

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут комп'ютерних систем

Кафедра інформаційних систем

Лабораторна робота №6

З дисципліни «Операційні системи»

**Тема: «Команди управління процесами в ОС Unix»**

Виконав:

Студент групи AI-202

Полянський М.О.

Перевірили:

Блажко О. А.

Дрозд М.О.

Одеса 2021

**Мета роботи:** отримання навичок в управлінні процесами в ОС Unix засобами командної оболонки.

## План роботи

### Завдання 1 Моделювання багатозадачності

Нехай оперативна пам'ять на комп'ютері-сервері становить  $N$  Гб. Системні процеси ОС займають до  $M$  Гбайт пам'яті, а кожна програма користувача може використовувати до  $K$  Гбайт пам'яті. Нехай в середньому процеси програм користувачів витрачають  $F\%$  свого часу на очікування завершення вводу/виводу. Визначте середню завантаженість процесора, використовуючи значення з таблиці 3 варіантів.

$$N=13 \text{ Гб}$$

$$M=1.5 \text{ Гб}$$

$$K=0.7 \text{ Гб}$$

$$F=30\%$$

$$1-p^n = 1 - 0.3^{16} \approx 1$$

### Завдання 2 Перегляд таблиці процесів

1. Отримайте ієрархію всіх процесів із зазначенням імен користувачів, їх запустили.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ pstree -p
systemd(1)─NetworkManager(500)─┬─{NetworkManager}(544)
                                └─{NetworkManager}(550)
                                └─agetty(545)
                                └─auditd(413)─┬─{auditd}(414)
                                                └─{auditd}(414)
                                └─belobrov.sh(27717)─┬─belobrov.sh(6302)
                                                        └─belobrov.sh(6302)
                                └─belobrov2.sh(13532)─┬─belobrov2.sh(6292)
                                                        └─belobrov2.sh(6292)
                                └─belobrov3.sh(14315)─┬─belobrov3.sh(6290)
                                                        └─belobrov3.sh(6290)
                                └─chronyd(520)
                                └─crond(535)
                                └─dbus-daemon(473)
                                └─exim(802)
                                └─gssproxy(482)─┬─{gssproxy}(492)
                                                ├──{gssproxy}(493)
                                                ├──{gssproxy}(494)
                                                ├──{gssproxy}(495)
                                                └─{gssproxy}(496)
                                └─httpd(932)─┬─httpd(7203)
                                                ├──httpd(8933)
                                                ├──httpd(9780)
                                                ├──httpd(10166)
                                                ├──httpd(21432)
                                                └─httpd(25999)
```

2. Отримайте ієрархію процесів, запущених від імені вашого профілю і з зазначенням PID цих процесів.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ pstree -p polyanskiy_mikola
sshd(8801)──bash(8824)──pstree(32767)
```

3. Отримайте список процесів, запущених в поточному терміналі, зі спрощеним набором колонок виведення таблиці процесів.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -A
  PID TTY          TIME CMD
    1 ?            00:41:33 systemd
    2 ?            00:00:01 kthreadd
    4 ?            00:00:00 kworker/0:0H
    6 ?            00:00:55 ksoftirqd/0
    7 ?            00:00:09 migration/0
    8 ?            00:00:00 rcu_bh
    9 ?            02:23:54 rcu_sched
   10 ?            00:00:00 lru-add-drain
   11 ?            00:00:17 watchdog/0
   12 ?            00:00:20 watchdog/1
   13 ?            00:00:05 migration/1
   14 ?            00:12:13 ksoftirqd/1
   16 ?            00:00:00 kworker/1:0H
   18 ?            00:00:00 kdevtmpfs
   19 ?            00:00:00 netns
   20 ?            00:00:01 khungtaskd
   21 ?            00:00:00 writeback
   22 ?            00:00:00 kintegrityd
   23 ?            00:00:00 bioset
   24 ?            00:00:00 bioset
   25 ?            00:00:00 bioset
```

4. Отримайте список процесів, запущених від імені вашого користувача, з розширеним набором колонок таблиці процесів.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola
  PID TTY          TIME CMD
 4568 pts/36      00:00:00 ps
 8801 ?            00:00:00 sshd
 8824 pts/30      00:00:00 bash
13859 ?            00:00:00 sshd
13884 pts/36      00:00:00 bash
16741 ?            00:00:00 sshd
16763 pts/4        00:00:00 bash
```

