

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА «ІНФОРМАЦІЙНИХ СИСТЕМ»

Лабораторна робота № 7
з дисципліни «Операційні
системи»

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

Виконав:

Студент групи AI-202
Лобко Даніїл Віталійович

Одеса-2021

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

Перелік завдань:

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.

Таблиця 3 – Варіанти завдань

№ команди	№ учасника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
1	1	6	1	0.5	15
1	2	7	1.5	0.7	20
2	1	8	2	0.9	25
2	2	9	1	0.5	30
3	1	10	1.5	0.7	35
3	2	11	2	0.9	40
4	1	12	1	0.5	45
4	2	13	1.5	0.7	50
5	1	14	2	0.9	60
5	2	6	1	0.5	65
6	1	7	1.5	0.7	60
6	2	8	2	0.9	55
7	1	9	1	0.5	50
7	2	10	1.5	0.7	45
8	1	11	2	0.9	40
8	2	12	1	0.5	35
9	1	13	1.5	0.7	30
9	2	14	2	0.9	25
10	1	8	2	0.9	20
10	2	9	1	0.5	15

Завдання 2 Перегляд таблиці процесів

1. Отримайте ієрархію всіх процесів із зазначенням імен користувачів, їх запустили.
2. Отримайте ієрархію процесів, запущених від імені вашого профілю і з зазначенням PID цих процесів.
3. Отримайте список процесів, запущених в поточному терміналі, зі спрощеним набором колонок виведення таблиці процесів.
4. Отримайте список процесів, запущених від імені вашого користувача, з розширеним набором колонок таблиці процесів.
5. Отримайте список процесів, запущених від імені вашого користувача із зазначенням наступного набору колонок: PID, TTY, PPID, STAT, NI, CMD
6. Отримайте список всіх сплячих процесів зі спрощеним набором колонок виведення таблиці процесів.
7. Отримайте список процесів, відсортованих по PID, і визначте:
 - a. загальна кількість запущених процесів;
 - b. кількість процесів, які виконуються;
 - c. кількість сплячих процесів.
8. Отримайте список процесів, відсортованих за % використання процесора.

Завдання 3 Керування станами процесів

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

роботу.

2. Запустіть другий термінал доступу до Linux-сервера.
3. У другому терміналі для команди `ring` отримаєте таблицю її процесу (колонки `PID, STAT, CMD`).
4. У другому терміналі призупиніть виконання процесу команди `ring`
5. У першому терміналі отримайте список фонових процесів
6. У другому терміналі відновіть виконання припиненого процесу
7. У другому терміналі зупиніть виконання процесу команди `ring`
8. У першому терміналі запустіть команду `ring` в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запусканий.
9. Закрийте перший термінал.
10. У другому терміналі для команди `ring` отримаєте таблицю її процесу (колонки `PID, STAT, CMD`). Зробіть висновок про стан процесу.
11. Завершіть роботу процесу.

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

2. Запустіть bash-програму у фоновому режимі.

3. Перегляньте таблицю процесів для запущеного процесу, пов'язаного з bash-

програмою, з урахуванням набором колонок: PID, PPID, STAT, NI,% CPU, CMD.

4. Виконайте команду призупинення запущеного процесу.

5. Ще раз перегляньте таблицю процесів для призупиненого процесу з урахуванням

набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан.

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.

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.

14

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.

11. Зменшить 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?

Хід 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.

Таблиця 3 – Варіанти завдань

№ команди	№ учасника команди	Пам'ять, <i>N</i> Гб	ОС, <i>M</i> Гб	Програма, <i>K</i> Гб	Очікування завершення вводу/виводу, <i>F</i>
1	1	6	1	0.5	15
1	2	7	1.5	0.7	20
2	1	8	2	0.9	25
2	2	9	1	0.5	30
3	1	10	1.5	0.7	35
3	2	11	2	0.9	40
4	1	12	1	0.5	45
4	2	13	1.5	0.7	50
5	1	14	2	0.9	60
5	2	6	1	0.5	65
6	1	7	1.5	0.7	60
6	2	8	2	0.9	55
7	1	9	1	0.5	50
7	2	10	1.5	0.7	45
8	1	11	2	0.9	40
8	2	12	1	0.5	35
9	1	13	1.5	0.7	30
9	2	14	2	0.9	25
10	1	8	2	0.9	20
10	2	9	1	0.5	15

4-2. $(13-1.5)/0.7 = \sim 16.4 \rightarrow$ одночасно можуть виконуватися 16 програм.

