

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:
ст. гр. АІ-204
Нестеренко А. О.

Перевірив:
Блажко О. А.
Дрозд М.О.

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

Завдання для виконання:

1. 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
6	2	8	2	0.9	55

2.

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

3.

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 другому 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 запусканий.

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

4.

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:

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. Визначимо середню завантаженість процесора, використовуючи значення з таблиці 3 варіантів.

№ команди	№ учасника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
6	1	7	1.5	0.7	60

За формулою $1 - p^n$, де p -ічікування завершення вводу\виводу, n - макс кіль-сть процесів

Кільсть процесів: $(7-1.5):0.5=11$

2 Перегляд таблиці процесів

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ pstree -u
systemd-+-NetworkManager--- 2*[{NetworkManager}]
        +-agetty
        +-auditd--- {auditd}
        +-belobrov.sh(belobrov_artur)--- belobrov.sh
        +-belobrov2.sh(belobrov_artur)--- belobrov2.sh
        +-belobrov3.sh(belobrov_artur)--- belobrov3.sh
        +-chronyd(chrony)
        +-crond
        +-dbus-daemon(dbus)
        +-exim(exim)
        +-gssproxy--- 5*[{gssproxy}]
        +-httpd--- 8*[httpd(apache)]
        +-irqbalance
        +-mysqld(mysql)--- 30*[{mysqld}]
        +-named(named)--- 4*[{named}]
        +-nano(bogachik_egor)
        +-nano(baranyuk_dmitro)
        +-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(bojchuk_oleksandr)
        +-polkitd(polkitd)--- 6*[{polkitd}]
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ pstree -p nesterenko_mikola
sshd(5408)--- bash(5578)--- pstree(19845)
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
 5578 pts/15      00:00:00 bash
28124 pts/15      00:00:00 ps
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -F -u nesterenko_mikola
UID          PID  PPID  C   SZ   RSS  PSR  STIME  TTY          TIME CMD
nester+  5408 23022  0 40939 2544   1 00:38  ?            00:00:00 sshd: nesterenko_mikola@pts/15
nester+  5578  5408  0 28887 2112   1 00:38 pts/15       00:00:00 -bash
nester+ 11890  5578  0 38869 1872   0 00:40 pts/15       00:00:00 ps -F -u nesterenko_mikola
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -u nesterenko_mikola -o pid, tty, ppid, stat, ni, cmd
PID TT      PPID STAT  NI CMD
4913 pts/15    5578 R+     0 ps -u nesterenko_mikola -o pid, tty, ppid, stat, ni, cmd
5408 ?        23022 S       0 sshd: nesterenko_mikola@pts/15
5578 pts/15    5408 Ss      0 -bash
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -ax | awk ' $3 == "S" '
 2 ?      S      0:01 [kthreadd]
 6 ?      S      0:24 [ksoftirqd/0]
 7 ?      S      0:09 [migration/0]
 8 ?      S      0:00 [rcu_bh]
11 ?      S      0:17 [watchdog/0]
12 ?      S      0:19 [watchdog/1]
13 ?      S      0:05 [migration/1]
14 ?      S     12:11 [ksoftirqd/1]
18 ?      S      0:00 [kdevtmpfs]
20 ?      S      0:01 [khungtaskd]
35 ?      S     18:50 [kswapd0]
114 ?     S      1:21 [kauditd]
256 ?     S      0:00 [scsi_eh_0]
258 ?     S      0:00 [scsi_eh_1]
273 ?     S      2:15 [jbd2/vda1-8]
520 ?     S      0:05 /usr/sbin/chronyd
883 ?     S      0:00 php-fpm: pool index
884 ?     S      0:00 php-fpm: pool index
885 ?     S      0:00 php-fpm: pool index
886 ?     S      0:00 php-fpm: pool regular
887 ?     S      0:00 php-fpm: pool regular
```

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

а. загальна кількість запущених процесів;172

б. кількість процесів, які виконуються;13

с. кількість сплячих процесів.158

