

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
Дорожкін Михайло

Перевірив(-ла):
Блажко О. А.
Дрозд М.О.

Одеса 2021

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

Завдання на виконання:

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 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.

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 виконання роботи:

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.

№ команди	№ учасника команди	Пам'ять, N Гб	ОС, M Гб	Програма, K Гб	Очікування завершення вводу/виводу, F
5	1	14	2	0.9	60

Пам'ять комп'ютера становить 14 Гбайт, ОС її таблиці процесів займають до 2 Гбайт, а кожна програма також займає до 0,9 Гбайт. Цей обсяг дозволяє одночасно розмістити в пам'яті 13 програм. При середньому очікуванні вводу/виводу, що становить 60% часу, ми маємо завантаженість процесора (якщо ігнорувати витрати на роботу самої ОС), $1 - 0.6^{13} = 0.998$, або близько 100%.

Завдання 2 Перегляд таблиці процесів

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

```
dorozhkin_mihajlo@vpsj3IeQ:~  
[dorozhkin_mihajlo@vpsj3IeQ ~]$ pstree -u  
systemd--NetworkManager--2*[{NetworkManager}]  
--agetty  
--auditd--({auditd})  
--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}]  
--php-fpm--3*[{php-fpm(emps)}]  
--php-fpm--3*[{php-fpm(soft)}]  
--ping(oracle)  
--ping(bojchuk_oleksandr)  
--polkitd(polkitd)--6*[{polkitd}]  
--pure-ftpd  
--rpcbind(rpc)  
--rsyslogd--2*[{rsyslogd}]  
--smartd  
--soffice.bin(nosov_andrij)--2*[{soffice.bin}]  
--soffice.bin(bojchuk_oleksandr)--2*[{soffice.bin}]  
--soffice.bin(shostak_roman)--2*[{soffice.bin}]  
--sshd--sshd(guda_liliya)--bash  
--sshd--2*[{sshd--bash--su--bash(oracle)}]  
--sshd--sshd(babich_artem)--bash--less  
--sshd--sshd(ozarchuk_anna)--bash--2*[{top}]  
--sshd--sshd(belobrov_artur)--bash--top  
--sshd--sshd(babich_artem)--bash--ping  
--sshd--sshd(babich_artem)--bash--top  
--sshd--sshd(veselkova_anna)--bash
```

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ pstree -p dorozhkin_mihajlo  
sshd(22360)---bash(22361)---pstree(22699)  
[dorozhkin_mihajlo@vpsj3IeQ ~]$
```

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
 22361 pts/18    00:00:00 bash
 22763 pts/18    00:00:00 ps
[dorozhkin_mihajlo@vpsj3IeQ ~]$
```

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -F -u dorozhkin_mihajlo
UID          PID  PPID  C  SZ  RSS  PSR  STIME  TTY          TIME CMD
dorozhk+ 22360 22316  0 41473 2480   1 23:05 ?            00:00:00 sshd: dorozhkin_mihajlo@pts
dorozhk+ 22361 22360  0 28888 2124   1 23:05 pts/18    00:00:00 -bash
dorozhk+ 22861 22361  0 38869 1860   0 23:10 pts/18    00:00:00 ps -F -u dorozhkin_mihajlo
```

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -u dorozhkin_mihajlo -o pid, tty, ppid, stat, ni, cmd
  PID TT          PPID  STAT  NI  CMD
 22360 ?            22316  S      0  sshd: dorozhkin_mihajlo@pts/18
 22361 pts/18        22360  Ss     0  -bash
 23055 pts/18        22361  R+     0  ps -u dorozhkin_mihajlo -o pid, tty, ppid, stat, ni, cmd
```

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

```
dorozhkin_mihajlo@vpsj3IeQ:~
[dorozhkin_mihajlo@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/vdal-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
888 ?          S      0:00 php-fpm: pool regular
2182 ?         S      0:00 [kworker/u4:1]
7126 ?         S      0:22 ping localhost
10111 ?        S      0:00 /usr/local/apps/apache/bin/httpd -k start
11194 ?        S      0:00 sshd: guda_liliya@pts/10
12335 ?        S      0:00 nano 3.sh
12757 ?        S      0:03 ping localhost
```

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

а. загальна кількість запущених процесів - 174;