5. Отримайте список процесів, запущених від імені вашого користувача із зазначенням наступного набору колонок: PID, TTY, PPID, STAT, NI, CMD

```
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid, tty, ppid, stat, ni, cmd
  PID TT          PPID STAT   NI CMD
 8801 ?            5599 S        0 sshd: polyanskiy_mikola@pts/30
 8824 pts/30        8801 Ss+      0 -bash
13859 ?            11253 S        0 sshd: polyanskiy_mikola@pts/36
13884 pts/36      13859 Ss       0 -bash
16741 ?            13579 S        0 sshd: polyanskiy_mikola@pts/4
16763 pts/4        16741 Ss+      0 -bash
26217 pts/36      13884 R+       0 ps -u polyanskiy_mikola -o pid, tty, ppid, stat, ni, cmd
```

6. Отримайте список всіх сплячих процесів зі спрощеним набором колонок виведення таблиці процесів.

```
[polyanskij_mikola@vpsj3IeQ ~]$ ps -e S
  PID TTY          STAT       TIME COMMAND
    1 ?        Ss      5804:07 /usr/lib/systemd/systemd --switched-root --system --
    2 ?        S         0:01 [kthreadd]
    4 ?        S<        0:00 [kworker/0:0H]
    6 ?        S         0:55 [ksoftirqd/0]
    7 ?        S         0:09 [migration/0]
    8 ?        S         0:00 [rcu_bh]
    9 ?        R      143:54 [rcu_sched]
   10 ?        S<        0:00 [lru-add-drain]
   11 ?        S         0:17 [watchdog/0]
   12 ?        S         0:20 [watchdog/1]
   13 ?        S         0:05 [migration/1]
   14 ?        S      12:13 [ksoftirqd/1]
   16 ?        S<        0:00 [kworker/1:0H]
   18 ?        S         0:00 [kdevtmpfs]
   19 ?        S<        0:00 [netns]
   20 ?        S         0:01 [khungtaskd]
   21 ?        S<        0:00 [writeback]
   22 ?        S<        0:00 [kintegrityd]
   23 ?        S<        0:00 [bioset]
   24 ?        S<        0:00 [bioset]
   25 ?        S<        0:00 [bioset]
```

7. Отримайте список процесів, відсортованих по PID, і визначте:

- загальна кількість запущених процесів;
- кількість процесів, які виконуються;
- кількість сплячих процесів.

```
top - 00:12:19 up 56 days,  7:15, 18 users,  load average: 32.83, 32.00, 28.43
Tasks:  total,      running,      sleeping,      stopped,      zombie
%Cpu(s):    us,      sy,      ni,      id,      wa,      hi,      si,      st
KiB Mem :      total,      free,      used,      buff/cache
KiB Swap:      total,      free,      used.      avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
32336 root        20   0 165892  6632  5116 S   0.0   0.4    0:00.29 sshd
32297 root        20   0 165888  6648  5140 S   0.0   0.4    0:00.34 sshd
31723 kolesni+    20   0 113416  1504  1184 S   0.0   0.1   61:51.43 sh
31225 markovs+   20   0 130680  1644  1268 S   0.0   0.1    0:01.83 ping
30973 markovs+  20   0 130680  1640  1264 S   0.0   0.1    0:01.77 ping
30769 baranyu+   20   0 115688  2156  1676 S   0.3   0.1    0:00.14 bash
30737 markovs+  20   0 130680  1644  1268 S   0.0   0.1    0:01.86 ping
30729 baranyu+  20   0 165888  2604  1092 S   0.0   0.1    0:00.38 sshd
30463 root        20   0 165892  6632  5120 S   0.0   0.4    0:00.32 sshd
30182 kovach_+   20   0 113284  1184  1008 T   0.0   0.1    0:10.52 bash
30137 baranyu+  20   0 113284  1188  1012 T   0.0   0.1    0:09.27 sh
29760 apache     20   0 113480  3120  1592 S   0.0   0.2    0:00.13 httpd
29494 root        20   0 165888  6648  5136 S   0.0   0.4    0:00.30 sshd
28772 apache     20   0 113480  3096  1576 S   0.0   0.2    0:00.05 httpd

28743 kovach_+   20   0 130680  1640  1264 S   0.0   0.1    0:00.48 ping
28415 evchev_+  20   0 130680  1644  1264 S   0.0   0.1    0:15.74 ping
```

