

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

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

Сідельніков М. В.

Перевірив:

Блажко О. А.

Одеса-2021

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

Завдання:

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.

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:
 - а. загальна кількість запущених процесів;

- б. кількість процесів, які виконуються;
 - с. 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 `ping` отримаєте таблицю її процесу (колонки PID, STAT, CMD).
4. У другому терміналі призупиніть виконання процесу команди `ping`.
5. У першому терміналі отримайте список фонових процесів
6. У другому терміналі відновіть виконання припиненого процесу
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.
9. Закрийте перший термінал.
10. У другому терміналі для команди `ping` 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.
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.

$N = 7; M = 1.5; K = 0.7; F = 60.$

Обсяг пам'яті що залишається на сторонні програми: $N - M = 5.5$ (Гб);

Тому максимальна к-сть процесів, що працюють одночасно дорівнює:

$5.5 / K = 7.$

Тоді середня завантаженість процесора дорівнює: $1 - 0.6^7 = 97.2\%.$

Завдання 2. Перегляд таблиці процесів.

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

```
sidelnikov_mikita@vpsj3IeQ:~  
login as: sidelnikov_mikita  
sidelnikov_mikita@91.219.60.189's password:  
Last login: Tue Apr 6 07:47:23 2021 from 31.31.105.118  
[sidelnikov_mikita@vpsj3IeQ ~]$ clear  
[sidelnikov_mikita@vpsj3IeQ ~]$ pstree -u  
systemd--NetworkManager--2*[{NetworkManager}]  
--agetty  
--auditd--{auditd}  
--belobrov.sh(belobrov_artur)  
--belobrov2.sh(belobrov_artur)--belobrov2.sh  
--belobrov3.sh(belobrov_artur)  
--chronyd(chrony)  
--crond  
--dbus-daemon(dbus)  
--exim(exim)  
--gssproxy--5*[{gssproxy}]  
--httpd--10*[{httpd(apache)}]  
--irqbalance  
--mysqld(mysql)--30*[{mysqld}]  
--named(named)--4*[{named}]  
--nano(bogachik_egor)  
--nano(baranyuk_dmitro)  
--nesterenko.sh(nesterenko_mikola)--nesterenko.sh  
--nesterenko2.sh(nesterenko_mikola)  
--nesterenko3.sh(nesterenko_mikola)  
--nginx--nginx(emps)  
--php-fpm--6*[{php-fpm(soft)}]  
--php-fpm--3*[{php-fpm}]  
--3*[{php-fpm(emps)}]  
--3*[{php-fpm(soft)}]  
--ping(oracle)  
--ping(malofeev_denis)  
--ping(stepanenko_gleb)  
--ping(kovach_dmitro)  
--6*[{ping(markovskij_danilo)}]  
--ping(sinyakov_igor)  
--2*[{ping(evchev_denis)}]  
--ping(kostetskiy_bogdan)  
--ping(maksimenko_andrij)  
--ping(bojchuk_oleksandr)  
--polkitd(polkitd)--6*[{polkitd}]  
--pure-ftpd  
--rpcbind(rpc)  
--rsyslogd--2*[{rsyslogd}]  
--3*[{sh(bodnar_illya)}]
```

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ pstree -p sidelnikov_mikita  
sshd(4588)--bash(4601)--pstree(10290)  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps  
  PID TTY          TIME CMD  
 4601 pts/10    00:00:00 bash  
21870 pts/10    00:00:00 ps  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps  
  PID TTY          TIME CMD  
 4601 pts/10    00:00:00 bash  
21870 pts/10    00:00:00 ps  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -u sidelnikov_mikita -F  
UID      PID  PPID  C  SZ  RSS  PSR  STIME TTY          TIME CMD  
sidelni+ 4588  3483  0 41473 2480   1 00:10 ?              00:00:00 sshd: sidelnikov  
sidelni+ 4601  4588  0 28920 2152   1 00:10 pts/10      00:00:00 -bash  
sidelni+ 24855 4601  1 38869 1872   1 00:13 pts/10      00:00:00 ps -u sidelnikov  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