Отже, завантаженість процесора становить:

$1-0.5^{16} \approx 100\%$ При навантаженні 16 процесами.

$1-0.5^1 = 50\%$ при навантаженні одним процесом.

Завдання 2 Перегляд таблиці процесів

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

```
[lobko_daniil@vpsj3IeQ ~]$ pstree -up | less
systemd(1) -- NetworkManager(500) -- {NetworkManager}(544)
    |-- {NetworkManager}(550)
    |-- agetty(545)
    |-- auditd(413) --- {auditd}(414)
    |-- belobrov.sh(27717, belobrov_artur) --- belobrov.sh(19419)
    |-- belobrov2.sh(13532, belobrov_artur) --- belobrov2.sh(19421)
    |-- belobrov3.sh(14315, belobrov_artur) --- belobrov3.sh(19416)
    |-- chronyd(520, chrony)
    |-- crond(535)
    |-- dbus-daemon(473, dbus)
    |-- exim(802, exim)
    |-- gssproxy(482) -- {gssproxy}(492)
        |-- {gssproxy}(493)
        |-- {gssproxy}(494)
        |-- {gssproxy}(495)
        |-- {gssproxy}(496)
    |-- httpd(932) -- httpd(7203, apache)
        |-- httpd(8933, apache)
        |-- httpd(9780, apache)
        |-- httpd(10166, apache)
        |-- httpd(21432, apache)
        |-- httpd(25999, apache)
        |-- httpd(27377, apache)
        |-- httpd(27540, apache)
        |-- httpd(28772, apache)
        |-- httpd(29760, apache)
    |-- irqbalance(501)
    |-- mysqld(1157, mysql) -- {mysqld}(1158)
        |-- {mysqld}(1159)
        |-- {mysqld}(1160)
        |-- {mysqld}(1161)
        |-- {mysqld}(1162)
        |-- {mysqld}(1163)
        |-- {mysqld}(1164)
        |-- {mysqld}(1165)
        |-- {mysqld}(1166)
```

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

```
[lobko_daniil@vpsj3IeQ ~]$ pstree lobko_daniil | less
[lobko_daniil@vpsj3IeQ ~]$ pstree lobko_daniil -p
sshd(16512) --- bash(16533) --- pstree(4511)
[lobko_daniil@vpsj3IeQ ~]$
```

3. Отримайте список процесів, запущених в поточному терміналі, зі спрощеним

набором колонок виведення таблиці процесів.

```
lobko_daniil@vpsj3IeQ:~  
lobko_daniil@vpsj3IeQ ~]$ ps -o user,pid,ppid,TTY,s  
USER      PID  PPID TT      S  
lobko_d+ 16533 16512 pts/4    S  
lobko_d+ 25371 16533 pts/4    R  
lobko_daniil@vpsj3IeQ ~]$
```

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

```
workManager} (550)  
lobko_daniil@vpsj3IeQ:~  
)  
_art[lobko_daniil@vpsj3IeQ ~]$ ps -F  
v_aUID      PID  PPID  C    SZ    RSS  PSR  STIME TTY      TIME CMD  
v_alobko_d+ 16533 16512 0 28920 2172  1 01:17 pts/4    00:00:00 -bash  
lobko_d+ 30196 16533 0 38869 1876  1 01:26 pts/4    00:00:00 ps -F  
[lobko_daniil@vpsj3IeQ ~]$
```

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

```
lobko_d+ 30196 16533 0 38869 1876  1 01:26 pts/4    00:00:00 ps -F  
[lobko_daniil@vpsj3IeQ ~]$ ps -o pid,TTY,PPID,STAT,NI,CMD  
PID TT      PPID STAT  NI  CMD  
16533 pts/4    16512 Ss      0  -bash  
32306 pts/4    16533 R+      0  ps -o pid,TTY,PPID,STAT,NI,CMD  
[lobko_daniil@vpsj3IeQ ~]$
```

