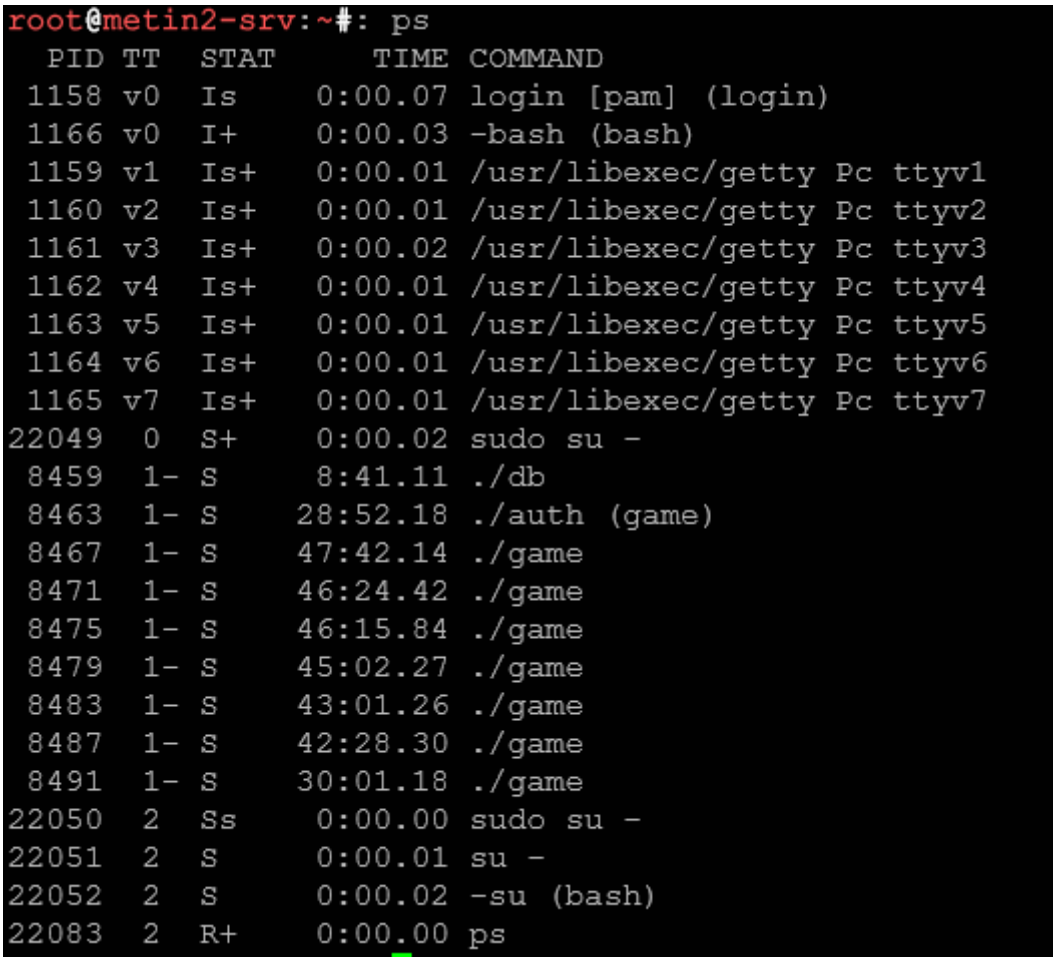
Obsługa komendy "ps" - Proces Status

Komenda `ps` (process status) jest jednym z fundamentalnych narzędzi administracyjnych do monitorowania procesów w systemach Unix/Linux. Pozwala na identyfikację, śledzenie i diagnostykę działających aplikacji. W pracy z serwerami (szczególnie przy Proxmox, FreeBSD czy administracji gra serwerów MT2) niezbędna jest umiejętność interpretacji wyniku `ps`.