8. Отримайте список процесів, відсортованих за % використання процесора.

```
polyanskij_mikola@vpsj3leQ:~  
top - 00:22:24 up 56 days, 7:25, 18 users, load average: 33.14, 33.21, 30.82  
Tasks:      total,      running,      sleeping,      stopped,      zombie  
%Cpu(s):    us,        sy,        ni,        id,        wa,        hi,        si,        st  
KiB Mem :    total,      free,        used,      buff/cache  
KiB Swap:    total,      free,        used,      avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7548	oracle	20	0	113288	1188	1004	R	14.3	0.1	4801:26	test1.sh
7574	oracle	20	0	113288	1188	1004	R	13.3	0.1	4784:59	test2.sh
9006	kovach_+	20	0	113284	1184	1012	R	13.0	0.1	3:22.22	bash
12903	nezhivi+	20	0	113284	1184	1008	R	13.0	0.1	54:25.37	sh
11879	stepanë+	20	0	113284	1188	1012	R	12.6	0.1	2:30.03	sh
18360	nezhivi+	20	0	113284	1184	1008	R	12.3	0.1	59:40.13	sh
6776	kovach_+	20	0	113284	1184	1008	R	12.0	0.1	3:41.33	bash
10866	kovach_+	20	0	113284	1184	1012	R	12.0	0.1	1:31.14	bash
12758	kovach_+	20	0	113284	1188	1012	R	12.0	0.1	1:29.29	bash
16180	bodnar_+	20	0	113284	1184	1008	R	12.0	0.1	71:26.88	sh
12052	kovach_+	20	0	113284	1188	1012	R	11.6	0.1	4:16.65	bash
14981	kovach_+	20	0	113284	1184	1012	R	11.6	0.1	2:56.77	bash
13958	kovach_+	20	0	113284	1184	1008	R	11.0	0.1	2:41.42	bash
14014	bodnar_+	21	1	113284	1188	1012	R	9.6	0.1	59:55.79	sh
7647	bodnar_+	21	1	113284	1188	1012	R	9.3	0.1	60:36.32	sh
11096	nezhivi+	23	5	113284	1184	1012	R	4.0	0.1	22:29.49	sh
6004	stepanë+	30	10	113284	1188	1012	R	1.7	0.1	1:23.42	sh

### Завдання 3 Керування станами процесів

1. У поточному терміналі виконайте команду ping localhost, але не завершуйте її роботу.

```
64 bytes from localhost (127.0.0.1): icmp_seq=90 ttl=64 time=0.040 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=91 ttl=64 time=0.036 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=92 ttl=64 time=0.036 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=93 ttl=64 time=0.037 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=94 ttl=64 time=0.040 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=95 ttl=64 time=0.035 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=96 ttl=64 time=0.040 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=97 ttl=64 time=0.036 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=98 ttl=64 time=0.020 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=99 ttl=64 time=0.031 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

3. У другому терміналі для команди ping отримаєте таблицю її процесу (колонки PID, STAT, CMD).

```
login as: polyanskiy_mikola
polyanskiy_mikola@91.219.60.189's password:
Last login: Wed Apr 14 01:02:29 2021 from 91.225.162.254
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid,stat,cmd
  PID STAT  CMD
  8135 S    sshd: polyanskiy_mikola@pts/22
  8153 Ss+   -bash
  9182 S    sshd: polyanskiy_mikola@pts/13
  9201 Ss    -bash
10861 S+    ping localhost
13487 S    sshd: polyanskiy_mikola@pts/21
13499 Ss    -bash
17990 S    sshd: polyanskiy_mikola@pts/6
18015 Ss+   -bash
18304 R+    ps -u polyanskiy_mikola -o pid,stat,cmd
23108 T    top
[polyanskiy_mikola@vpsj3IeQ ~]$
```

