

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут комп'ютерних систем
Кафедра інформаційних систем

Варіант №3

Лабораторна робота №6

По дисципліні “Операційні системи”

Тема: «Команди управління процесами в ОС Unix»

Виконав:

Студент групи AI-202

Боднар І.В.

Перевірили:

Блажко О.А

Одеса 2021

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

Хід роботи:

1. Нехай оперативна пам'ять на комп'ютері-сервері становить 10 Гб. Системні процеси ОС займають до 1.5 Гбайт пам'яті, а кожна програма користувача може використовувати до 0.6 Гбайт пам'яті. Нехай в середньому процеси програм користувачів витрачають 35% свого часу на очікування завершення вводу/виводу. Визначте середню завантаженість процесора, використовуючи значення з таблиці 3 варіантів. Тоді середня навантаженість: $1 - p^n = 1 - 0.35^{12} = 0.9999662$
2. Перегляд таблиці процесів
 - 2.1. Отримуємо ієрархію всіх процесів із зазначенням імен користувачів, їх запустили.

- 2.2. Отримуємо ієрархію процесів, запущених від імені вашого профілю із зазначенням PID цих процесів.

```
[bodnar_illya@vpsj3Ieq ~]$ pstree -p bodnar_illya
sshd(6898)---bash(7029)---pstree(13841)

sshd(18169)---bash(18287)
[bodnar_illya@vpsj3Ieq ~]$
```

- 2.3. Отримуємо список процесів, запущених в поточному терміналі, зі спрощеним набором колонок виведення таблиці процесів.

```
[bodnar_illya@vpsj3Ieq ~]$ ps
  PID TTY          TIME CMD
 1324 pts/20    00:00:00 bash
 2589 pts/20    00:00:00 ps
[bodnar_illya@vpsj3Ieq ~]$
```

- 2.4. Отримуємо список процесів, запущених від імені вашого користувача, з розширеним набором колонок таблиці процесів.

```
[bodnar_illya@vpsj3Ieq ~]$ ps -f
UID          PID  PPID  C  SZ  RSS  PSR  STIME  TTY          TIME CMD
bodnar_+    5277    5225  0 28887 2124   1 22:55 pts/4      00:00:00 -bash
bodnar_+   11566    5277  0 38869 1876   1 23:02 pts/4      00:00:00 ps -f
[bodnar_illya@vpsj3Ieq ~]$
```

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

```
[bodnar_illya@vpsj3Ieq ~]$ ps -o pid, tty, ppid, stat, ni, cmd
  PID TT  PPID STAT  NI CMD
 5277 pts/4    5225 Ss      0 -bash
 7500 pts/4    5277 R+      0 ps -o pid, tty, ppid, stat, ni, cmd
[bodnar_illya@vpsj3Ieq ~]$
```

- 2.6. Отримуємо список всіх сплячих процесів зі спрощеним набором колонок виведення таблиці процесів.

```
[bodnar_illya@vpsj3Ieq ~]$ top | grep S
kib Swap: 4194300 total, 3715324 free, 478976 used. 1220928 avail Mem
  PID U  S ER PR NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
27717 belobro+ 20  0 113416 1496 1176 S   4.5  0.1   75:24.07 belobrov.sh
   1 root      20  0 191168 3048 1660 S   0.0  0.2   41:30.50 systemd
   2 root      20  0      0      0    0 S   0.0  0.0    0:01.36 kthreadd
```

- 2.7. Отримуємо список процесів, відсортованих по PID, і визначте:

- 2.7.1. загальна кількість запущених процесів;
- 2.7.2. кількість процесів, які виконуються;
- 2.7.3. кількість сплячих процесів.

```

top - 13:51:54 up 55 days, 20:55, 5 users, load average: 15.33, 15.47, 14.71
Tasks: 165 total, 14 running, 149 sleeping, 2 stopped, 0 zombie
%cpu(s): 56.4 us, 34.6 sy, 9.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1881856 total, 71540 free, 325180 used, 1485136 buff/cache
KiB Swap: 4194300 total, 3715324 free, 478976 used. 1428076 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
31758	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.38	sshd
31723	kolesni+	20	0	113416	1504	1184	S	0.0	0.1	54:44.33	sh
30762	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.30	sshd
29760	apache	20	0	113480	3120	1592	S	0.0	0.2	0:00.11	httpd
29703	karauln+	20	0	130680	1644	1268	T	0.0	0.1	0:00.01	ping
28772	apache	20	0	113480	3096	1576	S	0.0	0.2	0:00.04	httpd
28598	markovs+	20	0	115680	2184	1660	S	0.0	0.1	0:00.04	bash
28578	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.30	sshd
28415	evchev_+	20	0	130680	1644	1264	S	0.0	0.1	0:11.90	ping
28344	markovs+	20	0	163756	2544	1052	S	0.0	0.1	0:00.05	sshd
27717	belobro+	20	0	113416	1496	1176	S	0.0	0.1	69:09.21	belobrov.sh
27587	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:1
27573	soft	30	10	279912	5224	972	S	0.0	0.3	0:00.00	php-fpm
27572	soft	30	10	279912	5224	972	S	0.0	0.3	0:00.00	php-fpm
27571	soft	30	10	279912	5224	972	S	0.0	0.3	0:00.00	php-fpm
27570	emps	30	10	279912	5224	972	S	0.0	0.3	0:00.00	php-fpm
27569	emps	30	10	279912	5224	972	S	0.0	0.3	0:00.00	php-fpm
27568	emps	30	10	279912	5224	972	S	0.0	0.3	0:00.00	php-fpm
27567	root	30	10	279912	5064	816	S	0.0	0.3	0:00.00	php-fpm
27566	root	30	10	279912	5064	816	S	0.0	0.3	0:00.00	php-fpm
27564	root	30	10	279936	5508	1256	S	0.0	0.3	0:09.11	php-fpm
27549	emps	30	10	21488	2128	784	S	0.0	0.1	0:00.00	nginx
27547	root	30	10	20972	1196	276	S	0.0	0.1	0:00.00	nginx
27540	apache	20	0	113480	3124	1604	S	0.0	0.2	0:00.11	httpd
27377	apache	20	0	113480	3100	1580	S	0.0	0.2	0:00.04	httpd
25999	apache	20	0	113480	3096	1576	S	0.0	0.2	0:00.04	httpd
24693	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0
24552	kolesni+	20	0	113416	1504	1184	S	0.0	0.1	55:47.00	sh
24289	demtsun+	20	0	115544	2108	1648	S	0.0	0.1	0:00.00	bash
24116	demtsun+	20	0	165892	2488	972	S	0.0	0.1	0:00.00	sshd
23286	root	30	10	279912	5064	812	S	0.0	0.3	0:00.00	php-fpm
21923	nosov_a+	20	0	696320	864	836	S	0.0	0.0	0:09.38	soffice.bin
21779	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0
21432	apache	20	0	113480	3096	1576	S	0.0	0.2	0:00.04	httpd
21331	markovs+	20	0	110412	992	820	T	0.0	0.1	0:00.00	less
21084	root	20	0	165892	6628	5116	S	0.0	0.4	0:00.34	sshd
20685	baranyu+	20	0	115072	512	508	S	0.0	0.0	0:00.00	nano
20316	nesterer+	25	5	113416	320	0	R	0.0	0.0	0:00.00	nesterenko2.sh
20315	karauln+	20	0	113416	316	0	R	0.0	0.0	0:00.00	karaulniy.sh
20314	kolesni+	20	0	113416	320	0	R	0.0	0.0	0:00.00	sh
20311	belobro+	30	10	113416	316	0	R	0.0	0.0	0:00.00	belobrov3.sh
20302	karauln+	20	0	113416	316	0	R	0.0	0.0	0:00.00	karaulniy3.sh
20300	nesterer+	20	0	113416	320	0	R	0.0	0.0	0:00.00	nesterenko3.sh
20284	nesterer+	20	0	113416	320	0	R	0.0	0.0	0:00.00	nesterenko.sh
20282	belobro+	20	0	113416	320	0	R	0.0	0.0	0:00.00	belobrov2.sh
20001	root	20	0	112924	4300	3276	S	0.0	0.2	0:00.00	sshd
19567	sshd	20	0	117264	2616	1540	S	0.0	0.1	0:00.00	sshd
19336	bodnar_+	20	0	162288	2284	1584	R	0.0	0.1	0:00.13	top
19273	root	20	0	112924	4472	3424	S	0.0	0.2	0:00.00	sshd
18055	bojchuk+	20	0	867800	4780	2008	S	0.0	0.3	0:42.70	soffice.bin
17310	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:1
16666	root	20	0	0	0	0	R	0.0	0.0	0:00.09	kworker/1:3
16171	sshd	20	0	117264	2620	1548	S	0.0	0.1	0:00.01	sshd
15815	root	20	0	158808	5308	4004	S	0.0	0.3	0:00.01	sshd
14520	nesterer+	20	0	113416	1496	1176	S	0.0	0.1	66:11.98	nesterenko.sh
14315	belobro+	30	10	113416	1492	1176	S	2.6	0.1	52:23.15	belobrov3.sh
14276	karauln+	20	0	113416	1492	1176	S	2.6	0.1	0:18.93	karaulniy3.sh
14275	karauln+	20	0	113416	1496	1176	S	2.6	0.1	0:19.19	karaulniy2.sh
14274	karauln+	20	0	113416	1492	1176	S	2.6	0.1	0:18.78	karaulniy.sh
14264	matnenk+	20	0	115680	2172	1660	S	0.0	0.1	0:00.01	bash
14008	matnenk+	20	0	165892	2488	972	S	0.0	0.1	0:00.00	sshd
13532	belobro+	20	0	113416	1496	1176	S	0.0	0.1	68:49.54	belobrov2.sh
12757	bojchuk+	20	0	130680	1644	1268	S	0.0	0.1	0:17.87	ping
12463	bodnar_+	20	0	115548	2116	1656	S	0.0	0.1	0:00.00	bash
12368	bodnar_+	20	0	165892	2476	964	S	0.0	0.1	0:00.02	sshd
12335	bogachi+	20	0	115600	656	652	S	0.0	0.0	0:00.09	nano
11700	root	20	0	163756	6580	5088	S	0.0	0.3	0:00.31	sshd
11579	nesterer+	20	0	113416	1496	1176	S	0.0	0.1	66:20.25	nesterenko3.sh
11578	nesterer+	25	5	113416	1496	1176	S	2.6	0.1	61:08.61	nesterenko2.sh
10894	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/u4:0
10166	apache	20	0	113480	3100	1580	S	0.0	0.2	0:00.04	httpd

2.8. Отримуємо список процесів, відсортованих за % використання процесора.