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

```
[lobko_daniil@vpsj3IeQ ~]$ clear
```

```
[lobko_daniil@vpsj3IeQ ~]$ ps -e s n
```

UID	PID	PENDING	BLOCKED	IGNORED	CAUGHT	STAT	TTY	TIME	COMMAND
0	1	0000000000000000	7be3c0fe28014a03	0000000000001000	00000001800004ec	Ss	?	41:39	/usr/lib/systemd
0	2	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:01	[kthreadd]
0	4	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kworker/0:0H]
0	6	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:55	[ksoftirqd/0]
0	7	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:09	[migration/0]
0	8	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:00	[rcu_bh]
0	9	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	R	?	144:03	[rcu_sched]
0	10	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[lru-add-drain]
0	11	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:17	[watchdog/0]
0	12	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:20	[watchdog/1]
0	13	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:05	[kintegrityd]
0	14	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	12:13	[ksoftirqd/1]
0	16	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kworker/1:0H]
0	18	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:00	[kdevtmpfs]
0	19	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[netns]
0	20	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:01	[khungtaskd]
0	21	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[writeback]
0	22	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kintegrityd]
0	23	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[bioset]
0	24	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[bioset]
0	25	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[bioset]
0	26	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kblockd]
0	27	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[md]
0	28	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[edac-poller]
0	29	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[watchdogd]
0	35	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	18:50	[kswapd0]
0	36	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	SN	?	0:00	[ksmd]
0	37	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[crypto]
0	45	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kthrotld]
0	47	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kmpath_rdacd]
0	48	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kaluad]
0	49	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[kpsmouse]
0	50	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[ip6_addrconf]
0	64	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[deferwq]
0	114	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	1:24	[kauditd]
0	247	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[ata_sff]
0	248	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[ttm_swap]
0	256	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:00	[scsi_ah_0]
0	257	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[scsi_tm_0]
0	258	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S	?	0:00	[scsi_ah_1]
0	259	0000000000000000	0000000000000000	ffffffffffffffff	0000000000000000	S<	?	0:00	[scsi_tm_1]

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

а. загальна кількість запущених процесів;

```
[lobko_daniil@vpsj3IeQ ~]$ ps -ef --sort pid | wc -l
```

```
220
```

```
[lobko_daniil@vpsj3IeQ ~]$ ps -ef --sort pid | wc -l
```

б. кількість процесів, які виконуються;

```
[lobko_daniil@vpsj3IeQ ~]$ ps -ef r -N --sort pid | wc -l
```

```
189
```

с. кількість сплячих процесів.

```
[lobko_daniil@vpsj3IeQ ~]$ ps -ef r --sort pid | wc -l
```

```
29
```

```
[lobko_daniil@vpsj3IeQ ~]$
```