Podstawowe użycie `ps`

```
ps
```

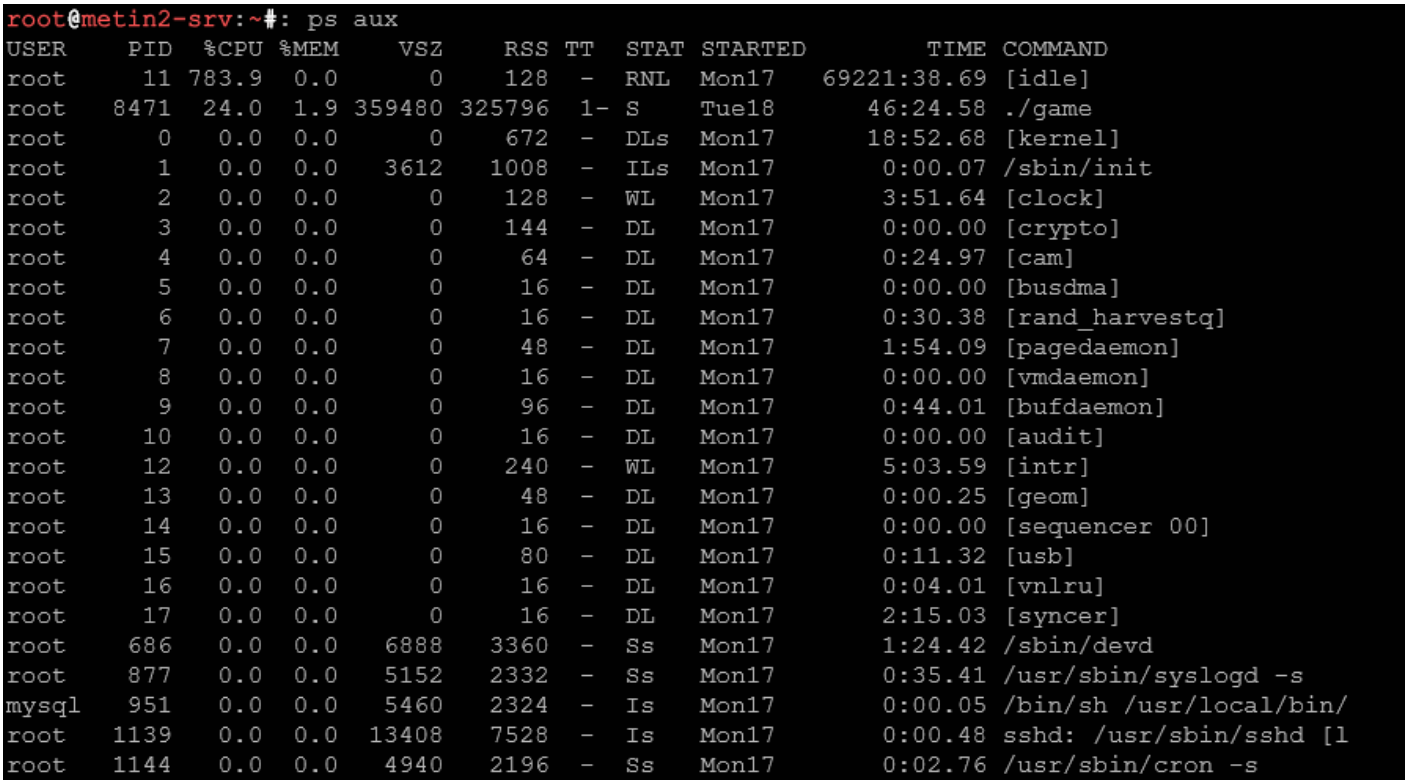
Wyświetla procesy bieżącego użytkownika w aktualnej sesji terminalowej. Bardzo podstawowe, rzadko wystarczające dla administratora.



Przeglądanie wszystkich procesów z pełnymi informacjami

```
ps aux
```

To najczęstsze polecenie dla administratorów. Wyświetla wszystkie procesy w systemie (`a` – wszystkie, `u` – szczegółowy format, `x` – procesy bez terminala). Kolumny oznaczają: USER (właściciel), PID (ID procesu), %CPU (procent CPU), %MEM (procent pamięci), VSZ (rozmiar wirtualny), RSS (resident set size – fizyczna pamięć), TTY (terminal), STAT (status), START (czas startu), TIME (czas CPU), COMMAND (polecenie).