```
top - 00:44:19 up 54 days, 7:47, 10 users, load average: 11.02, 11.99, 11.22
Tasks: 172 total, 13 running, 158 sleeping, 1 stopped, 0 zombie
%Cpu(s): 83.7 us, 14.0 sy, 1.9 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.4 si, 0.0 st
KiB Mem : 1881856 total, 169280 free, 336560 used, 1376016 buff/cache
KiB Swap: 4194300 total, 3715068 free, 479232 used, 1420068 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
32101	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.25	sshd
31453	berisla+	20	0	115548	2028	1600	S	0.0	0.1	0:00.00	bash
31397	berisla+	20	0	165892	2476	964	S	0.0	0.1	0:00.00	sshd
31186	veselko+	20	0	130680	1640	1264	T	0.0	0.1	0:00.00	ping
31122	babich_+	20	0	113416	1496	1176	S	1.5	0.1	0:16.61	babich3.sh
31121	babich_+	25	5	113416	1492	1176	S	1.1	0.1	0:14.02	babich2.sh
30863	root	20	0	165892	6632	5116	S	0.0	0.4	0:00.27	sshd
29760	apache	20	0	113480	3108	1580	S	0.0	0.2	0:00.03	httpd
28885	apache	20	0	113480	3096	1576	S	0.0	0.2	0:00.04	httpd
28489	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/0:3
28040	belobro+	20	0	113416	320	0	R	0.0	0.0	0:00.00	belobrov.sh
28039	babich_+	20	0	113416	320	0	R	0.0	0.0	0:00.00	babich.sh
28038	belobro+	30	10	113416	316	0	R	0.0	0.0	0:00.00	belobrov3.sh
28037	babich_+	25	5	113416	316	0	R	0.0	0.0	0:00.00	babich2.sh
28036	belobro+	20	0	113416	320	0	R	0.0	0.0	0:00.00	belobrov2.sh
28034	babich_+	20	0	113416	320	0	R	0.0	0.0	0:00.00	babich3.sh
28027	root	20	0	127968	524	312	S	0.0	0.0	0:00.00	pure-ftpd
28020	root	20	0	127968	516	376	S	0.0	0.0	0:00.00	pure-ftpd
27845	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kworker/1:3

8. Отримайте список процесів, відсортованих за % використання процесора

```
top - 01:29:09 up 54 days, 8:32, 12 users, load average: 9.93, 10.07, 10.27
Tasks: 170 total, 10 running, 160 sleeping, 0 stopped, 0 zombie
%Cpu(s): 83.9 us, 13.9 sy, 2.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 1881856 total, 177048 free, 336244 used, 1368564 buff/cache
KiB Swap: 4194300 total, 3715068 free, 479232 used, 1423800 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7548	oracle	20	0	113288	1188	1004	R	87.0	0.1	3565.30	test1.sh
7574	oracle	20	0	113288	1188	1004	R	85.0	0.1	3563.08	test2.sh
25939	babich_+	20	0	113416	1496	1176	S	1.3	0.1	1:01.58	babich.sh
27717	belobro+	20	0	113416	1496	1176	S	1.3	0.1	1:07.90	belobrov.sh
7577	oracle	39	19	113288	1188	1000	R	1.0	0.1	53:45.92	test3.sh
13532	belobro+	20	0	113416	1496	1176	S	1.0	0.1	1:00.97	belobrov2.sh
31121	babich_+	25	5	113416	1492	1176	S	1.0	0.1	0:43.08	babich2.sh
31122	babich_+	20	0	113416	1496	1176	S	1.0	0.1	0:51.36	babich3.sh
14315	belobro+	30	10	113416	1492	1176	S	0.7	0.1	0:42.68	belobrov3.sh
9	root	20	0	0	0	0	S	0.3	0.0	141:04.54	rcu_sched
500	root	20	0	476444	1696	1000	S	0.3	0.1	771:15.84	NetworkManager
1	root	20	0	191168	3004	1668	S	0.0	0.2	40:32.88	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:01.35	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	0:24.46	ksoftirqd/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:09.86	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:17.05	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:19.83	watchdog/1

3 Керування станами процесів

1. У поточному терміналі виконаємо команду ping localhost, але не завершуємо її роботу.