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

```
411
[lobko_daniil@vpsj3IeQ ~]$ ps -ef --sort %cpu
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0  0 Feb16 ?        00:41:39 /usr/lib/systemd/systemd --switched-root --system --deserialize 22
root      2    0  0 Feb16 ?        00:00:01 [kthreadd]
root      4    2  0 Feb16 ?        00:00:00 [kworker/0:0H]
root      6    2  0 Feb16 ?        00:00:55 [ksoftirqd/0]
root      7    2  0 Feb16 ?        00:00:09 [migration/0]
root      8    2  0 Feb16 ?        00:00:00 [rcu_bh]
root     10    2  0 Feb16 ?        00:00:00 [lru-add-drain]
root     11    2  0 Feb16 ?        00:00:17 [watchdog/0]
root     12    2  0 Feb16 ?        00:00:20 [watchdog/1]
root     13    2  0 Feb16 ?        00:00:05 [migration/1]
root     14    2  0 Feb16 ?        00:12:13 [ksoftirqd/1]
root     16    2  0 Feb16 ?        00:00:00 [kworker/1:0H]
root     18    2  0 Feb16 ?        00:00:00 [kdevtmpfs]
root     19    2  0 Feb16 ?        00:00:00 [netns]
root     20    2  0 Feb16 ?        00:00:01 [khungtaskd]
root     21    2  0 Feb16 ?        00:00:00 [writeback]
root     22    2  0 Feb16 ?        00:00:00 [kintegrityd]
root     23    2  0 Feb16 ?        00:00:00 [bioset]
root     24    2  0 Feb16 ?        00:00:00 [bioset]
root     25    2  0 Feb16 ?        00:00:00 [bioset]
root     26    2  0 Feb16 ?        00:00:00 [kblockd]
root     27    2  0 Feb16 ?        00:00:00 [md]
root     28    2  0 Feb16 ?        00:00:00 [edac-poller]
root     29    2  0 Feb16 ?        00:00:00 [watchdogd]
root     35    2  0 Feb16 ?        00:18:50 [kswapd0]
root     36    2  0 Feb16 ?        00:00:00 [ksmd]
root     37    2  0 Feb16 ?        00:00:00 [crypto]
root     45    2  0 Feb16 ?        00:00:00 [kthrotld]
root     47    2  0 Feb16 ?        00:00:00 [kmpath_rdacd]
root     48    2  0 Feb16 ?        00:00:00 [kaluad]
root     49    2  0 Feb16 ?        00:00:00 [kpsmoused]
root     50    2  0 Feb16 ?        00:00:00 [ipv6_addrconf]
root     64    2  0 Feb16 ?        00:00:00 [deferwq]
root    114    2  0 Feb16 ?        00:01:24 [kauditd]
root    247    2  0 Feb16 ?        00:00:00 [ata_sff]
root    248    2  0 Feb16 ?        00:00:00 [ttm_swap]
root    256    2  0 Feb16 ?        00:00:00 [scsi_eh_0]
root    257    2  0 Feb16 ?        00:00:00 [scsi_tmf_0]
```

Завдання 3 Керування станами процесів

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

```
[lobko_daniil@vpsj3IeQ ~]$ ping localhost
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=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.025 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=0.025 ms
64 bytes from localhost (127.0.0.1): icmp_seq=12 ttl=64 time=0.027 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```
[lobko_daniil@vpsj3IeQ ~]$ ps -C ping -o pid,stat,cmd
  PID STAT  CMD
  3110 S      ping localhost
  3575 S      ping localhost
  4317 S      ping localhost
  7126 S      ping localhost
  7817 S      ping localhost
 12757 S      ping localhost
 12820 S      ping localhost
 17416 S      ping localhost
 21724 S      ping localhost
 22468 S      ping localhost
 26589 S      ping localhost
 28415 S      ping localhost
 28743 S      ping localhost
 30185 S+     ping localhost
 30450 S      ping localhost
 30737 S      ping localhost
 30973 S      ping localhost
 31225 S      ping localhost
[lobko_daniil@vpsj3IeQ ~]$
```

4. У другому терміналі призупиніть виконання процесу команди ping

```
[1]+  Stopped                  ping localhost
[lobko_daniil@vpsj3IeQ ~]$
```

```
lobko_d+ 12902 0.0 0.1 115680 2180 pts/17  Ss  01:59  0:00 -bash
lobko_d+ 16533 0.0 0.1 115680 2196 pts/4   Ss  01:17  0:00 -bash
lobko_d+ 31603 0.0 0.1 115680 2176 pts/10  Ss+ 01:39  0:00 -bash
[lobko_daniil@vpsj3IeQ ~]$ kill -19 7208
[lobko_daniil@vpsj3IeQ ~]$
```

5. У першому терміналі отримайте список фонових процесів

```
[1]+  Stopped                  ping localhost
[lobko_daniil@vpsj3IeQ ~]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
lobko_d+  7208  0.0  0.0 130680 1640 pts/4    T   02:06    0:00 ping localhost
lobko_d+  9941  0.0  0.0 155476 1880 pts/4    R+  02:07    0:00 ps -u
lobko_d+ 12902  0.0  0.1 115680 2180 pts/17  Ss+  01:59    0:00 -bash
lobko_d+ 16533  0.0  0.1 115680 2200 pts/4    Ss   01:17    0:00 -bash
lobko_d+ 31603  0.0  0.1 115680 2176 pts/10  Ss+  01:39    0:00 -bash
[lobko_daniil@vpsj3IeQ ~]$
```