б. кількість процесів, які виконуються - 7;

с. кількість сплячих процесів - 163.

```
dorozhkin_mihajlo@vpsj3leQ:~
top - 23:18:24 up 54 days, 6:21, 15 users, load average: 3.97, 4.08, 4.20
Tasks: 174 total, 7 running, 163 sleeping, 4 stopped, 0 zombie
%Cpu(s): 90.9 us, 8.3 sy, 0.7 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem : 1881856 total, 162072 free, 349568 used, 1370216 buff/cache
KiB Swap: 4194300 total, 3715068 free, 479232 used. 1406740 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 7574 oracle    20   0 113288  1188  1004  R   99.0   0.1   3449:15 test2.sh
 7548 oracle    20   0 113288  1188  1004  R   98.7   0.1   3449:44 test1.sh
 7577 oracle    39  19 113288  1188  1000  R    1.7   0.1   52:03.96 test3.sh
   500 root       20   0 476444  1696  1000  R    0.7   0.1   770:52.40 NetworkManager
22756 belobro+   20   0 162132  2340  1600  S    0.3   0.1    0:00.99 top
23410 plaksiv+   20   0 162128  2400  1600  S    0.3   0.1    0:00.18 top
23440 dorozhk+  20   0 162156  2368  1600  R    0.3   0.1    0:00.17 top
23579 guda_li+   20   0 162052  4812  3544  S    0.3   0.3    0:00.01 mc
    1 root       20   0 191168  3004  1668  S    0.0   0.2   40:29.70 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
```

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

```
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 7548 oracle    20   0 113288  1188  1004  R   99.0   0.1   3451:14 test1.sh
 7574 oracle    20   0 113288  1188  1004  R   98.0   0.1   3450:44 test2.sh
 7577 oracle    39  19 113288  1188  1000  R    1.7   0.1   52:05.25 test3.sh
   357 root       20   0 39232   3344  3196  S    0.3   0.2   36:47.56 systemd-journal
   500 root       20   0 476444  1696  1000  S    0.3   0.1   770:52.63 NetworkManager
23410 plaksiv+   20   0 162128  2400  1600  S    0.3   0.1    0:00.41 top
    1 root       20   0 191168  3004  1668  S    0.0   0.2   40:29.74 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    0:24.37 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
    9 root       20   0      0      0      0  S    0.0   0.0   141:00.50 rcu_sched
   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    0:17.03 watchdog/0
   12 root       rt   0      0      0      0  S    0.0   0.0    0:19.80 watchdog/1
   13 root       rt   0      0      0      0  S    0.0   0.0    0:05.93 migration/1
   14 root       20   0      0      0      0  S    0.0   0.0   12:11.21 ksoftirqd/1
   16 root       0 -20      0      0      0  S    0.0   0.0    0:00.00 kworker/1:0H
```

Завдання 3 Керування станами процесів

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

```
dorozhkin_mihajlo@vpsj3IeQ:~  
[dorozhkin_mihajlo@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=0.021 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.031 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.032 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.030 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.057 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.031 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.033 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.040 ms
```

2. Запустіть другий термінал доступу до Linux-сервера.

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 23956 -o pid,stat,cmd  
PID STAT CMD  
23956 S+ ping localhost  
[dorozhkin_mihajlo@vpsj3IeQ ~]$
```

4. У другому терміналі призупиніть виконання процесу команди `ping`

```
dorozhkin_mihajlo@vpsj3IeQ:~  
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -19 23956  
[dorozhkin_mihajlo@vpsj3IeQ ~]$  
[2]+ Stopped ping localhost  
[dorozhkin_mihajlo@vpsj3IeQ ~]$
```

5. У першому терміналі отримайте список фонових процесів

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -aux | egrep "ping localhost"  
oracle 7126 0.0 0.0 130680 1644 ? S Apr09 0:22 ping localhost  
bojchuk+ 12757 0.0 0.0 130680 1644 ? S 14:11 0:03 ping localhost  
dorozhk+ 23956 0.0 0.0 130680 1644 pts/18 T 23:21 0:00 ping localhost
```