```
[nesterenko_mikola@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=0.023 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.036 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=10 ttl=64 time=0.053 ms
64 bytes from localhost (127.0.0.1): icmp_seq=11 ttl=64 time=0.030 ms
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 636 -o pid,stat,cmd
  PID STAT  CMD
   636 S+    ping localhost
[nesterenko_mikola@vpsj3IeQ ~]$
```

4. У другому терміналі призупинимо виконання процесу команди ping

```
6[nesterenko_mikola@vpsj3IeQ ~]$ kill -19 636
6[nesterenko_mikola@vpsj3IeQ ~]$
64 bytes from localhost (127.0.0.1): icmp_seq=278 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=279 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_seq=280 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=281 ttl=64 time=0.019 ms

[2]+  Stopped                  ping localhost
[nesterenko_mikola@vpsj3IeQ ~]$
```

5. У другому терміналі отримаємо список фонових процесів

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -aux |egrep "ping localhost"
nester+  636  0.0  0.0 130680 1644 pts/14  T    01:31   0:00 ping localhost
nester+  636  0.0  0.0 130680 1644 pts/14  T    01:31   0:00 ping localhost
```

6. У другому терміналі відновимо виконання припиненого процесу

```
64 by[nesterenko_mikola@vpsj3IeQ ~]$ kill -18 636
64 by[nesterenko_mikola@vpsj3IeQ ~]$
64 bytes from localhost (127.0.0.1): icmp_seq=278 ttl=64 time=0.018 ms
64 bytes from localhost (127.0.0.1): icmp_seq=279 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_seq=280 ttl=64 time=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=281 ttl=64 time=0.019 ms

[2]+  Stopped                  ping localhost
[nesterenko_mikola@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=282 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=283 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=284 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=285 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=286 ttl=64 time=0.022 ms
64 bytes from localhost (127.0.0.1): icmp_seq=287 ttl=64 time=0.030 ms
64 bytes from localhost (127.0.0.1): icmp_seq=288 ttl=64 time=0.015 ms
```

7. У другому терміналі зупинимо виконання процесу команди ping


```

64[nesterenko_mikola@vpsj3IeQ ~]$ kill -9 16613
64[nesterenko_mikola@vpsj3IeQ ~]$ 
64 bytes from localhost (127.0.0.1): icmp_seq=27 ttl=64 time=0.030 ms
64 bytes from localhost (127.0.0.1): icmp_seq=28 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=29 ttl=64 time=0.032 ms
Killed
[nesterenko_mikola@vpsj3IeQ ~]$ 

```

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

```

[nesterenko_mikola@vpsj3IeQ ~]$ nohup ping localhost &
[2] 25118
[nesterenko_mikola@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'

```

9. Закрийте перший термінал.

10. У другому терміналі для команди ping отримаємо таблицю її процесу (колонки PID, STAT, CMD). Зробимо висновок про стан процесу. - спячий

```

[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 25118 -o pid,stat,cmd
  PID STAT CMD
 25118 S    ping localhost
[nesterenko_mikola@vpsj3IeQ ~]$ 

```

11. Завершимо роботу процесу.

```

[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 25118 -o pid,stat,cmd
  PID STAT CMD
 25118 S    ping localhost
[nesterenko_mikola@vpsj3IeQ ~]$ 

```

4 Управління пріоритетами процесів

1. Створемо 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
#!/bin/bash
x=10
n=6
while ( true )
do
    x=$(( $x+$n ))
done

```

2. Запускаємо bash-програму у фоновому режимі.