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

```
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -u sidelnikov_mikita -o pid, tty, ppid, stat, ni, cmd  
  PID TT          PPID STAT  NI CMD  
 4588 ?              3483 S      0 sshd: sidelnikov_mikita@pts/10  
 4601 pts/10        4588 Ss     0 -bash  
30889 pts/10        4601 R+     0 ps -u sidelnikov_mikita -o pid, tty, ppid, stat, ni, cmd  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -x | grep [SD]  
  PID TTY          STAT      TIME COMMAND  
 4060 pts/10    S+         0:00 grep --color=auto [SD]  
 4588 ?          S          0:00 sshd: sidelnikov_mikita@pts/10  
 4601 pts/10    Ss         0:00 -bash
```

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

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

```
top - 00:18:41 up 56 days, 7:21, 17 users, load average: 33.83, 33.18, 30.17
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
31723	kolesni+	20	0	113416	1504	1184	S	0.0	0.1	61:52.31	sh
31225	markovs+	20	0	130680	1644	1268	S	0.0	0.1	0:01.87	ping
30973	markovs+	20	0	130680	1640	1264	S	0.0	0.1	0:01.82	ping
30769	baranyu+	20	0	115688	2156	1676	S	0.0	0.1	0:00.16	bash
30737	markovs+	20	0	130680	1644	1268	S	0.0	0.1	0:01.89	ping
30729	baranyu+	20	0	165888	2604	1092	S	0.0	0.1	0:00.42	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	0:00.51	ping
28415	evchev_+	20	0	130680	1644	1264	S	0.0	0.1	0:15.79	ping
28265	kovach_+	20	0	130680	1644	1268	T	0.0	0.1	0:00.01	ping
27717	belobro+	20	0	113416	1496	1176	S	0.0	0.1	76:24.09	belobrov.sh
27573	soft	30	10	279912	5080	828	S	0.0	0.3	0:00.00	php-fpm
27572	soft	30	10	279912	5080	828	S	0.0	0.3	0:00.00	php-fpm
27571	soft	30	10	279912	5080	828	S	0.0	0.3	0:00.00	php-fpm
27570	emps	30	10	279912	5080	828	S	0.0	0.3	0:00.00	php-fpm
27569	emps	30	10	279912	5080	828	S	0.0	0.3	0:00.00	php-fpm
27568	emps	30	10	279912	5080	828	S	0.0	0.3	0:00.00	php-fpm
27567	root	30	10	279912	4920	672	S	0.0	0.3	0:00.00	php-fpm
27566	root	30	10	279912	4920	672	S	0.0	0.3	0:00.00	php-fpm
27564	root	30	10	279936	5364	1112	S	0.0	0.3	0:10.73	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	0:00.12	httpd
27377	apache	20	0	113480	3112	1592	S	0.0	0.2	0:00.05	httpd
26589	malofee+	20	0	130680	1640	1264	S	0.0	0.1	0:01.18	ping
26319	barkar_+	20	0	115548	2120	1660	S	0.0	0.1	0:00.07	bash
26295	barkar_+	20	0	165892	2588	1076	S	0.0	0.1	0:00.15	sshd
25999	apache	20	0	113480	3108	1588	S	0.0	0.2	0:00.05	httpd
24552	kolesni+	20	0	113416	1504	1184	S	0.3	0.1	63:02.16	sh
24066	root	20	0	165892	6628	5116	S	0.0	0.4	0:00.32	sshd
23418	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0
23286	root	30	10	279912	4920	668	S	0.0	0.3	0:00.00	php-fpm

- 239
- 196
- 7

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


