# Tworzenie i operacje na procesach

Jakub Szczyrk
235477

Poniedzałek 13:15-14:00

11 listopada 2018

# Spis treści

**1 Zapoznanie się z programem ps** **2**

**2 Skrypty osierocone** **3**

**3 Potoki** **3**

**4 Fifo** **3**

# 1 Zapoznanie się z programem ps

Komenda „ps” pozwala wyświetlić działające procesy, z opcjami „-l”(długa lista/szczegółowa), „-f” wyświetla informacje na temat każdego procesu.

Procesy w systemie w chwili przykładowego wywołania ps(3 procesy):

| F | S | UID | PID | PPID | C | PRI | NI | ADDR | SZ | WCHAN | TTY | TIME | CMD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | R | 14965 | 5890 | 10542 | 0 | 80 | 0 | - | 7839 | - | pts/11 | 00:00:00 | ps |
| 0 | S | 14965 | 10542 | 13332 | 0 | 80 | 0 | - | 3452 | - | pts/11 | 00:00:00 | bash |
| 0 | S | 14965 | 13332 | 13302 | 0 | 80 | 0 | - | 3805 | - | pts/11 | 00:00:00 | bash |

Procesy sshd w systemie w chwili wykonywania ćwiczenia, wywołane komendą ps -fp $(pgrep -d, -x sshd)(45 procesów):

```
UID          PID  PPID  C STIME TTY          TIME CMD
root        1826     1  0 paź23 ?        00:00:16 /usr/sbin/sshd -D
root        2245  1826  0 13:31 ?        00:00:00 sshd: jpal [priv]
root        4719  1826  0 13:25 ?        00:00:00 sshd: sstelmac [priv]
jpal        5633  2245  0 13:31 ?        00:00:01 sshd: jpal@pts/4
jszczyrk   13302 58414  0 13:20 ?        00:00:00 sshd: jszczyrk@pts/11
root       17190  1826  0 13:22 ?        00:00:00 sshd: ppawlows [priv]
sstelmac   19631  4719  0 13:25 ?        00:00:00 sshd: sstelmac@pts/10
ppawlows   20799 17190  0 13:23 ?        00:00:00 sshd: ppawlows@pts/15
root       22633  1826  0 13:23 ?        00:00:00 sshd: jszczyrk [priv]
jszczyrk   25049 22633  0 13:23 ?        00:00:00 sshd: jszczyrk@pts/16
root       27431  1826  0 13:25 ?        00:00:00 sshd: afilosek [priv]
root       29236  1826  0 10:31 ?        00:00:00 sshd: prodak [priv]
prodak     29539 29236  0 10:31 ?        00:00:00 sshd: prodak@pts/17
root       30162  1826  0 11:31 ?        00:00:00 sshd: amielcza [priv]
amielcza   30281 30162  0 11:31 ?        00:00:00 sshd: amielcza@pts/36
afilosek   30974 27431  0 13:25 ?        00:00:00 sshd: afilosek@pts/18
root       35676  1826  0 13:28 ?        00:00:00 sshd: mkowals3 [priv]
root       37949  1826  0 13:22 ?        00:00:00 sshd: srajca [priv]
mkowals3   38663 35676  0 13:28 ?        00:00:00 sshd: mkowals3@pts/24
root       43098  1826  0 13:27 ?        00:00:00 sshd: srychel [priv]
srychel    46886 43098  0 13:27 ?        00:00:01 sshd: srychel@pts/19
srajca     48548 37949  0 13:22 ?        00:00:00 sshd: srajca@pts/12
root       49003  1826  0 13:16 ?        00:00:00 sshd: mkowals3 [priv]
root       52327  1826  0 13:34 ?        00:00:00 sshd: kzajac [priv]
root       52764  1826  0 13:28 ?        00:00:00 sshd: jkliszc2 [priv]
root       54334  1826  0 13:35 ?        00:00:00 sshd: srychel [priv]
root       54928  1826  0 13:35 ?        00:00:00 sshd: unknown [priv]
sshd       54929 54928  0 13:35 ?        00:00:00 sshd: unknown [net]
srychel    56222 54334  0 13:35 ?        00:00:00 sshd: srychel@pts/7
jkliszc2   56289 52764  0 13:28 ?        00:00:00 sshd: jkliszc2@pts/25
root       56403  1826  0 13:22 ?        00:00:00 sshd: akrawczu [priv]
kzajac     56511 52327  0 13:34 ?        00:00:00 sshd: kzajac@pts/29
root       58414  1826  0 13:20 ?        00:00:00 sshd: jszczyrk [priv]
root       58582  1826  0 paź25 ?        00:00:00 sshd: witold [priv]
witold     58592 58582  0 paź25 ?        00:00:00 sshd: witold@pts/6
akrawczu   60077 56403  0 13:22 ?        00:00:00 sshd: akrawczu@pts/13
mkowals3   62621 49003  0 13:16 ?        00:00:00 sshd: mkowals3@pts/5
```

Polecenie ps:

- -u pokazuje właściciela procesu i dodatkowe dane danego użytkownika
- -p wyszukuje procesy po pid
- -t wyszukuje procesy podpięte pod dany terminal
- -f długi format, wyświetla id użytkownika,procesu, rodzica procesu,tty na którym uruchomiono komendę
- -l długi format, oprócz informacji przy -f, wyświetla również priorytet, liczbę nice i rozmiar procesu.

Liczba procesów można otrzymać za pomocą komendy wc -l .

# 2 Skrypty osierocone

Pod jednym terminalem wywołano procesy skr.sh, skr2.sh i skr3.sh. W kolejnym terminalu za pomocą ps zweryfikowano zależność potomków i rodziców w hierarchii, gdzie pierwszy proces był rodzicem drugiego, a drugi trzeciego. Zabijając poleceniem kill proces drugi, proces trzeci został osierocony i przejęty przez proces init.

# 3 Potoki

Uruchomiono potok wywołujący skrypt, który wyświetla co sekundę komunikat, pgrep i cat:

```
sh komunikat.sh | grep KOMUNIKAT | cat
```

Między procesami zachodzi pokrewieństwo – wszystkie mają wspólnego rodzica, którym jest terminal.

# 4 Fifo

Stworzono potok o nazwie FIFO za pomocą polecenia mknod potok p.

```
cat > potok
```

utworzyło potok piszący, a cat potok utworzyło potok czytający. Cat piszący czekał na wpisane dane oraz enter i wysyłał je, a cat czytający czekał na dane lub zakończenie się cat piszącego i wtedy również się kończył. Proces który obudził się jako pierwszy przeczytał dany fragment z racji tego, że jest to komunikacja jeden do jednego. Unicestwienie cat piszącego unicestwia wszystkie cat czytające, a unicestwienie cat czytającego zabija jedynie ten proces.