

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»

Виконала:  
ст.гр. АІ -204  
Сіренко Марія

Перевірили:  
Блажко О.А.  
Дрозд М.О

**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.

Таблиця 3 – Варіанти завдань

№ команди	№ учасника команди	Пам'ять, $N$ Гб	ОС, $M$ Гб	Програма, $K$ Гб	Очікування завершення вводу/виводу, $F$
8	2	12	1	0.5	35

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 `ping` отримаєте таблицю її процесу (колоники 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 запущений.
9. Закрийте перший термінал.
10. У другому терміналі для команди `ping` отримаєте таблицю її процесу (колоники 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?

### Виконання завдань:

#### 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, $F$
8	2	12	1	0.5	35

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

Кільсть процесів:  $(12-1):0.5=22$

Середня завантеженість процесу  $1-0.35^{22}=0.99$

#### 2 Перегляд таблиці процесів

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

```
[sirenko_mariya@vpsj3IeQ ~]$ pstree -u
systemd--NetworkManager--2*[{NetworkManager}]
      |
      |--agetty
      |--auditd--{auditd}
      |--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)
      |--nginx--nginx(emps)
      |--php-fpm--6*[php-fpm(soft)]
      |--php-fpm--3*[php-fpm]
      |       |--3*[php-fpm(emps)]
      |       |--3*[php-fpm(soft)]
      |--ping(oracle)
      |--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--bash
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ pstree -p sirenko_mariya
sshd(30714)--bash(30715)--pstree(30863)
[sirenko_mariya@vpsj3IeQ ~]$ █
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ ps
  PID TTY          TIME CMD
 30715 pts/5        00:00:00 bash
 30900 pts/5        00:00:00 ps
[sirenko_mariya@vpsj3IeQ ~]$ █
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -F -u sirenko_mariya
UID      PID  PPID  C   SZ   RSS  PSR  STIME  TTY      TIME  CMD
sirenko+ 30714 30615  0 39883 2516   1 19:40 ?        00:00:00 sshd: sirenko_mariya@pts/5
sirenko+ 30715 30714  0 28887 2140   1 19:40 pts/5    00:00:00 -bash
sirenko+ 31019 30715  0 38869 1920   0 19:44 pts/5    00:00:00 ps -F -u sirenko_mariya
[sirenko_mariya@vpsj3IeQ ~]$
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -u sirenko_mariya -o pid,TTY,PPID,STAT,NI,CMD
  PID TT      PPID STAT  NI CMD
30714 ?        30615 S      0 sshd: sirenko_mariya@pts/5
30715 pts/5    30714 Ss     0 -bash
31084 pts/5    30715 R+    0 ps -u sirenko_mariya -o pid,TTY,PPID,STAT,NI,CMD
[sirenko_mariya@vpsj3IeQ ~]$
```

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

```
[sirenko_mariya@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      0:16 [watchdog/0]
 12 ?      S      0:19 [watchdog/1]
 13 ?      S      0:05 [migration/1]
 14 ?      S     12:10 [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      2:14 [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
888 ?      S      0:00 php-fpm: pool regular
912 ?      S      0:00 [kworker/u4:1]
7126 ?     S      0:17 ping localhost
12335 ?    S      0:00 nano 3.sh
19044 ?    S      0:00 [kworker/u4:2]
20685 ?    S      0:00 nano
27290 ?    S      0:00 [kworker/1:1]
27540 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
27541 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
27546 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
27556 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
27557 ?    S      0:00 /usr/local/apps/apache/bin/httpd -k start
27633 ?    S      0:00 [kworker/0:0]
```

## 7. Отримаємо список процесів, відсортованих по PID(за спаданням) Shift+N

За total визначимо :