```
sidelnikov_mikita@vpsj3leQ:~  
top - 00:28:10 up 56 days, 7:31, 17 users, load average: 33.69, 33.53, 31.72  
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
29017	root	20	0	161532	5844	4476	S	0.7	0.3	0:00.02	sshd
11578	nesteren	25	5	113416	1496	1176	S	0.3	0.1	68:07.25	nesterenko2.sh
11579	nesteren	20	0	113416	1496	1176	S	0.3	0.1	73:27.20	nesterenko3.sh
13532	belobrov	20	0	113416	1496	1176	S	0.3	0.1	76:01.48	belobrov2.sh
14520	nesteren	20	0	113416	1496	1176	S	0.3	0.1	73:17.22	nesterenko.sh
14756	barkar_+	20	0	162276	2448	1600	S	0.3	0.1	0:03.28	top
24552	kolesni+	20	0	113416	1504	1184	S	0.3	0.1	63:03.45	sh
27717	belobrov	20	0	113416	1496	1176	S	0.3	0.1	76:25.36	belobrov.sh
30737	markovs+	20	0	130680	1644	1268	S	0.3	0.1	0:01.94	ping
1	root	20	0	191304	3124	1664	S	0.0	0.2	41:37.41	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:01.37	kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:55.16	ksoftirqd/0

Завдання 3. Керування станами процесів.

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

```
sidelnikov_mikita@vpsj3leQ:~  
[sidelnikov_mikita@vpsj3leQ ~]$ 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=0.025 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.022 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.022 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.
3. У другому терміналі для команди `ping` отримаєте таблицю її процесу (колонки PID, STAT, CMD).

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -u sidelnikov_mikita -F  
UID          PID    PPID  C   SZ   RSS  PSR  STIME TTY          TIME CMD  
sidelni+ 7388 23329  0 32670 1640  1 00:49 pts/17  00:00:00 ping localhost  
sidelni+ 12313 23339  0 38869 1876  1 00:50 pts/21  00:00:00 ps -u sidelnikov  
sidelni+ 23307 21093  0 41473 2480  1 00:33 ?        00:00:00 sshd: sidelnikov  
sidelni+ 23320 21889  0 41473 2484  1 00:46 ?        00:00:00 sshd: sidelnikov  
sidelni+ 23329 23307  0 28920 2156  1 00:33 pts/17  00:00:00 -bash  
sidelni+ 23339 23320  0 28920 2156  1 00:46 pts/21  00:00:00 -bash  
sidelni+ 26312 23329  0 32670 1640  0 00:46 pts/17  00:00:00 ping localhost  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -p 26312 -o pid,stat,cmd  
PID STAT CMD  
26312 T ping localhost  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

4. У другому терміналі призупиніть виконання процесу команди ping.

```
[sidelnikov_mikita@vpsj3IeQ ~]$ kill -19 26312
```

5. У першому терміналі отримайте список фонових процесів

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -x | grep [SD]  
PID TTY STAT TIME COMMAND  
4588 ? S 0:00 sshd: sidelnikov_mikita@pts/10  
4601 pts/10 Ss 0:00 -bash  
9657 pts/10 S+ 0:00 grep --color=auto [SD]  
23307 ? S 0:00 sshd: sidelnikov_mikita@pts/17  
23329 pts/17 Ss+ 0:00 -bash  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

6. У другому терміналі відновіть виконання припиненого процесу

```
[sidelnikov_mikita@vpsj3IeQ ~]$ kill -18 26312
```

7. У другому терміналі зупиніть виконання процесу команди ping

```
[sidelnikov_mikita@vpsj3IeQ ~]$ kill -2 26312
```

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.
9. Закрийте перший термінал.