```
[nesterenko_mikola@vpsj3IeQ ~]$ ./nesterenko.sh&
[1] 14520
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 14520 -o pid,ppid,stat,ni,cpu,cmd
PID PPID STAT NI CPU CMD
14520 12134 S 0 - /bin/bash ./nesterenko.sh
[nesterenko_mikola@vpsj3IeQ ~]$
```

4. Виконуємо команду призупинення запущеного процесу

```
[nesterenko_mikola@vpsj3IeQ ~]$ kill -19 14520
[nesterenko_mikola@vpsj3IeQ ~]$ jobs
[1]+  Stopped                  ./nesterenko.sh
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 14520 -o pid,ppid,stat,ni,cpu,cmd
PID PPID STAT NI CPU CMD
14520 12134 T 0 - /bin/bash ./nesterenko.sh
[nesterenko_mikola@vpsj3IeQ ~]$
```

6. Виконаємо команду продовження виконання припиненого процесу.

```
[nesterenko_mikola@vpsj3IeQ ~]$ kill -18 14520
[nesterenko_mikola@vpsj3IeQ ~]$ jobs
[1]+  Running                  ./nesterenko.sh &
[nesterenko_mikola@vpsj3IeQ ~]$
```

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

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 14520 -o pid,ppid,stat,ni,cpu,cmd
PID PPID STAT NI CPU CMD
14520 12134 S 0 - /bin/bash ./nesterenko.sh
[nesterenko_mikola@vpsj3IeQ ~]$
```

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`

```
[nesterenko_mikola@vpsj3IeQ ~]$ ln -s nesterenko.sh nesterenko2.sh
[nesterenko_mikola@vpsj3IeQ ~]$ ln -s nesterenko.sh nesterenko3.sh
[nesterenko_mikola@vpsj3IeQ ~]$ ls
1.csv                               nesterenko_lab_3
2.csv                               NesterenkoMykola.csv
3.csv                               nesterenko.sh
accounts.csv                        nesterenko.txt
darkdir                             new.txt
directory                           nohup.out
glue_csv.sh                         null.txt
hard_link_1                         Operating-System.-Laboratory-Work-1
hard_link_2                         Operating-System.-Laboratory-Work-4
index.html                          os.lab1.cp1251.html
lowchar                             os.lab1.utf.html
morethanfourteencharsbutnodigits  sym_link_1
my_delete_directory.sh             testdoc.docx
MyOSParam.sh                       testdoc.pdf
nesterenko2.sh                     BigxoguIIVкласівнебезпеку
nesterenko3.sh
```

9. Запустимо два файли у фоновому режимі

```
[nesterenko_mikola@vpsj3IeQ ~]$ ./nesterenko2.sh& ./nesterenko3.sh&
[2] 11578
[3] 11579
[nesterenko_mikola@vpsj3IeQ ~]$
```

10. Переглянемо таблицю процесів для трьох запущених процесів з урахуванням набором колонок: PID, PPID, STAT, NI, %CPU, CMD. Зробимо висновки за поточними значеннями NI та %CPU. - мають однаковий рівень пріоритетності ,даних про процесор немає

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 14520 -p 11578 -p 11579 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
11578 12134  S      0   -  /bin/bash ./nesterenko2.sh
11579 12134  S      0   -  /bin/bash ./nesterenko3.sh
14520 12134  S      0   -  /bin/bash ./nesterenko.sh
[nesterenko_mikola@vpsj3IeQ ~]$
```

11. Зменшимо пріоритет виконання одного з трьох процесів. (для `nesterenko2.sh`)

```
[nesterenko_mikola@vpsj3IeQ ~]$ renice -n 5 -p 11578
11578 (process ID) old priority 0, new priority 5
[nesterenko_mikola@vpsj3IeQ ~]$
```

12. Переглянемо 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: рівень пріоритетності у процесі 11578 зменшився на 5

```
[nesterenko_mikola@vpsj3IeQ ~]$ ps -p 14520 -p 11578 -p 11579 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
11578 12134  SN      5   -  /bin/bash ./nesterenko2.sh
11579 12134  S       0   -  /bin/bash ./nesterenko3.sh
14520 12134  S       0   -  /bin/bash ./nesterenko.sh
[nesterenko_mikola@vpsj3IeQ ~]$
```

Висновки: в ході виконання лабораторної роботи, були придбані навички управління процесами в ОС Unix засобами командної оболонки.

Складнощів у виконанні роботи не було виявлено.