopped
```

- Wyświetl procesy poleceniem **ps -l**, zwróć uwagę na status procesu (**T** - wstrzymany)

```
foo@debian:~$ ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1001  1169  1163  0  80   0 -  1600 wait   pts/0    00:00:00 bash
0 T  1001  1414  1169  0  80   0 -  1660 signal pts/0    00:00:00 xosview
0 R  1001  1420  1169  0  80   0 -  1850 -      pts/0    00:00:00 ps
```

- Wznów proces poleceniem **bg** *number*

```
foo@debian:~$ bg 1
[1]+ xosview &
```

- Program zaczął działać w tle (**&**), a terminal nie jest zablokowany, sprawdź procesy w tle

```
foo@debian:~$ jobs
[1]+  Running                 xosview &
```

- Zamknij terminal, program **xosview** przestał działać
- Uruchom terminal i jeszcze dwa programy poleceniami

```
foo@debian:~$ nohup xosview &
[1] 1461
nohup: ignoring input and appending output to 'nohup.out'
foo@debian:~$ xeyes &
[2] 1462
```

- Sprawdź procesy w tle

```
foo@debian:~$ jobs
[1]-  Running                 nohup xosview &
[2]+  Running                 xeyes &
```

- Uczyń proces 2 pierwszoplanowym (**fg 2**) wyślij proces do tła klawiszami **[CTRL Z]**, wznów proces w tle (**bg 2**), wyświetl procesy w tle (**jobs**)

```
foo@debian:~$ fg 2
xeyes
^Z
[2]+  Stopped                 xeyes
```

```
foo@debian:~$ bg 2
[2]+ xeyes &
```

```
foo@debian:~$ jobs
[1]-  Running                 nohup xosview &
[2]+  Running                 xeyes &
```

- Zamknij terminal, program **xeyes** został zamknięty, a **xosview** nadal działa (**nohup** - ignorowanie zamknięcia terminala)

Zmiana priorytetu procesu

- Uruchom dwa programy **xosview** w tle

```
foo@debian:~$ xosview & xosview &
[1] 1513
[2] 1514
```

- Wyświetl procesy i ich priorytety

```
foo@debian:~$ ps x --format " pid %p cputime %C time %t nice %n name %c" | grep xosview
 pid  1513 cputime  0.4 time       01:32 nice   0 name xosview
 pid  1514 cputime  0.4 time       01:32 nice   0 name xosview
```

- Sprawdź numery PID procesów

```
foo@debian:~$ pidof xosview
1514 1513
```

- Zmień priorytet jednego procesu (może wynosić od 19 do -20, czym mniejsza wartość, tym większy priorytet)

```
foo@debian:~$ renice 19 1514
1514 (process ID) old priority 0, new priority 19
```

```
foo@debian:~$ ps x --format " pid %p cputime %C time %t nice %n name %c" | grep xosview
 pid  1513 cputime  0.4 time       04:06 nice   0 name xosview
 pid  1514 cputime  0.4 time       04:06 nice  19 name xosview
```

- Priorytety ujemne może tylko ustawiać użytkownik *root*
- Uruchom program z maksymalnym priorytetem

```
root@debian:~# nice -n -20 xosview &
[1] 1566
```

```
root@debian:~# ps ax --format " pid %p cputime %C time %t nice %n name %c" | grep
       xosview
 pid  1513 cputime  0.5 time       07:24 nice   0 name xosview
 pid  1514 cputime  0.4 time       07:24 nice  19 name xosview
 pid  1566 cputime  0.5 time       00:07 nice -20 name xosview
```

Zabijanie procesów

- Za pomocą polecenia `kill` można wysyłać sygnały do procesów, pozwalających na ich zakończenie

`SIGHUP, 1`	odczytanie plików konfiguracyjnych
`SIGINT, 2`	zatrzymanie z klawiatury **`[CTRL C]`**
`SIGKILL, 9`	wymusza natychmiastowe zakończenie
`SIGTERM, 15`	łagodne zakończenie procesu, pozwala na pozamykanie zasobów i zwolnienie pamięci
`SIGCONT, 18`	wznowienie procesu
`STOP, 19`	zatrzymanie procesu

- Obsługa sygnałów **SIGINT** i **SIGTERM** powinna być zaimplementowana w programie, jeśli proces ignoruje te sygnały można go zabić sygnałem **SIGKILL** (robi to system)
- Sygnały **SIGCONT** i **STOP** również obsługuje system
- Uruchom dwa dowolne programy i zakończ je poleceniem **`kill`**

```
root@debian:~# xosview & xosview &
[1] 1807
[2] 1808
```

```
root@debian:~# kill -15 1807
```

```
root@debian:~# kill -9 1808
[1]-  Terminated              xosview
```

```
root@debian:~# ps ax | grep xosview
 1810 pts/0    S+     0:00 grep xosview
[2]+  Killed                  xosview
```

```
root@debian:~# ps ax | grep xosview
 1813 pts/0    S+     0:00 grep xosview
```

- Innym poleceniem pozwalającym na zamykanie procesów jest **`killall`** (usuwa wszystkie instancje programu, można podać nazwę procesu)

```
root@debian:~# xeyes & xeyes &
[1] 1817
[2] 1818
```

```
root@debian:~# ps ax | grep xeyes
 1817 pts/0    S      0:00 xeyes
 1818 pts/0    S      0:00 xeyes
 1820 pts/0    S+     0:00 grep xeyes
```

```
root@debian:~# killall xeyes
[1]-  Terminated              xeyes
[2]+  Terminated              xeyes
```

```
root@debian:~# ps ax | grep xeyes
 1825 pts/0    S+     0:00 grep xeyes
```

- Sprawdź działanie polecenia **`top`**