```
[sidelnikov_mikita@vpsj3IeQ ~]$ nohup ping localhost &  
[1] 27204  
[sidelnikov_mikita@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ ps -p 27204 -o pid,stat,cmd  
  PID STAT  CMD  
27204 S      ping localhost  
[sidelnikov_mikita@vpsj3IeQ ~]$
```

Процес має статус `S` (sleep), тобто спить.

11. Завершіть роботу процесу.

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ kill -9 27204  
[sidelnikov_mikita@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`

```
sidelnikov_mikita@vpsj3IeQ:~/lab6_files
GNU nano 2.3.1 File: sidelnykov.sh

#!/bin/bash

x=10 #sidelnykov(10)
while [ true ];do
    x=$(( $x + 6 )) #nikita(6)
done

[Wrote 6 lines]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

2. Запустіть bash-програму у фоновому режимі.

```
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ nano sidelnykov.sh
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ bash sidelnykov.sh &
[1] 4140
[sidelnikov_mikita@vpsj3IeQ lab6_files]$
```

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

```
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ ps -p 4140 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  4140  23329  R      0  10.7  bash sidelnykov.sh
```

4. Виконайте команду призупинення запущеного процесу.

```
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ kill -19 4140
```

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

```
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ ps -p 4140 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  4140 23329  T       0  10.3  bash sidelnikov.sh

[1]+  Stopped                  bash sidelnikov.sh
```

Програма зупинена.

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.

```
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ kill -18 4140
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ ps -p 4140 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  4140 23329  R       0   9.9  bash sidelnikov.sh
```

Процес має статус R (running), тобто виконується.

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.

```
sidelnikov_mikita@vpsj3IeQ:~/lab6_files
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ ln -s sidelnikov.sh sidelnikov1.sh
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ ln -s sidelnikov.sh sidelnikov2.sh
[sidelnikov_mikita@vpsj3IeQ lab6_files]$ ls
sidelnikov1.sh  sidelnikov2.sh  sidelnikov.sh
[sidelnikov_mikita@vpsj3IeQ lab6_files]$
```

9. Запустіть два файли у фоновому режимі.

```
sidelnikov_mikita@vpsj3leQ:~/lab6_files
[sidelnikov_mikita@vpsj3leQ lab6_files]$ bash sidelnikov1.sh &
[2] 6722
[sidelnikov_mikita@vpsj3leQ lab6_files]$ bash sidelnikov2.sh &
[3] 6954
[sidelnikov_mikita@vpsj3leQ lab6_files]$
```

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.

```
sidelnikov_mikita@vpsj3leQ:~/lab6_files
[sidelnikov_mikita@vpsj3leQ lab6_files]$ ps -p 4140,6722,6954 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  4140  23329  R       0  10.3  bash sidelnikov.sh
  6722  23329  R       0   8.6  bash sidelnikov1.sh
  6954  23329  R       0   8.5  bash sidelnikov2.sh
[sidelnikov_mikita@vpsj3leQ lab6_files]$
```

Пріоритет однаковий, % ЦП трохи більше у першого процесу, але вони на приблизно однаковому рівні.

11. Зменшить пріоритет виконання одного з трьох процесів.

```
sidelnikov_mikita@vpsj3leQ:~/lab6_files
[sidelnikov_mikita@vpsj3leQ lab6_files]$ renice -n 5 6722
6722 (process ID) old priority 0, new priority 5
```

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?

```
sidelnikov_mikita@vpsj3leQ:~/lab6_files
[sidelnikov_mikita@vpsj3leQ lab6_files]$ ps -p 4140,6722,6954 -o pid,ppid,stat,ni,%cpu,cmd
  PID  PPID  STAT  NI  %CPU  CMD
  4140  23329  R       0  10.4  bash sidelnikov.sh
  6722  23329  RN       5   6.9  bash sidelnikov1.sh
  6954  23329  R       0   8.7  bash sidelnikov2.sh
[sidelnikov_mikita@vpsj3leQ lab6_files]$
```

Для процесу з меншим пріоритетом стало відводитися менше ресурсів процесора.

Висновки: під час виконання лабораторної роботи було придбано навичок в управлінні процесами в ОС Unix засобами командної оболонки.