а. загальна кількість запущених процесів; 129

б. кількість процесів, які виконуються; 5

с. кількість сплячих процесів. 124

```
top - 19:59:47 up 53 days, 3:03, 4 users, load average: 4,51, 4,38, 4,28
Tasks: 129 total, 5 running, 124 sleeping, 0 stopped, 0 zombie
%Cpu(s): 91,2 us, 8,0 sy, 0,7 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,2 si, 0,0 st
KiB Mem : 1881856 total, 507460 free, 110136 used, 1264260 buff/cache
KiB Swap: 4194300 total, 3682556 free, 511744 used. 1643980 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
32267	bojchuk+	20	0	115680	2184	1664	S	0,0	0,1	0:00.26	bash
32266	bojchuk+	20	0	165888	2600	1092	S	0,0	0,1	0:00.57	sshd
32251	root	20	0	165888	6644	5136	S	0,0	0,4	0:00.34	sshd
32195	root	20	0	112924	4308	3276	S	0,0	0,2	0:00.00	sshd
32192	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/0:1
32138	sirenko+	20	0	162124	2304	1628	R	0,0	0,1	0:00.16	top
31970	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kworker/1:0
31721	sirenko+	20	0	115548	2152	1700	S	0,0	0,1	0:00.01	bash
31720	sirenko+	20	0	159400	2444	1060	S	0,0	0,1	0:00.05	sshd
31715	root	20	0	159400	6140	4752	S	0,0	0,3	0:00.25	sshd
31494	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/1:1
31422	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0
30715	sirenko+	20	0	115548	2144	1696	S	0,0	0,1	0:00.03	bash
30714	sirenko+	20	0	159532	2516	1048	S	0,0	0,1	0:00.05	sshd
30667	root	20	0	0	0	0	S	0,0	0,0	0:00.06	kworker/1:3
30615	root	20	0	159532	6224	4756	S	0,0	0,3	0:00.28	sshd
30262	root	20	0	0	0	0	S	0,0	0,0	0:00.26	kworker/u4:0
29868	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kworker/0:2
25780	apache	20	0	113380	1888	1124	S	0,0	0,1	0:00.00	httpd
23429	apache	20	0	113380	1776	1196	S	0,0	0,1	0:00.30	httpd
23428	apache	20	0	113380	1760	1184	S	0,0	0,1	0:00.25	httpd
23427	apache	20	0	113380	1972	1332	S	0,0	0,1	0:00.25	httpd
21923	nosov_a+	20	0	696320	864	836	S	0,0	0,0	0:08.73	soffice.bin
21071	root	20	0	0	0	0	S	0,0	0,0	0:00.06	kworker/u4:2
20685	baranyu+	20	0	115072	512	508	S	0,0	0,0	0:00.00	nano
20460	root	30	10	279912	644	360	S	0,0	0,0	0:00.00	php-fpm

## 8. Отримаємо список процесів, відсортованих за % використання процесора.<Shift> + <P>

```
top - 20:01:56 up 53 days, 3:05, 4 users, load average: 4,16, 4,31, 4,27
Tasks: 130 total, 5 running, 125 sleeping, 0 stopped, 0 zombie
%Cpu(s): 91,0 us, 8,3 sy, 0,7 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 1881856 total, 507076 free, 110348 used, 1264432 buff/cache
KiB Swap: 4194300 total, 3682556 free, 511744 used. 1643652 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
7548	oracle	20	0	113288	1188	1004	R	100,0	0,1	1836:15	test1.sh
7574	oracle	20	0	113288	1188	1004	R	98,7	0,1	1835:37	test2.sh
7577	oracle	39	19	113288	1188	1000	R	1,3	0,1	28:32.01	test3.sh
32429	sirenko+	20	0	162152	2356	1632	R	0,3	0,1	0:00.04	top
1	root	20	0	191168	2840	1580	S	0,0	0,2	39:52.69	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:01.34	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	0:24.16	ksoftirqd/0
7	root	rt	0	0	0	0	S	0,0	0,0	0:09.84	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	R	0,0	0,0	140:29.89	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	0:16.74	watchdog/0
12	root	rt	0	0	0	0	S	0,0	0,0	0:19.42	watchdog/1
13	root	rt	0	0	0	0	S	0,0	0,0	0:05.91	migration/1
14	root	20	0	0	0	0	S	0,0	0,0	12:10.55	ksoftirqd/1
16	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0H