```
top - 19:59:11 up 56 days, 3:02, 14 users, load average: 24.69, 24.27, 24.12
Tasks: 213 total, 27 running, 185 sleeping, 1 stopped, 0 zombie
%Cpu(s): 60.9 us, 13.4 sy, 25.4 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
KiB Mem : 1881856 total, 143648 free, 534204 used, 1204004 buff/cache
KiB Swap: 4194300 total, 3715324 free, 478976 used. 1218436 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7548	oracle	20	0	113288	1188	1004	R	26.2	0.1	4756:20	test1.sh
16180	bodnar_+	20	0	113284	1184	1008	R	26.2	0.1	22:59.02	sh
14014	bodnar_+	21	1	113284	1188	1012	R	21.5	0.1	17:35.48	sh
4692	lukasha+	20	0	113284	1188	1008	R	16.1	0.1	13:58.40	bash
7574	oracle	20	0	113288	1188	1004	R	16.1	0.1	4734:10	test2.sh
12903	nezhivi+	20	0	113284	1184	1008	R	15.4	0.1	8:04.18	sh
18360	nezhivi+	20	0	113284	1184	1008	R	15.4	0.1	11:18.11	sh
3656	lukasha+	20	0	113284	1188	1008	R	14.8	0.1	14:00.34	bash
7647	bodnar_+	21	1	113284	1188	1012	R	13.4	0.1	17:27.12	sh
11096	nezhivi+	25	5	113284	1184	1012	R	8.1	0.1	3:25.47	sh
14973	kolduno+	25	5	113284	1204	1016	R	4.7	0.1	11:18.82	sh
4410	lukasha+	30	10	113284	1188	1008	R	2.7	0.1	2:40.35	bash
17277	kolduno+	30	10	113284	1200	1016	R	2.7	0.1	4:20.78	sh
22514	root	20	0	161532	5840	4476	S	1.3	0.3	0:00.02	sshd
500	root	20	0	476444	1696	1000	R	0.7	0.1	779:34.67	NetworkMan+
7577	oracle	39	19	113288	1188	1000	R	0.7	0.1	173:51.18	test3.sh
7888	kolesni+	30	10	113416	1504	1184	S	0.7	0.1	48:18.34	sh