Alternatywny format wyświetlania

```
ps -ef
```

Inny sposób na wyświetlenie wszystkich procesów, bardziej zbliżony do formatu Linuksa (zamiast BSD-owego `aux`). PPID (ID procesu nadrzędnego) jest tu bardziej widoczne.

```
root@metin2-srv:~#: ps -ef
  PID TT  STAT      TIME COMMAND
 1158 v0  Is     0:00.07 TERM=xterm login [pam] (login)
 1166 v0  I+     0:00.03 USER=root LOGNAME=root HOME=/root SHELL=/usr/local/bin/bash BLOCKSIZE=K MAIL=
 1159 v1  Is+    0:00.01 TERM=xterm /usr/libexec/getty Pc ttyv1
 1160 v2  Is+    0:00.01 TERM=xterm /usr/libexec/getty Pc ttyv2
 1161 v3  Is+    0:00.02 TERM=xterm /usr/libexec/getty Pc ttyv3
 1162 v4  Is+    0:00.01 TERM=xterm /usr/libexec/getty Pc ttyv4
 1163 v5  Is+    0:00.01 TERM=xterm /usr/libexec/getty Pc ttyv5
 1164 v6  Is+    0:00.01 TERM=xterm /usr/libexec/getty Pc ttyv6
 1165 v7  Is+    0:00.01 TERM=xterm /usr/libexec/getty Pc ttyv7
22049  0  S+     0:00.03 SHELL=/usr/local/bin/bash PWD=/home/konio LOGNAME=konio HOME=/home/konio LANG
 8459  1- S     8:41.17 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8463  1- S    28:52.37 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8467  1- S    47:42.42 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8471  1- S    46:24.73 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8475  1- S    46:16.12 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8479  1- S    45:02.55 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8483  1- S    43:01.53 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8487  1- S    42:28.56 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
 8491  1- S    30:01.37 LANG=C.UTF-8 MAIL=/var/mail/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/loca
22050  2  Is     0:00.00 SHELL=/usr/local/bin/bash PWD=/home/konio LOGNAME=konio HOME=/home/konio LANG
22051  2  I      0:00.01 LANG=C.UTF-8 TERM=xterm-256color PS1=\\[\\033[00;36m\\]\\u\\[\\033[00;32m\\]@
22052  2  S      0:00.03 TERM=xterm-256color BLOCKSIZE=K MAIL=/var/mail/root MM_CHARSET=UTF-8 LANG=C.U
22085  2  R+     0:00.00 SHELL=/usr/local/bin/bash PWD=/root HOME=/root LANG=C.UTF-8 TERM=xterm-256col
```

Filtrowanie procesów konkretnego użytkownika

```
ps -u 'nazwa_usera'
```

```
root@rivia:~$: ps -u konio
    PID TTY          TIME CMD
 751935 pts/4    00:00:00 bash
 752053 pts/4    00:00:00 cat
root@rivia:~$:
```

Pokazuje tylko procesy należące do użytkownika `admin`. Przydatne gdy szukamy co robi konkretny użytkownik.

```
ps -u root,www-data
```

Procesy należące do wielu użytkowników naraz.

Wyszukiwanie konkretnych procesów

```
ps aux | grep 'nazwa procesu'
```

Wyszukuje procesy związane z danym procesem

```
root@metin2-srv:~#: ps aux | grep game
root    8471  24.0  1.9 359460 325776  1- S    Tue18      46:25.60 ./game
root    8463   0.0  0.1  32804  14744  1- S    Tue18      28:52.88 ./auth (game)
root    8467   0.0  1.7 313256 280992  1- S    Tue18      47:43.27 ./game
root    8475   0.0  2.0 378280 343216  1- S    Tue18      46:16.94 ./game
root    8479   0.0  1.7 315012 282844  1- S    Tue18      45:03.38 ./game
root    8483   0.0  1.9 359492 326000  1- S    Tue18      43:02.30 ./game
root    8487   0.0  2.0 370216 336644  1- S    Tue18      42:29.32 ./game
root    8491   0.0  0.3  88180  56548  1- S    Tue18      30:01.91 ./game
root   22096   0.0  0.0   4600   2008  2  S+   19:23       0:00.00 grep game
root@metin2-srv:~#:
```

```
ps aux | grep -v grep
```

```
konio@metin2-srv:~$: ps | grep -v grep
  PID TT  STAT      TIME COMMAND
22017  0  Ss    0:00.04 -bash (bash)
22108  0  R+    0:00.00 ps
konio@metin2-srv:~$:
```

```
konio@metin2-srv:~$: ps | grep ps
22110  0  R+    0:00.00 ps
22111  0  S+    0:00.00 grep ps
konio@metin2-srv:~$:
```