### 3 Керування станами процесів

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

```
[sirenko_mariya@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=0.029 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.034 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=0.030 ms
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 315 -o pid,stat,cmd
  PID STAT CMD
   315 S+  ping localhost
[sirenko_mariya@vpsj3IeQ ~]$
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ kill -19 1438
[sirenko_mariya@vpsj3IeQ ~]$
```

```
64 bytes from localhost (127.0.0.1): icmp_seq=14 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=15 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=16 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=17 ttl=64 time=0.046 ms
64 bytes from localhost (127.0.0.1): icmp_seq=18 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=19 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=20 ttl=64 time=0.028 ms
64 bytes from localhost (127.0.0.1): icmp_seq=21 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=22 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=23 ttl=64 time=0.031 ms
64 bytes from localhost (127.0.0.1): icmp_seq=24 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=25 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=26 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=27 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=28 ttl=64 time=0.032 ms
64 bytes from localhost (127.0.0.1): icmp_seq=29 ttl=64 time=0.033 ms
64 bytes from localhost (127.0.0.1): icmp_seq=30 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=31 ttl=64 time=0.030 ms
64 bytes from localhost (127.0.0.1): icmp_seq=32 ttl=64 time=0.032 ms

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



5. У другому терміналі отримаємо список фонових процесів

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -aux |egrep "ping localhost"
sirenko+ 2650  0.0  0.0 130680 1692 pts/17  T   20:34  0:00 ping localhost
```

6. У другому терміналі відновимо виконання припиненого процесу

```
File Edit View Bookmarks Settings Help
[sirenko_mariya@vpsj3IeQ ~]$ ps -aux |egrep "ping localhost"
sirenko+ 3015  0.0  0.0 130680 1688 pts/18  T   20:39  0:00 ping localhost
sirenko+ 3025  0.0  0.0 112812 1024 pts/16  R+   20:39  0:00 grep -E --color=auto ping l
ocalhost
oracle 7126  0.0  0.0 130680 1644 ?        S    kb109  0:12 ping localhost
[sirenko_mariya@vpsj3IeQ ~]$ kill -18 3015
[sirenko_mariya@vpsj3IeQ ~]$ █

ash@ThinkPad-E14:~$ ssh sirenko_mariya@91.219.60.189
sirenko_mariya@91.219.60.189's password:
Last login: Sat Apr 10 20:31:58 2021 from host-91-90-11-33.soborka.net
sirenko_mariya@vpsj3IeQ ~]$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
4 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.020 ms
4 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.030 ms
4 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.033 ms
4 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.049 ms
^C
1]+ Stopped                  ping localhost
sirenko_mariya@vpsj3IeQ ~]$ 64 bytes from localhost (127.0.0.1): icmp_seq=5 tt
0 ms
4 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.033 ms
4 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.025 ms
4 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.040 ms
```

7. У другому терміналі зупинимо виконання процесу команди ping

```
[sirenko_mariya@vpsj3IeQ ~]$ kill -9 3015
[sirenko_mariya@vpsj3IeQ ~]$ ps -aux |egrep "ping localhost"
sirenko+ 3169  0.0  0.0 112812 1024 pts/16  R+   20:40  0:00 grep -E --color=auto ping l
ocalhost
oracle 7126  0.0  0.0 130680 1644 ?        S    kb109  0:12 ping localhost
[sirenko_mariya@vpsj3IeQ ~]$ █
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ nohup ping localhost &
[1] 3333
[sirenko_mariya@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 (колони PID, STAT, CMD). Зробимо висновок про стан процесу. - сплячий

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 3333 -o pid,stat,cmd
PID STAT CMD
3333 S    ping localhost
[sirenko_mariya@vpsj3IeQ ~]$ █
```

11. Завершимо роботу процесу.