3. Керування станами процесів

3.1. У поточному терміналі виконуємо команду ping localhost, але не завершуйте її роботу.

```
64 bytes from localhost (127.0.0.1): icmp_seq=800 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=801 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=802 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=803 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=804 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=805 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=806 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=807 ttl=64 time=0.012 ms
64 bytes from localhost (127.0.0.1): icmp_seq=808 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=809 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=810 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=811 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=812 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=813 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=814 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=815 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=816 ttl=64 time=0.016 ms
64 bytes from localhost (127.0.0.1): icmp_seq=817 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=818 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=819 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=820 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=821 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=822 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=823 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=824 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=825 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=826 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=827 ttl=64 time=0.017 ms
64 bytes from localhost (127.0.0.1): icmp_seq=828 ttl=64 time=0.018 ms
```

3.2. Запускаємо другий термінал доступу до Linux-сервера.

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

3.4. У другому терміналі зупиняємо виконання процесу команди ping

- 3.5. У першому терміналі отримуємо список фонових процесів
- 3.6. У другому терміналі відновлюємо виконання припиненого процесу
- 3.7. У другому терміналі зупиняємо виконання процесу команди ping

```
[bodnar_illya@vpsj3IeQ ~]$ ps -u bodnar_illya -o pid,stat,cmd
  PID STAT  CMD
  1829 S+    ping localhost
  5209 S      sshd: bodnar_illya@pts/16
  5235 Ss    -bash
  6822 T      top
  9094 R+    ps -u bodnar_illya -o pid,stat,cmd
17009 S      sshd: bodnar_illya@pts/18
17135 Ss    -bash
[bodnar_illya@vpsj3IeQ ~]$ kill -19 1829
[bodnar_illya@vpsj3IeQ ~]$ ps -u bodnar_illya -o pid,stat,cmd
  PID STAT  CMD
  1829 T      ping localhost
  5209 S      sshd: bodnar_illya@pts/16
  5235 Ss+    -bash
  6822 T      top
17009 S      sshd: bodnar_illya@pts/18
17135 Ss    -bash
23235 R+    ps -u bodnar_illya -o pid,stat,cmd
[bodnar_illya@vpsj3IeQ ~]$ kill -18 1829
[bodnar_illya@vpsj3IeQ ~]$ kill 1829
[bodnar_illya@vpsj3IeQ ~]$ ps -u bodnar_illya -o pid,stat,cmd
  PID STAT  CMD
  5209 S      sshd: bodnar_illya@pts/16
  5235 Ss+    -bash
  6822 T      top
17009 R      sshd: bodnar_illya@pts/18
17135 Ss    -bash
21098 R+    ps -u bodnar_illya -o pid,stat,cmd
[bodnar_illya@vpsj3IeQ ~]$
```

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

```
[bodnar_illya@vpsj3IeQ ~]$ nohup ping localhost &
[3] 29292
[2] Terminated                  nohup ping localhost
[bodnar_illya@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'
[bodnar_illya@vpsj3IeQ ~]$
```

- 3.9. Закриваємо перший термінал.
- 3.10. У другому терміналі для команди ping отримуємо таблицю її процесу (колонки PID, STAT, CMD).