4. У другому терміналі призупиніть виконання процесу команди ping

```
[polyanskiy_mikola@vpsj3IeQ ~]$ kill -19 10861
[polyanskiy_mikola@vpsj3IeQ ~]$
```

5. У першому терміналі отримайте список фонових процесів

```
[polyanskiy_mikola@vpsj3IeQ ~]$ jobs -l
[1]+ 10861 Stopped (signal)          ping localhost
```

6. У другому терміналі відновіть виконання припиненого процесу

```
[polyanskiy_mikola@vpsj3IeQ ~]$ kill -18 10861
```

7. У другому терміналі зупиніть виконання процесу команди ping

```
[polyanskiy_mikola@vpsj3IeQ ~]$ kill 10861
```

8. У першому терміналі запусіть команду ping в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запущений.

```
64 bytes from localhost (127.0.0.1): icmp_seq=371 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=372 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=373 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=374 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=375 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=376 ttl=64 time=0.039 ms
64 bytes from localhost (127.0.0.1): icmp_seq=377 ttl=64 time=0.021 ms

[1]+  Terminated                  ping localhost
[polyanskiy_mikola@vpsj3IeQ ~]$ ping localhost &
[1] 26966
[polyanskiy_mikola@vpsj3IeQ ~]$ PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp seq=1 ttl=64 time=0.028 ms
```

9. Закрийте перший термінал.

10. У другому терміналі для команди `ping` отримаєте таблицю її процесу (колонки PID, STAT, CMD). Зробіть висновок про стан процесу.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid,stat,cmd
  PID STAT  CMD
  2313 S      sshd: polyanskiy_mikola@pts/4
  2356 Ss+    -bash
  2956 S      ping localhost
  4287 R+    ps -u polyanskiy_mikola -o pid,stat,cmd
  8135 S      sshd: polyanskiy_mikola@pts/22
  8153 Ss+    -bash
 13487 S      sshd: polyanskiy_mikola@pts/21
 13499 Ss     -bash
 17990 S      sshd: polyanskiy_mikola@pts/6
 18015 Ss+    -bash
 23108 T      top
```

11. Завершіть роботу процесу.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ kill 2956
```

#### Завдання 4 Управління пріоритетами процесів

1. Створіть `bash`-програму, що виконує операцію циклічного складання за формулою:  $x = x + n$ , де початкове значення  $x$  = кількість букв вашого прізвища,  $n$  - кількість букв у вашому імені. Ім'я програми збігається з транслітерацією вашого прізвища з розширенням `.sh`, наприклад, `ivanov.sh`

```
GNU nano 2.3.1      File: polanskiy.sh      Modified
#!/bin/bash

x=10
n=6
while [ true ]
do
    x=$((x+n))
done
```

2. Запустіть `bash`-програму у фоновому режимі.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ sh polanskiy.sh &
[1] 31271
```

3. Перегляньте таблицю процесів для запущеного процесу, пов'язаного з `bash`- програмою, з урахуванням набором колонок: PID, PPID, STAT, NI,% CPU, CMD.



```
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  2561  13499  S+     0   0.0  nano polyanskiy.sh
  6053  25724  R+     0   0.0  ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
  8135  6987   S      0   0.0  sshd: polyanskiy_mikola@pts/22
  8153  8135   Ss+    0   0.0  -bash
 13487 11611   S      0   0.0  sshd: polyanskiy_mikola@pts/21
 13499 13487   Ss     0   0.0  -bash
 17990 14049   S      0   0.0  sshd: polyanskiy_mikola@pts/6
 18015 17990   Ss+    0   0.0  -bash
 23108 18015   T      0   0.0  top
 25706 24559   S      0   0.0  sshd: polyanskiy_mikola@pts/24
 25724 25706   Ss     0   0.0  -bash
 31271 25724   R      0  10.6  sh polanskiy.sh
```

4. Виконайте команду призупинення запущеного процесу.

```
31271 25724 R      0  10.6  sh polanskiy.sh
[polyanskiy_mikola@vpsj3IeQ ~]$ kill -19 31271
```

5. Ще раз перегляньте таблицю процесів для призупиненого процесу з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан.