6. У другому терміналі відновіть виконання припиненого процесу

```

lobko_d+ 12902  0.0  0.1 115680  2180 pts/17   Ss   01:59   0:00 -bash
lobko_d+ 16533  0.0  0.1 115680  2196 pts/4     Ss   01:17   0:00 -bash
lobko_d+ 31603  0.0  0.1 115680  2176 pts/10   Ss+  01:39   0:00 -bash
[lobko_daniil@vpsj3IeQ ~]$ kill -19 7208
[lobko_daniil@vpsj3IeQ ~]$ kill -18 7208
[lobko_daniil@vpsj3IeQ ~]$ 
lobko_d+ 16533  0.0  0.1 115680  2200 pts/4     Ss   01:17   0:00 -bash
lobko_d+ 31603  0.0  0.1 115680  2176 pts/10   Ss+  01:39   0:00 -bash
[lobko_daniil@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64
4 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=16 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=17 ttl=64 time=0.035 ms
64 bytes from localhost (127.0.0.1): icmp_seq=18 ttl=64 time=0.037 ms
64 bytes from localhost (127.0.0.1): icmp_seq=19 ttl=64 time=0.035 ms

```

7. У другому терміналі зупиніть виконання процесу команди ping
8. У першому терміналі запустіть команду ping в фоновому режимі так, щоб він не був автоматично зупинений навіть після закриття терміналу, з якого був запущений.

```

[1]+  Stopped                  nohup ping localhost
[lobko_daniil@vpsj3IeQ ~]$ ping localhost &
[2] 11289
[lobko_daniil@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=0.013 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.013 ms

```

```

[1]+  Stopped                  ping localhost

```

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

```
lobko_d+ 31603 0.0 0.1 115680 2176 pts/10 Ss+ 01:39 0:00 -bash
[lobko_daniil@vpsj3IeQ ~]$ nohup ping localhost &
[1] 26623
[lobko_daniil@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.
out'
ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
lobko_d+ 16225  0.0  0.1 115680 2176 pts/12  Ss   02:17   0:00 -bash
lobko_d+ 26623  0.0  0.0 130680 1640 pts/12  S    02:19   0:00 ping localhost
lobko_d+ 26768  0.0  0.0 155476 1876 pts/12  R+   02:19   0:00 ps -u
lobko_d+ 31603  0.0  0.1 115680 2176 pts/10  Ss+  01:39   0:00 -bash
[lobko_daniil@vpsj3IeQ ~]$
```

11. Завершіть роботу процесу.

```
[lobko_daniil@vpsj3IeQ ~]$ kill -19 26623
[lobko_daniil@vpsj3IeQ ~]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
lobko_d+  9305  0.0  0.0 155472 1816 pts/12  R+   02:23   0:00 ps -u
lobko_d+ 16225  0.0  0.1 115680 2176 pts/12  Ss   02:17   0:00 -bash
lobko_d+ 26623  0.0  0.0 130680 1640 pts/12  T    02:19   0:00 ping localhost
lobko_d+ 31603  0.0  0.1 115680 2176 pts/10  Ss+  01:39   0:00 -bash

[1]+  Stopped                  nohup ping localhost
[lobko_daniil@vpsj3IeQ ~]$
```

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

```
lobko_daniil@vpsj3IeQ:~/lobko_lab_7
GNU nano 2.3.1 File: lobko.sh

x=6
y=5
while [ true ]
do
    x=$((x+y))
done
```

2. Запустіть bash-програму у фоновому режимі.

```
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ nano lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ sh lobko.sh &
[1] 14374
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

3. Перегляньте таблицю процесів для запущеного процесу, пов'язаного з bash-

програмою, з урахуванням набором колонок: PID, PPID, STAT, NI,% CPU, CMD.