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 3333 -o pid,stat,cmd
PID STAT CMD
[sirenko_mariya@vpsj3IeQ ~]$ █
```

#### Завдання 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 File: sirenko.s
#!/bin/bash
x=7                                #sirenko
n=6                                #maria
while( true )
do
    x=$((x+n))
done
```

2. Запускаємо bash-програму у фоновому режимі.

```
[sirenko_mariya@vpsj3IeQ ~]$ ./sirenko.sh&
[1] 26903
[sirenko_mariya@vpsj3IeQ ~]$ █
```

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

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 26903 -o pid,ppid,stat,ni,cpu,cmd
PID  PPID STAT  NI CPU CMD
26903 26514 S      0  -  -bash
[sirenko_mariya@vpsj3IeQ ~]$ █
```

```
[sirenko_mariya@vpsj3IeQ ~]$ jobs
[1]+  Running                  ./sirenko.sh &
[sirenko_mariya@vpsj3IeQ ~]$ █
```

4. Виконуємо команду призупинення запущеного процесу.

```
[sirenko_mariya@vpsj3IeQ ~]$ kill -19 26903
[sirenko_mariya@vpsj3IeQ ~]$ jobs
[1]+  Stopped                  ./sirenko.sh
[sirenko_mariya@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

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 26903 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
26903 26514  T      0   -  -bash
[sirenko_mariya@vpsj3IeQ ~]$ █
```

6. Виконаємо команду продовження виконання припиненого процесу.

```
-----
[sirenko_mariya@vpsj3IeQ ~]$ kill -18 26903
[sirenko_mariya@vpsj3IeQ ~]$ jobs
[1]+  Running                  ./sirenko.sh &
[sirenko_mariya@vpsj3IeQ ~]$ █
```

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.-сплячий

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 26903 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
26903 26514  S      0   -  -bash
[sirenko_mariya@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

```
[sirenko_mariya@vpsj3IeQ ~]$ ln -s sirenko.sh sirenko2.sh
[sirenko_mariya@vpsj3IeQ ~]$ ln -s sirenko.sh sirenko3.sh
[sirenko_mariya@vpsj3IeQ ~]$ ls
1                               hard_link_1  mkFileDir.sh          sirenko          sirenko.sh  року
2021                           hard_link_2  my_change_file.sh     sirenko2.sh     sym_link_1  січня
3.csv                           lab_2.odt    MyOSProgram.sh        sirenko3.sh     test.html
accounts.csv                   lab_2.pdf    Operating-System.Laboratory-Work-1  sirenko_lab_3   на
file1.txt                      lab5_1234567 os.lab1.cp1251.html   sirenko_mariia_2.csv  населення
file.txt                      lab_6        os.lab1.utf.html      sirenko_mariia.csv   Наявне
[sirenko_mariya@vpsj3IeQ ~]$ █
```

9. Запустимо два файли у фоновому режимі.

```
[sirenko_mariya@vpsj3IeQ ~]$ ./sirenko2.sh& ./sirenko3.sh&
[2] 6892
[3] 6893
[sirenko_mariya@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 пріорітетності ,данниз про процессор немає

```
[sirenko_mariya@vpsj3IeQ ~]$ ps -p 26903 -p 6892 -p 6893 -o pid,ppid,stat,ni,cpu,cmd
  PID  PPID  STAT  NI  CPU  CMD
  6892 26514  S      0   -  -bash
  6893 26514  S      0   -  -bash
 26903 26514  R      0   -  -bash
[sirenko_mariya@vpsj3IeQ ~]$ █
```

11. Зменшимо пріоритет виконання одного з трьох процесів. (для sirenko2.sh)

```
[sirenko_mariya@vpsj3IeQ ~]$ renice -n 5 -p 6892
6892 (process ID) old priority 0, new priority 5
[sirenko_mariya@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 пріорітетності у процесі 6892 зменшився на 5

```
  PID  PPID  STAT  NI  CPU  CMD
  6892 26514  SN      5   -  -bash
  6893 26514  S      0   -  -bash
 26903 26514  R      0   -  -bash
[sirenko_mariya@vpsj3IeQ ~]$ █
```

Висновок: було отримано навички з управління процесами в ОС Unix засобами командної оболонки. При виконанні завдань ніяких труднощів не виникло