Eliminuje z wyniku samo polecenie grep (często pojawia się w wynikach).

```
pgrep -lf 'nazwa_procesu'
```

Szuka procesu po pełnej nazwie polecenia (bardziej precyzyjne niż grep).

```
root@rivia:~$: pgrep -lf 'bash'
1293 ksmtuned
754577 bash
754723 bash
root@rivia:~$:
```

Sortowanie procesów

```
ps aux --sort=-%cpu | head -20
```

Wyświetla 20 procesów zużywających najwięcej CPU, posortowanych malejąco.

```
root@rivia:~$: ps a --sort=-%cpu | head -20
    PID TTY      STAT   TIME COMMAND
 754577 pts/0    Ss     0:00 -bash
2100932 pts/3    Ss+    0:05 lxc-console -n 102 -e -1
   1542 tty1     Ss+    0:00 /sbin/agetty -o -- \u --noreset --noclear - linux
   2649 pts/0    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 linux
   2650 pts/1    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - vt220
   2651 pts/2    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - vt220
   2950 pts/0    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 linux
   2952 pts/2    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - linux
 698691 pts/1    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - linux
 757463 pts/1    Ss+    0:00 /bin/bash /usr/lib/inithooks/bin/login_script.sh
 757495 pts/1    S+     0:00 login -p
 757722 pts/0    R+     0:00 ps a --sort=-%cpu
 757723 pts/0    S+     0:00 head -20
2100047 pts/0    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 linux
2100053 pts/2    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - linux
2100830 pts/1    Ss+    0:00 lxc-console -n 100 -e -1
2100877 pts/2    Ss+    0:00 lxc-console -n 101 -e -1
root@rivia:~$:
```

```
ps aux --sort=-%mem | head -20
```

Wyświetla 20 procesów zużywających najwięcej pamięci RAM.

```
root@rivia:~$: ps a --sort=-%mem | head -20
    PID TTY      STAT   TIME COMMAND
 754577 pts/0    Ss     0:00 -bash
2100830 pts/1    Ss+    0:00 lxc-console -n 100 -e -1
 757548 pts/0    R+     0:00 ps a --sort=-%mem
2100932 pts/3    Ss+    0:05 lxc-console -n 102 -e -1
2100877 pts/2    Ss+    0:00 lxc-console -n 101 -e -1
 757463 pts/1    Ss+    0:00 /bin/bash /usr/lib/inithooks/bin/login_script.sh
 757495 pts/1    S+     0:00 login -p
   1542 tty1     Ss+    0:00 /sbin/agetty -o -- \u --noreset --noclear - linux
   2952 pts/2    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - linux
 757549 pts/0    S+     0:00 head -20
   2950 pts/0    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 linux
   2650 pts/1    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - vt220
   2651 pts/2    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - vt220
   2649 pts/0    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 linux
 698691 pts/1    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - linux
2100047 pts/0    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 115200,38400,9600 linux
2100053 pts/2    Ss+    0:00 /sbin/agetty -o -p -- \u --noclear - linux
root@rivia:~$:
```