6. У другому терміналі відновіть виконання припиненого процесу

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -19 23956  
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -aux | egrep "ping"  
oracle 7126 0.0 0.0 130680 1644 ? S  
bojchuk+ 12757 0.0 0.0 130680 1644 ? S  
dorozhk+ 23956 0.0 0.0 130680 1644 pts/18 T  
babich_+ 25847 0.0 0.0 130680 1640 ? S  
plaksiv+ 25903 0.0 0.0 130680 1644 pts/22 S+  
dorozhk+ 26116 0.0 0.0 112812 976 pts/10 R+  
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -18 23956  
[dorozhkin_mihajlo@vpsj3IeQ ~]$  
[2]+ Stopped ping localhost  
[dorozhkin_mihajlo@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_  
time=0.028 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1290 ttl=64 time=0.041 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1291 ttl=64 time=0.036 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1292 ttl=64 time=0.034 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1293 ttl=64 time=0.032 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1294 ttl=64 time=0.032 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1295 ttl=64 time=0.034 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=1296 ttl=64 time=0.032 ms
```


7. У другому терміналі зупиніть виконання процесу команди `pingps`

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -9 23956
```

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ nohup ping localhost &
[2] 26940
[dorozhkin_mihajlo@vpsj3IeQ ~]$ nohup: ignoring input and appending output to 'nohup.out'
```

9. Закрийте перший термінал.

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 26940 -o pid,stat,cmd
PID STAT CMD
26940 S    ping localhost
```

11. Завершіть роботу процесу.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -9 26940
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 26940 -o pid,stat,cmd
PID STAT CMD
[dorozhkin_mihajlo@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`

```
#!/bin/bash
x=8
n=7
while( true )
do
    x=$((x+n))
done
```

[Read 7 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.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ sh dorozhkin.sh&
[1] 23260
[dorozhkin_mihajlo@vpsj3IeQ ~]$ █
```

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

```
dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 23260 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
 3260 25158  S      0  - sh dorozhkin.sh
dorozhkin_mihajlo@vpsj3IeQ ~]$ █
```

4. Виконайте команду призупинення запущеного процесу.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -19 23260
[dorozhkin_mihajlo@vpsj3IeQ ~]$ █
```

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ jobs
[1]+  Stopped                  sh dorozhkin.sh
[dorozhkin_mihajlo@vpsj3IeQ ~]$ █
```

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

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 23260 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
 23260 25158  T      0  - sh dorozhkin.sh
[dorozhkin_mihajlo@vpsj3IeQ ~]$ █
```

6. Виконайте команду продовження виконання припиненого процесу.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ kill -18 23260
[dorozhkin_mihajlo@vpsj3IeQ ~]$ jobs
[1]+  Running                  sh dorozhkin.sh &
```

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.

```
[1]+  Running                  sh dorozhkin.sh &
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 23260 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
 23260 25158  S      0  - sh dorozhkin.sh
[dorozhkin_mihajlo@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`

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ln -s dorozhkin.sh dorozhkin2.sh
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ln -s dorozhkin.sh dorozhkin3.sh
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ls
1.csv          dorozhkin2.sh      Lab2.docx
2.csv          dorozhkin3.sh      Lab2.pdf
```

9. Запустіть два файли у фоновому режимі.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ sh dorozhkin2.sh&
[2] 8776

[dorozhkin_mihajlo@vpsj3IeQ ~]$ sh dorozhkin3.sh&
[3] 10879
```

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.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 23260 -p 8776 -p 10879 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
  8776  25158  S      0  -  sh dorozhkin2.sh
 10879  25158  S      0  -  sh dorozhkin3.sh
 23260  25158  S      0  -  sh dorozhkin.sh
```

11. Зменшить пріоритет виконання одного з трьох процесів.

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ renice -n 2 -p 8776
8776 (process ID) old priority 0, new priority 2
```

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?

```
[dorozhkin_mihajlo@vpsj3IeQ ~]$ ps -p 23260 -p 8776 -p 10879 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI CPU CMD
  8776  25158  SN     2  -  sh dorozhkin2.sh
 10879  25158  S      0  -  sh dorozhkin3.sh
 23260  25158  S      0  -  sh dorozhkin.sh
```

Висновок: Під час виконання лабораторної роботи були отримані навички управління та аналізу процесів в ОС Unix. Під час ходу виконання роботи було складним робота у двох терміналах одночасно.