```
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
12413 12388  Ss      0   -  -bash
14374 12413  R       0   -  sh lobko.sh
22012 12413  R+      0   -  ps -o pid,ppid,stat,ni,cpu,cmd
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

4. Виконайте команду призупинення запущеного процесу.

```
lobko_d+ 31603  0.0  0.1 115680  2176 pts/10  Ss+  01:39   0:00 -
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ sh lobko.sh &
[1] 31885
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ kill -19 31885

[1]+  Stopped                  sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

5. Ще раз перегляньте таблицю процесів для призупиненого процесу з урахуванням

набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробіть висновки про його стан.


```
[1]+  Stopped                  sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
  6649 29341 R+      0   - ps -o pid,ppid,stat,ni,cpu,cmd
 29341 29311 Ss      0   - -bash
 31885 29341 T       0   - sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

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.

```
9341 29311 Ss      0   - -bash
1885 29341 T       0   - sh lobko.sh
lobko_daniil@vpsj3IeQ lobko_lab_7]$ kill -18 31885
lobko_daniil@vpsj3IeQ lobko_lab_7]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID STAT  NI CPU CMD
  9427 29341 R+      0   - ps -o pid,ppid,stat,ni,cpu,cmd
  9341 29311 Ss      0   - -bash
 1885 29341 R       0   - sh lobko.sh
lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

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.

```
29341 29311 Ss      0   - -bash
31885 29341 R       0   - sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ln -s lobko.sh lobko2.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ln -s lobko.sh lobko3.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ sh lobko2.sh &
[2] 18109
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ sh lobko3.sh &
[3] 18331
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
lobko_d+  4775  0.0  0.1 115680 2172 pts/4    Ss   02:28   0:00 -bash
lobko_d+  5181  0.0  0.1 115480 2004 pts/4    S+   02:33   0:00 nano /usr/share/nano/mysettings.nanorc
lobko_d+ 18109 12.1  0.0 113284 1184 pts/6    R    03:03   0:00 sh lobko2.sh
lobko_d+ 18331 15.6  0.0 113284 1184 pts/6    R    03:04   0:00 sh lobko3.sh
lobko_d+ 18657  0.0  0.0 155476 1876 pts/6    R+   03:04   0:00 ps -u
lobko_d+ 29341  0.0  0.1 115684 2216 pts/6    Ss   03:00   0:00 -bash
lobko_d+ 31603  0.0  0.1 115680 2176 pts/10   Ss+  01:39   0:00 -bash
lobko_d+ 31885 11.0  0.0 113284 1184 pts/6    R    03:00   0:21 sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```


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.

```
lobko_d+ 31885 11.0 0.0 113284 1184 pts/6 R 03:00 0:21 sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
 18109 29341 R      0   -  sh lobko2.sh
 18331 29341 R      0   -  sh lobko3.sh
 21371 29341 R+     0   -  ps -o pid,ppid,stat,ni,cpu,cmd
 29341 29311 Ss     0   -  -bash
 31885 29341 R      0   -  sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

8. Створіть два файли як символічні посилання на створену б

11. Зменшить пріоритет виконання одного з трьох процесів.

```
31885 (process ID) old priority 5, new priority 5
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ renice -n 10 -p 29341
29341 (process ID) old priority 5, new priority 10
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

```
nzhelika@vpsj3IeQ ~]$ ps -o pid,ppid,stat,ni,%cpu,cmd 12346 6630 8143
  STAT  NI  %CPU  CMD
  R      0  8.0  sh grebenik2.sh
  R      0  7.7  sh grebenik3.sh
```

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?

```
29341 (process ID) old priority 5, new priority 10
[lobko_daniil@vpsj3IeQ lobko_lab_7]$ ps -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  4622 29341 RN+    10   -  ps -o pid,ppid,stat,ni,cpu,cmd
 18109 29341 RN      5   -  sh lobko2.sh
 18331 29341 RN      5   -  sh lobko3.sh
 29341 29311 SNs    10   -  -bash
 31885 29341 RN      5   -  sh lobko.sh
[lobko_daniil@vpsj3IeQ lobko_lab_7]$
```

Висновки: Виконуючи цю лабораторну роботи ми закріпили навички роботи з управління процесами в Linux.

Найскладнішим було завдання 3 через його запутаність.