3.11. Завершаємо роботу процесу.

```
[bodnar_illya@vpsj3IeQ ~]$ ps -u bodnar_illya -o pid,stat,cmd
PID STAT CMD
11791 R+  ps -u bodnar_illya -o pid,stat,cmd
17009 S   sshd: bodnar_illya@pts/18
17135 Ss   -bash
29292 S   ping localhost
[bodnar_illya@vpsj3IeQ ~]$ kill 29292
[bodnar_illya@vpsj3IeQ ~]$ ps -u bodnar_illya -o pid,stat,cmd
PID STAT CMD
17009 S   sshd: bodnar_illya@pts/18
17135 Ss   -bash
31614 R+  ps -u bodnar_illya -o pid,stat,cmd
[bodnar_illya@vpsj3IeQ ~]$
```

4. Управління пріоритетами процесів

4.1. Створюємо bash-програму, що виконує операцію циклічного складання за формулою: $x = x + n$

```
bodnar_illya@vpsj3IeQ:~
GNU nano 2.3.1 File: bodnar.sh

bin!/bin/bash

x=6
n=4
while [ true ]
do
    x=$((x + $n))
done
```

4.2. Запускаємо bash-програму у фоновому режимі.

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

```
[bodnar_illya@vpsj3IeQ ~]$ nohup sh bodnar.sh&
[1] 10875
[bodnar_illya@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'

[bodnar_illya@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,%cpu,cmd
PIDppid,stat,ni,COMMANDpu,cmd
10875pid,ppid,stat,ni,shpu,cmd
17135pid,ppid,stat,ni,bashpu,cmd
32176pid,ppid,stat,ni,pspu,cmd
```

4.4. Виконуємо команду призупинення запущеного процесу.

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

4.6. Виконуємо команду продовження виконання припиненого процесу.

4.7. Ще раз переглядаємо таблицю процесів для процесу, який продовжив виконуватися, з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD.


```
[bodnar_illya@vpsj3Ieq ~]$ kill -18 10875
[bodnar_illya@vpsj3Ieq ~]$ ps -o pid,ppid,stat,ni,%cpu,cmd
  PID ppid stat  ni  %cpu  cmd
  5488   17135 R    0   0.0  ps   -o pid,ppid,stat,ni,%cpu,cmd
  10875  17135 R    0   0.0  sh   bodnar.sh
  17135  17135 R    0   0.0  sh   bodnar2.sh
  17135  17135 R    0   0.0  sh   bodnar3.sh
[bodnar_illya@vpsj3Ieq ~]$
```

- 4.8. 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

```
[bodnar_illya@vpsj3Ieq ~]$ ps -u bodnar_illya -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  7647  17135  R     0   28.2  sh bodnar.sh
  7889  27177  R+    0   0.0  ps  -u bodnar_illya -o pid,ppid,stat,ni,%cpu,cmd
 14014  17135  R     0  25.7  sh bodnar2.sh
 16180  17135  R     0  25.6  sh bodnar3.sh
 17009  16432  S     0   0.0  sshd: bodnar_illya@pts/18
 17135  17009  Ss+   0   0.0  -bash
 27141  23864  S     0   0.0  sshd: bodnar_illya@pts/12
 27177  27141  Ss    0   0.0  -bash
[bodnar_illya@vpsj3Ieq ~]$
```

- 4.9. Запускаємо два файли у фоновому режимі.
 4.10. Переглядаємо таблицю процесів для трьох запущених процесів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD.
 4.11. Зменшуємо пріоритет виконання одного з трьох процесів.

```
[bodnar_illya@vpsj3Ieq ~]$ renice -n 1 -p 17135
17135 (process ID) old priority 0, new priority 1
[bodnar_illya@vpsj3Ieq ~]$
```

- 4.12. Переглядаємо таблицю процесів для трьох запущених файлів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD.

```
[bodnar_illya@vpsj3Ieq ~]$ ps -u bodnar_illya -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  6736  27177  R+    0   0.0  ps  -u bodnar_illya -o pid,ppid,stat,ni,%cpu,cmd
  7647  17135  RN    1  27.5  sh bodnar.sh
 14014  17135  RN    1  24.2  sh bodnar2.sh
 16180  17135  R     0  28.0  sh bodnar3.sh
 17009  16432  S     0   0.0  sshd: bodnar_illya@pts/18
 17135  17009  Sns+  1   0.0  -bash
 27141  23864  S     0   0.0  sshd: bodnar_illya@pts/12
 27177  27141  Ss    0   0.0  -bash
[bodnar_illya@vpsj3Ieq ~]$
```

Висновок: У процесі виконання лабораторної роботи ми отримали навички в управлінні процесами в ОС Unix засобами командної оболонки.