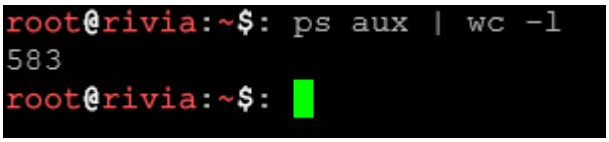
```
ps aux --sort=-vsz
```

Sortuje po wirtualnym rozmiarze pamięci.

```
root@rivia:~$: ps aux --sort=-vsz
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root     2101790  1.9  1.8 22144620 1988276 ?    Sl   Oct30 677:46 /usr/bin/kvm -id 203 -name leo003,d
root       23891  2.9  3.9 9004348 4248596 ?     Sl   Oct26 1185:01 /usr/bin/kvm -id 201 -name leo001,
root        3648  2.8  3.9 8981024 4234896 ?     Sl   Oct26 1158:59 /usr/bin/kvm -id 202 -name leo002,
root     1991453  1.8  2.7 5942868 2938288 ?     Sl   Nov05 494:09 /usr/bin/kvm -id 504 -name kon003,d
root     2102087  8.0  0.0 5082176 98444 ?       Sl   Oct30 2812:36 /usr/bin/kvm -id 507 -name graylog
root     2238514  0.8  1.4 3461988 1548892 ?     Sl   Oct30 291:57 /usr/bin/kvm -id 402 -name mar002-a
root     2215576  0.5  1.7 3068684 1925888 ?     Sl   Oct30 202:37 /usr/bin/kvm -id 401 -name mar001-d
zabbix      1492  0.0  0.0 2437904 23796 ?       Ssl  Oct26   9:28 /usr/sbin/zabbix_agent2 -c /etc/zab
root        2939  0.0  0.0 1390932 37268 ?       Ssl  Oct26  22:36 /usr/bin/python3 /usr/bin/fail2ban-
ceph        1715  0.1  0.3 1194276 403348 ?      Ssl  Oct26  74:17 /usr/bin/ceph-mgr -f --cluster ceph
root        1286  0.0  0.0 1125756 9444 ?        Ssl  Oct26  21:50 /usr/bin/rrdcached -g
systemd+    3033  0.0  0.0 1088632 43076 ?       Ssl  Oct26   6:28 /usr/sbin/mariadbd
root        1514  0.1  0.0 887876 68488 ?        Ssl  Oct26  46:33 /usr/bin/pmxcfs
root        1852  0.0  0.0 728268 16044 ?        SLsl Oct26   3:33 /usr/sbin/glusterfsd -s 192.168.1.1
```

```
ps aux | wc -l
```

Liczy całkowitą liczbę procesów w systemie.



Monitorowanie procesów hiearchy – relacje między procesami

```
ps auxf
```

Wyświetla procesy w formie drzewa, pokazując które procesy są "dziećmi" innych procesów.

```
pstree -p user
```

Pokazuje drzewo procesów danego użytkownika z PID.

Monitorowanie procesów w czasie – status i sygnały

```
ps aux | grep -E 'S|Ss|R|T'
```

Statusy procesów: S (sleeping – uśpiony), Ss (session leader), R (running – działający), T (stopped – zatrzymany), Z (zombie).

Zombie procesy – problem w naszych czatach:

```
ps aux | grep Z
```

Wyszukuje procesy zombie (znane z naszych dzienników z serwera MT2SRV, gdzie czasami pojawiały się procesy nieusunięte prawidłowo).

Wyświetlanie konkretnych kolumn

```
ps -o pid,user,%cpu,%mem,comm
```

Wyświetla tylko wybrane kolumny: PID, użytkownik, CPU, pamięć i nazwa polecenia.

```
ps aux -o pid,ppid,user,comm
```

PID, PPID (proces nadrzędny), użytkownik i nazwa polecenia – przydatne do zrozumienia zależności między procesami.

Monitorowanie procesów w pętli rzeczywistej – ale nie programem top

```
while true; do ps aux | grep game; sleep 2; done
```

Monitoruje konkretny proces co 2 sekundy. Przydatne gdy chcemy obserwować czy proces się restartuje.

Specjalistyczne zapytania

```
ps aux | awk '$3 > 10 { print $0 }'
```

Wyświetla procesy zużywające więcej niż 10% CPU.

```
ps aux | awk '$4 > 20 { print $2, $4, $11 }'
```

Wyświetla procesy zużywające więcej niż 20% pamięci (PID, %MEM, komenda).

Praktyczne scenariusze

```
# Sprawdzenie stanu serwera gry MT2SRV
ps aux | grep -E 'game|db|auth'
```

Szukamy trzech kluczowych procesów serwera.

```
# Znalezienie procesu zużywającego zasoby
ps aux --sort=-%mem | head -1
```

Proces zużywający największą pamięci.

```
# Znalezienie i zabicie zombie procesów
ps aux | grep Z | awk '{print $2}' | xargs kill -9
```

Ostrożnie! Usuwa procesy zombie (choć powinny być usunięte automatycznie).

Revision #1
Created 23 November 2025 18:12:08 by Tomasz Konieczny
Updated 23 November 2025 18:36:37 by Tomasz Konieczny