```
31271 25724 T      0   9.2  sh polanskiy.sh
[1]+  Stopped                  sh polanskiy.sh
```

6. Виконайте команду продовження виконання припиненого процесу.

7. Ще раз перегляньте таблицю процесів для процесу, який продовжив виконуватися, з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ kill -18 31271
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  2561  13499  S+     0   0.0  nano polyanskiy.sh
  6427  25724  R+     0   0.0  ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
  8135  6987   S      0   0.0  sshd: polyanskiy_mikola@pts/22
  8153  8135   Ss+    0   0.0  -bash
 13487 11611   S      0   0.0  sshd: polyanskiy_mikola@pts/21
 13499 13487   Ss     0   0.0  -bash
 17990 14049   S      0   0.0  sshd: polyanskiy_mikola@pts/6
 18015 17990   Ss+    0   0.0  -bash
 23108 18015   T      0   0.0  top
 25706 24559   S      0   0.0  sshd: polyanskiy_mikola@pts/24
 25724 25706   Ss     0   0.0  -bash
 31271 25724   R      0   5.1  sh polanskiy.sh
```

8. Створіть два файли як символічні посилання на створену bash-програму з іменами як ім'я поточного файлу з додаванням цифр 2 і 3, відповідно, наприклад: ivanov2.sh, ivanov3.sh

9. Запустіть два файли у фоновому режимі.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ sh polanskiy2.sh &
[2] 17718
[polyanskiy_mikola@vpsj3IeQ ~]$ sh polanskiy3.sh &
[3] 18316
```



10. Перегляньте таблицю процесів для трьох запущених процесів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки за поточними значеннями NI та %CPU.

```
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
PID  PPID  STAT  NI  %CPU  CMD
2561  13499 S+    0   0.0  nano polyanskiy.sh
8135   6987 S     0   0.0  sshd: polyanskiy_mikola@pts/22
8153   8135 Ss+   0   0.0  -bash
13487 11611 S     0   0.0  sshd: polyanskiy_mikola@pts/21
13499 13487 Ss    0   0.0  -bash
17718 25724 R     0   8.6  sh polanskiy2.sh
17990 14049 S     0   0.0  sshd: polyanskiy_mikola@pts/6
18015 17990 Ss+   0   0.0  -bash
18316 25724 R     0   8.5  sh polanskiy3.sh
20918 25724 R+    0   0.0  ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
23108 18015 T     0   0.0  top
25706 24559 S     0   0.0  sshd: polyanskiy_mikola@pts/24
25724 25706 Ss    0   0.0  -bash
31271 25724 R     0   6.3  sh polanskiy.sh
```

11. Зменшити пріоритет виконання одного з трьох процесів.

12. Перегляньте таблицю процесів для трьох запущених файлів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки щодо змін значень% CPU для кожного процесу: як вони змінилися?

```
[polyanskiy_mikola@vpsj3IeQ ~]$ renice -n 10 -p 18316
18316 (process ID) old priority 0, new priority 10
[polyanskiy_mikola@vpsj3IeQ ~]$ ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
PID  PPID  STAT  NI  %CPU  CMD
2561  13499 S+    0   0.0  nano polyanskiy.sh
4323  25724 R+    0   0.0  ps -u polyanskiy_mikola -o pid,ppid,stat,ni,%cpu,cmd
8135   6987 S     0   0.0  sshd: polyanskiy_mikola@pts/22
8153   8135 Ss+   0   0.0  -bash
13487 11611 S     0   0.0  sshd: polyanskiy_mikola@pts/21
13499 13487 Ss    0   0.0  -bash
17718 25724 R     0   8.2  sh polanskiy2.sh
17990 14049 S     0   0.0  sshd: polyanskiy_mikola@pts/6
18015 17990 Ss+   0   0.0  -bash
18316 25724 RN   10   8.2  sh polanskiy3.sh
23108 18015 T     0   0.0  top
25706 24559 S     0   0.0  sshd: polyanskiy_mikola@pts/24
25724 25706 Ss    0   0.0  -bash
31271 25724 R     0   7.3  sh polanskiy.sh
```

**Висновок:** в ході лабораторної роботи отримали навички в управлінні процесами в ОС Unix засобами командної оболонки.